POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Prediction interval estimation on seasonal adjusted electricity prices using conformal prediction theory

Tesi di Laurea Magistrale in
Mathematical Engineering - Quantitative Finance

Author: **Luca Bellomi**

Student ID: 953314
Advisor: Prof. Roberto Baviera
Co-advisors: Dott. Pietro Manzoni
Academic Year: 2022-2023

Abstract

In the electricity markets, point forecast models for day-ahead prices are widely used. The usefulness of these models depends on their accuracy in forecasting. The open question is on their ability to estimate prediction uncertainty, resulting in tight and reliable prediction intervals. In this study, we explore the application of conformal prediction intervals. Specifically, our aim is to assess the impact of this method when applied to price time series after removing their seasonal components. We have observed a significant improvement in the quality of prediction intervals when applying this technique to regular data. However, for more challenging datasets with higher variance and more outliers, it remains more convenient to use the original price time series. Additionally, we have outlined that the quality of the intervals depends on the accuracy of the point forecast model, resulting in more reliable intervals when the model demonstrates higher accuracy in point predictions.

Keywords: EPF, Electricity price forecasting, Conformal prediction interval, tree-structured Parzen estimator, seasonal differencing.

Sommario

Nei mercati dell'energia elettrica, è consuetudine utilizzare modelli di previsione puntuale per i prezzi day-ahead. L'utilità di questi modelli dipende dalla loro accuratezza nelle previsioni. L'interrogativo che rimane aperto riguarda la loro capacità di stimare l'incertezza della previsione, producendo intervalli di previsione stretti e affidabili. In questo studio, esploriamo l'applicazione di intervalli di previsione conformi. In particolare, miriamo a valutare l'impatto di questo metodo quando applicato a serie temporali di prezzi dopo aver rimosso le loro componenti stagionali. Abbiamo evidenziatio come l'uso di questa tecnica possa migliorare significativamente la qualità degli intervalli di previsione se applicata a dati regolari. Tuttavia, quando il dataset risulta più difficile da prevedere a causa di una maggiore varianza, o data la presenza di più valori anomali, le prestazioni di questa tecnica si deteriorano, ed è più conveniente utilizzare la serie temporale originale dei prezzi. Inoltre, osserviamo che la qualità degli intervalli dipende dall'accuratezza del modello di previsione puntuale, producendo intervalli più affidabili quando il modello dimostra una maggiore accuratezza nella previsione.

Contents

1 | Introduction

1.1. General Framework

The growing integration of renewable energy sources into modern power systems has heightened the fluctuations in electricity generation. Consequently, predicting electricity prices has become more challenging than ever before. In addition to short-term load forecasting, short-term electricity price forecasting (EPF) has emerged as a central component of an energy company's operational activities. Due to the structure of day-ahead markets, market players must submit their bids the day before actual trades take place. Therefore, a robust and accurate model for short-term electricity price forecasting can have a significant impact on reducing a company's operational costs.

In addition to providing point predictions, it is crucial to assess the uncertainty in these predictions. This enables market participants to adjust their trading volumes in line with their confidence in the realized price. Given the growing complexity of price prediction, advances in electricity price forecasting (EPF) continually introduce new tools aimed at bridging the gap between forecasts and actual prices. Developments in this field have accelerated over the past decade, driven by the increasing availability of large datasets and hardware advancements, such as GPUs, which can be utilized to train machine learning models.

Electricity price and uncertainty forecasting presents a challenging task for several reasons. Firstly, and most significantly, electricity markets are often driven by irrational behavior, and significant price movements can be triggered by exogenous events such as press releases, geopolitical conflicts, and weather-related incidents. Consequently, the information that could be crucial for predicting new observations is typically not embedded in the market data itself.

In particular, as a larger portion of electricity demand is satisfied by renewable energy sources, price volatility increases, and prices become more dependent on external, uncontrollable factors. This makes it increasingly challenging to obtain accurate predictions, and the field of electricity price forecasting (EPF) will need to continue its research efforts

and adapt to the evolving market dynamics.

1.2. Brief description of this work

Since the emergence of electronic markets, a substantial body of literature has been dedicated to the prediction of electricity prices. This is partly due to the abundance of freely available public data, in contrast to many other financial time series analyses.

The early research in this area involves books (see e.g Bunn (2004), and Weron (2007)) where the authors initiated the investigation of the difficulties in electricity price prediction, the data structures involved, and the application of both statistical and machine learning models.

After these contributions, the literature on Electric Price Forecasting (EPF) has experienced a significant increase, leading to numerous published articles that continue to advance research in this field. The progress in the field, however, has not been consistent and is challenging to track. Major review publications have concluded that comparing EPF methods is very difficult due to variations in datasets, software implementations, and error measures used across different studies.

Among these studies, some present advanced statistical techniques for short-term price point prediction (see e.g Uniejewski et al. (2017); Marcjasz et al. (2019); and Cruz et al. (2011)).

We can also find several research studies that introduce new machine learning models in the field of Electric Price Forecasting (EPF) (see e.g. Wang et al. (2016); Ugurlu et al. (2018); Zhang et al. (2018); Luo and Weng (2019)).

Regarding the point forecast methods used in this study, we were inspired by an article (see e.g Uniejewski et al. (2016)) that extensively explored and compared various linear autoregressive models. We chose to implement two of the best-performing models, with some modifications related to hyperparameter optimization. We also employed a more sophisticated model (see e.g Lago et al. (2021)) that exhibited superior point forecast performance. This model is a deep neural network, and we optimized its hyperparameters using Bayesian optimization techniques.

On the prediction intervals front, which constitutes the core of our work, the available literature is more limited. Some of the early studies that introduced models for uncertainty prediction (see e.g Zhao et al. (2008)) proposed methods by incorporating a heteroscedastic variance in a Support Vector Machine (SVM). In another significant paper (see e.g

Zhou et al. (2006)), the authors explore a parametric approach to uncertainty prediction by using an ARIMA (AutoRegressive Integrated Moving Average) model for point forecast. Among the more recent studies that utilize advanced methods for constructing prediction intervals (see e.g Nowotarski and Weron (2015)), we find the quantile regression for uncertainty estimation, a technique that is widely used today.

In the article on which our study is based (see e.g Kath and Ziel (2021)), the authors compare the quantile regression technique against the conformal prediction intervals, obtaining promising results for this new uncertainty prediction technique. In particular, they empirically prove that the normalized conformal prediction intervals produce narrow and reliable intervals in the field of EPF, which could outperform quantile regression averaging. We aim to delve deeper into this study by exploring the application of this new technique on data with the seasonal component removed, using models of varying complexity, and on datasets with different characteristics.

1.3. Chapters summary

This thesis is organized as follows:

- **Chapter 2**: in this chapter, we offer a concise overview of electricity markets, their underlying infrastructure, and the key stakeholders engaged in these markets. Subsequently, we provide a comprehensive explanation of the day-ahead and intra-day markets. Lastly, we introduce the exchanges that we consider.

- **Chapter 3**: we introduce the data we utilize, outlining the essential features of these datasets, and describing the techniques employed to derive a deseasonalized price time series.

- **Chapter 4:** we describe the linear models we utilize, and their structure, we explain the methods employed for hyperparameter optimization, and we detail the process of building and optimizing the final model.

- **Chapter 5:** in this chapter, we introduce the neural network model. It is a more complex model compared to the linear models, with superior point prediction capabilities. We consider a Bayesian optimization method to select the best hyperparameters for each dataset.

- **Chapter 6:** we introduce conformal prediction intervals and a modified version of it, which we use as the final model for comparison.

- **Chapter 7:** results are presented and discussed.

- **Chapter 8:** we conclude, highlighting the study's advantages and limitations, and present some ideas for future work.

2 | Electricity Markets

2.1. Overview

An outline of the European Electricity Market should be preliminary to the acknowledgment of this work (see e.g. Glachant et al. (2021)). The infrastructure and the market participants, the features of electricity order-driven markets, and the specific exchanges that have been studied. The most important and unique feature of these markets is that electricity, in most of the cases, is a **non-storable** commodity. This fact is crucial in the discussion that follows.

2.2. The Infrastructure

The electricity infrastructure is a critical sector on a global scale. A stable electricity supply is essential for the well-being of communities, as it underpins public health and safety. Virtually all economic activities require a reliable power supply.

- **Generation:** Electric power in Europe is produced in various types of power plants converting into electricity all sort of different energy sources: Fossil Fuels, Nuclear Energy, Biomass, Hydroelectric, Geothermal, Wind and Solar radiation [1]. Each one of these fuels present some limitations and a well-being community should always plan a wide combination of these in order to maintain power supply reliability. Fossil fuels for instance are responsible for environmental problems and climate changes, Nuclear is considered potentially dangerous for nuclear disasters and produces toxic wastes. Hydroelectric and Geothermal cause environmental impacts on local communities and are limited by availability on site. Bio masses, wind and solar offer limited energy density as compared to traditional sources, furthermore wind and solar generate low voltage electricity.

[1] see European Commission website `https://energy.ec.europa.eu/energy-explained/energy-infrastructure-eu_en` (last access 22 November 2023)

- **Transmission:** After generation, electricity is sent through a high-voltage transmission network. This network consists of high-voltage power lines, substations, and other infrastructure, high voltage is required in order to reduce the amount of energy converted into heat and lost during transportation. Transmission networks connect power plants to regional distribution networks and high voltage is required in order to reduce the amount of energy lost in transport (see e.g. Glachant et al. (2021)).
- **Distribution:** Distribution networks deliver electric power from the transmission lines to homes, businesses, and local industries. These networks operate at lower voltages than transmission lines and include transformers, substations, and distribution lines (see e.g. Glachant et al. (2021)).

The key participants in the electrical infrastructure comprise:

- **Energy Producers:** These are organizations operating power plants responsible for energy generation. They can be either private companies or publicly owned entities and utilize various energy sources for power generation.
- **Transmission System Operators (TSO):** Each European country has one or more TSOs tasked with overseeing high-voltage transmission networks (In Italy is Terna). They manage the transportation of electrical energy at both national and international levels, ensuring a stable and reliable supply [2].
- **Distribution System Operators (DSO):** These are companies that operate at a regional or local level and manage the distribution networks of electricity in their respective areas of competence. They supply electricity to homes and businesses (see e.g. Anaya et al. (2022)).
- **Regulatory Authorities:** National or regional authorities are responsible for regulating the electricity sector. They establish policies, rules, and tariffs that influence the operation and development of the electrical infrastructure (In Italy is ARERA).
- **Union for the Coordination of Transmission of Electricity (UCTE):** his European organization coordinates the activities of transmission system operators at the continental level.
- **Energy Market Operators:** These organizations handle the buying and selling of electricity in energy markets, contributing to price establishment and ensuring efficient energy distribution.

[2] see Entsoe website `https://www.entsoe.eu/about/system-development/` (last access 22 November 2023)

This work is primarily directed towards the **Energy Market Operators**, as they stand to gain the most from precise electricity price predictions and a robust estimation of prediction uncertainty. These companies rely heavily on accurate forecasts to optimize their operations and ensure the efficient management of electricity markets.

2.3. Order-driven electricity markets

2.3.1. Day-ahead market

The day-ahead electricity market is a crucial element within liberalized energy markets where electricity producers, consumers, and intermediaries engage in buying and selling electricity for delivery on the following day. This market is also referred to as the **'spot market'** or **'wholesale day-ahead market'** (see e.g. Glachant et al. (2021)).

Here's how it operates (**Double auction process**)
In the lead-up to the day of electricity delivery (the day ahead), electricity producers, suppliers, and other market participants must submit their bids. These bids specify the quantity of electricity they are willing to supply and the price at which they are willing to provide it. After receiving all bids, the electricity market operator (often an energy exchange or energy management system) determines the equilibrium price, known as the clearing price, at which electricity will be traded. The clearing price is calculated based on the interplay between supply and demand for electricity, taking into account factors such as the availability of electricity resources, weather forecasts, and other market-influencing factors. Following the establishment of the clearing price, electricity is allocated to market participants based on their bid submissions. Producers who offered electricity at the clearing price receive payment based on this established price, while consumers who purchased electricity at the clearing price pay for the electricity they receive. **It is therefore strategically important for both parties (supplier and consumer) to make the right price request, in order to avoid being left out of the trade.** In some instances, market participants may also enter into bilateral contracts, which are direct agreements between buyers and sellers for the supply of electricity. These contracts can affect the volume of electricity that needs to be bought or sold in the day-ahead market.[3]

Market operators must also ensure that there is a reliable supply of electricity available to meet demand at all times. Consequently, security and capacity mechanisms may be implemented to guarantee a consistent energy supply, particularly during peak demand

[3]See the EPEX SPOT Exchange website `https://www.epexspot.com/en/basicspowermarket` (Last access 1 November 2023)

situations.

The day-ahead electricity market is an essential instrument for the efficient allocation of energy resources and price determination. As a wholesale market, it directly influence the electricity costs for end consumers. Furthermore, it plays a pivotal role in the transition to a more sustainable energy system by facilitating the integration of variable renewable energy sources such as wind and solar power, thereby contributing to improved energy management at regional and national levels (see e.g. Glachant et al. (2021)).

This is the market we are going to work with, and specifically, we will focus on forecasting the prices for the next day. It is essential for our purposes to emphasize that the submitted bids are **hour-specific**, specifying both the quantity (in MW) of electricity to trade and the price for each hour of the next day.

2.3.2. Intra-day market

The intra-day electricity market is a vital component of liberalized energy markets, focusing on the buying and selling of electricity for delivery within the next few hours or even minutes. This market is also referred to as the 'intra-day market' and is designed to offer participants greater flexibility in managing fluctuations in electricity supply and demand compared to the day-ahead market.

The intra-day market operates during the day, allowing participants to submit bids or modify their buying and selling positions in real-time or within very short time intervals (often from an hour or so up to a few minutes before actual energy delivery).

The intra-day market is particularly useful for managing unexpected fluctuations in electricity supply and demand, such as sudden changes in weather conditions, technical issues, or other unforeseen situations that can impact energy production or consumption.

As in the day-ahead market, the market operator establishes a clearing price based on current bids and demand. However, in the intra-day market, the clearing price can vary significantly in response to supply and demand fluctuations. Electricity is then allocated in real-time or within the specified time interval to the relevant parties based on their bids or transactions made in the intra-day market. The intra-day market is often used to address emergency situations, such as sudden spikes in demand or technical issues in power plants. Participants can react in real-time to ensure the continuity of supply and grid stability.[4]

[4]see the EPEX SPOT website `https://www.epexspot.com/en/basicspowermarket` (last access 22 November 2023)

The intra-day market is critical for maintaining the real-time balance between electricity production and consumption. It helps ensure that electricity is available when and where it is needed, contributing to the security and reliability of the electrical grid. Additionally, it enables participants to optimize their financial and operational positions, minimizing costs and maximizing the efficiency of electricity utilization. In the electricity-finance field, making predictions in this market is less essential, as it is primarily used for position adjustments. In contrast, in the day-ahead market, the trading volumes are much higher, and the time horizon is more extensive (see e.g. Glachant et al. (2021)).

2.4. Exchanges

In this section, we introduce the European exchanges we work on, highlighting their primary features and the most influential risk factors.

2.4.1. Nordic countries

Nord Pool is a prominent electricity market in Europe, operating as an electricity exchange with a strong emphasis on the Nordic region. This region encompasses Norway, Sweden, Denmark, Finland, and the Baltic countries.

Nord Pool is renowned for its commitment to transparency in market data.[5] It provides comprehensive information on trades, prices, and electricity availability, empowering participants to make well-informed decisions. The exchange has played a pivotal role in supporting and promoting the integration of renewable energy sources into electricity markets, contributing significantly to the transition to a more sustainable energy system. The most influential risk factors (see e.g. Liu and Wu (2007)) impacting the supply and demand dynamics in this market include:

- **Weather Conditions:** Weather patterns, particularly in regions like the Nordic ones, can have a substantial impact on energy prices. This is especially true because in many of these countries, electricity is used for heating. Wind, solar, and hydropower generation are directly affected by weather conditions. Severe winters or hot summers can alter energy demand and production.

[5] see the Nord Pool exchange website `https://www.nordpoolgroup.com/en/About-us/` (last access 22 November 2023)

- **Renewable Energy Generation:** Electricity derived from renewable sources, such as wind and solar, is inherently variable and can lead to price fluctuations. The quantity of energy generated from these sources depends heavily on prevailing weather conditions.

- **Energy Policies and Regulations:** Policy decisions and government regulations wield considerable influence over energy prices. Factors like carbon emission reduction objectives, subsidies for renewable energy, and market policies can exert a significant impact on energy costs and pricing.

- **Geopolitical Events and Natural Disasters:** Events like disruptions in gas supply, geopolitical conflicts, or natural disasters can disrupt electricity prices and create market uncertainty.

Given the objective of price forecasting, our task is to identify the key drivers that have a notable impact on supply and demand. Furthermore, we aim to pinpoint factors that, although subject to inherent randomness, remain measurable and somewhat predictable. Among the factors mentioned above, our primary focus is on renewable energy generation, particularly from wind sources, which play a significant role in the energy production of the northern countries.

2.4.2. France

EPEX SPOT France (EPEX FR) is a pivotal player in the European electricity market, with a specific focus on the French electricity market(see e.g. Graf and Wozabal (2013)). As a subsidiary of EPEX SPOT, the leading electricity spot market operator in Europe, EPEX FR is dedicated to serving the French market. It holds a vital role in shaping electricity prices in France and in fostering an efficient and liquid electricity market. Its platform affords operators the opportunity to manage their financial and operational positions, thereby enhancing the stability and reliability of the French electrical grid. In the context of risk factors, the supply in this market is less affected by weather conditions, given the predominant reliance on nuclear power plants for electricity generation. However, it's important to note that weather conditions can still have a significant impact on demand and should be taken into consideration when assessing risk factors.

2.4.3. Germany

EPEX SPOT Germany (EPEX DE) has a primary role in the European electricity market, with a focus on the German electricity market (see e.g. Graf and Wozabal (2013)). It is interconnected with other European electricity markets, facilitating cross-border trading. These interconnections enable operators to engage in electricity exchange with neighboring countries, thereby contributing to the stability and efficiency of the electrical grid. In this market, weather-related risks and renewable energy generation have a significant impact on prices. Germany generates substantial amounts of electricity from wind and solar sources, and the inherent unpredictability of weather conditions plays a fundamental role in determining supply and demand.

3 | Dataset

3.1. Overview

In this chapter, we introduce the data utilized in this project. We present the features available for each exchange and conduct a descriptive statistical analysis. Following that, we delve into an examination of data's seasonality and outline the method for its removal. Lastly, we explore the properties and characteristics of the resulting time series. The goal is to comprehensively examine the attributes of the accessible data in order to make well-informed decisions when choosing predictive models.

It's important to note that a valid benchmark dataset for electricity price forecasting should meet three key conditions (see e.g. Lago et al. (2021)):

- **Market Diversity:** It should encompass multiple markets, allowing model capabilities to be tested under various conditions. Different markets can have distinct characteristics, and a diverse dataset helps evaluate model's adaptability.
- **Sufficient Length:** Time-series should be long enough over a period of some years. This length is crucial to assess how well a model performs over different seasons, demand patterns, and market dynamics.
- **Up to Date Data:** The dataset should be recent enough to include the effects of renewable energy sources on prices. As renewable energy generation increases, its impact on electricity prices becomes more significant. Including this recent data helps evaluate a model's ability to capture these effects.

A dataset that meets these conditions is essential for developing and evaluating effective electricity price forecasting models that can account for real-world market dynamics and conditions. Based on the conditions stated, we propose three different day-ahead markets, each of which includes data time span of 6 years, with no data older than 2010. All the following analyses are conducted on the first 4 years of data (training and validation), as we use the last 2 as a test set.

3.2. Nord Pool

3.2.1. Dataset Statistics

The first dataset represents Nord Pool, the European electricity market for northern states. This dataset spans from January 1, 2013, to December 24, 2018, and includes hourly observations of day-ahead prices, forecasted demand for the following day, and forecasted wind energy generation. The dataset has been created using data freely accessible on the website of the Nordic power exchange Nord Pool.[1]

We analyze the data from 01.01.2013 to 26.12.16, with the remaining period representing our test set, which we treat as unavailable future data.

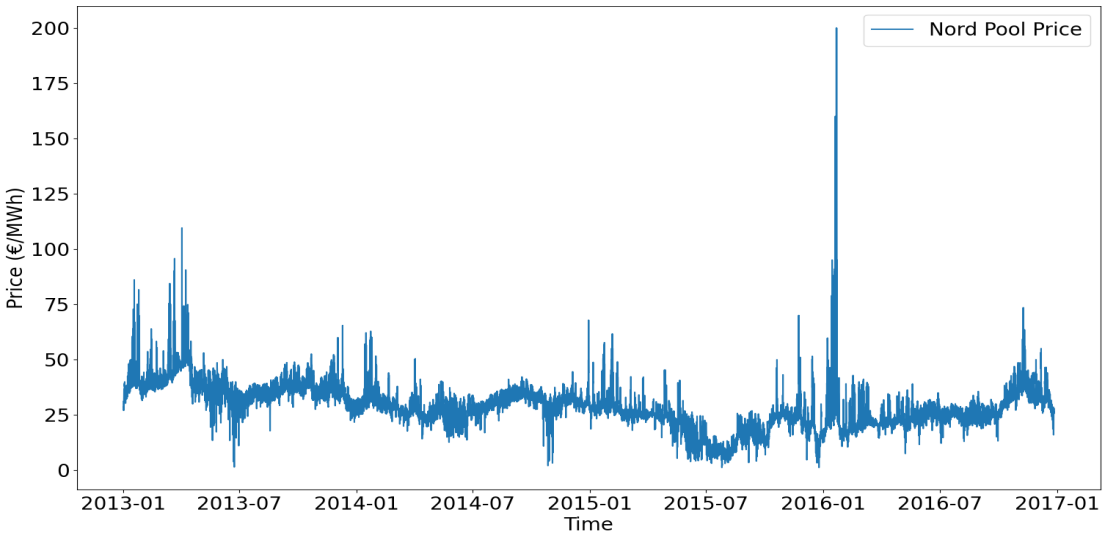

Figure 3.1: Nord Pool price time series of day-ahead prices. It is noticeable that prices are consistently positive, zero prices are rare, and price spikes occur infrequently

In Figure 3.1 it is noticeable that the data is somewhat regular.

[1] see Nord Pool exchange website `https://www.nordpoolgroup.com/en/Market-data1/#/nordic/table` (last access 22 November 2023)

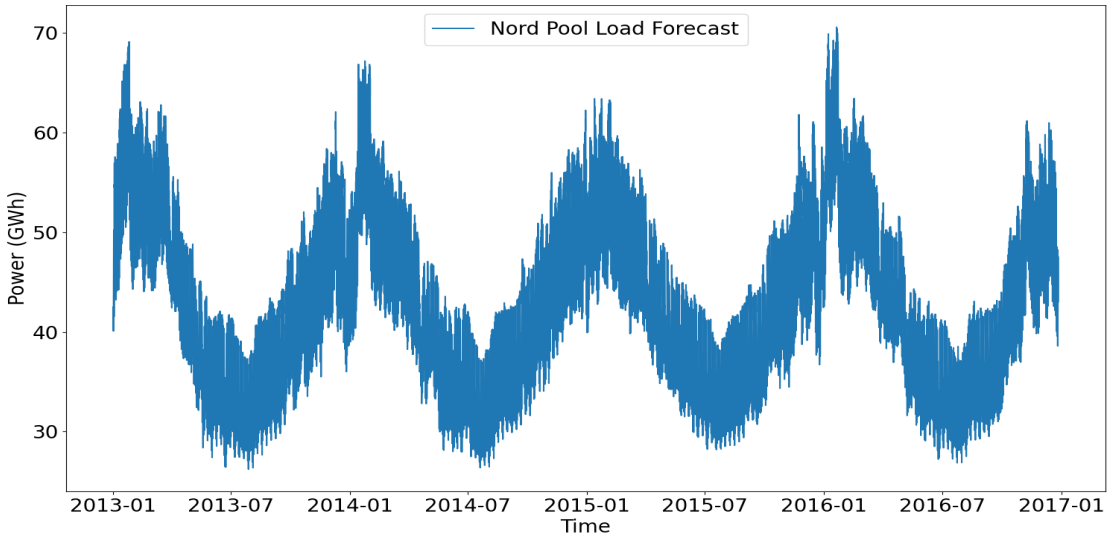

Figure 3.2: Nord Pool forecasted volume required for the following day in GWh. There is a distinct annual seasonality, with consumption peaking during the winter months and declining during the summer

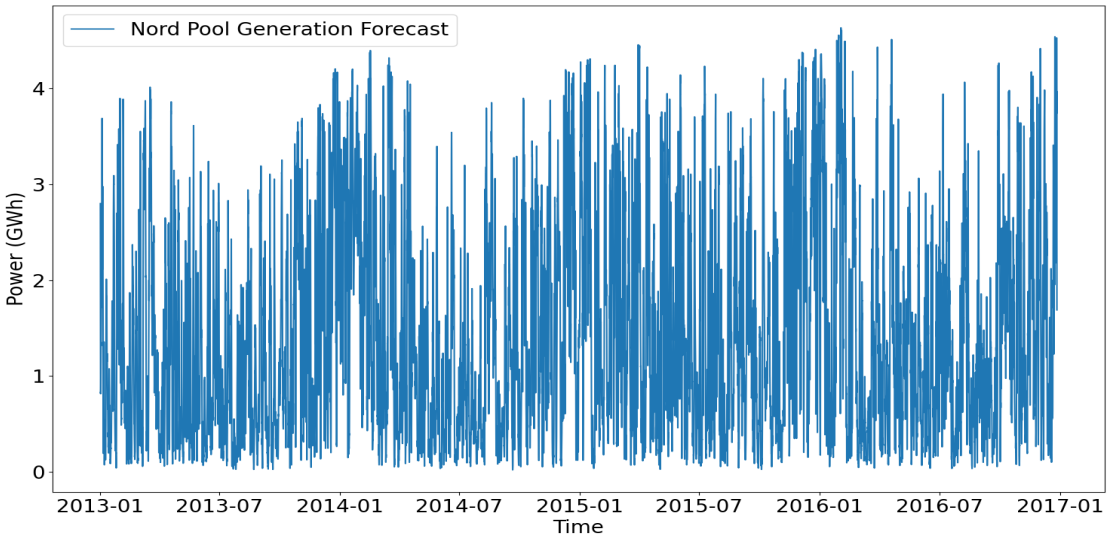

Figure 3.3: Nord Pool wind generation forecast in GWh. Also this exhibit a distinct annual seasonality, with production peaking during the winter months and declining during the summer

It's noteworthy that this seasonal feature is not readily apparent in the price chart (Figure 3.1). After all, prices are influenced by imbalances between supply and demand, and since the two profiles share a similar trend, it's not surprising to see the absence of annual seasonality in the day-ahead price time series.

Examining Figure 3.3, it becomes evident that the forecasted generation from wind sources exhibits a higher variance in comparison to the forecasted demand. This variance can be attributed to the production's dependence on weather conditions, with its behavior significantly impacting price fluctuations.

Nord Pool					
Feature	*Mean*	*Max*	*Min*	*Median*	*Std.deviation*
Price (€/MWh)	31.44	200	1.14	30.15	10.65
Load forecast (GWh)	43.85	70.58	26.2	42.7	8.69
Wind generation forecast (GWh)	1.5	5.1	0	1.2	1.14

Table 3.1: Nord Pool statistics table, obtained from the data plotted above.

In Table 3.1, we observe that the mean value of wind generation is considerably lower than that of demand, as only a small portion of the demand is satisfied by wind production. Examining the standard deviation of the production, we can see that it is nearly 100% of it's mean. This emphasizes once more how this production mechanism is profoundly affected by weather conditions and, consequently, is more susceptible to uncertainty. Regarding the price, we observe that the median is close to the mean, suggesting the presence of a modest number of outliers in the time series.

3.2.2. Autocorrelation and seasonality

Let's proceed with the analysis by studying the distribution of the time series of prices. This will provide us with additional information to make informed choices about which models to use, based on the required assumptions.

We will focus first on the autocorrelation of prices and then proceed by identifying and removing seasonality.

Definition 3.2.1
Given a time series X_t the **partial autocorrelation** of lag k, denoted $\Phi_{k,k}$, is the autocorrelation between X_t and X_{t+k} with the linear dependence of X_t on X_{t+1} through X_{t+k-1} removed.

$$\Phi_{k,k} = corr(z_{t+k} - \hat{z}_{t+k}, z_t - \hat{z}_t) \tag{3.1}$$

where $\hat{z}_{t+k}$ and $\hat{z}_t$ are linear combinations of $\{z_{t+1}, ..., z_{t+k-1}\}$ that minimize the mean square error of z_t and z_{t+k} respectively (see e.g Ramsey (1974)).

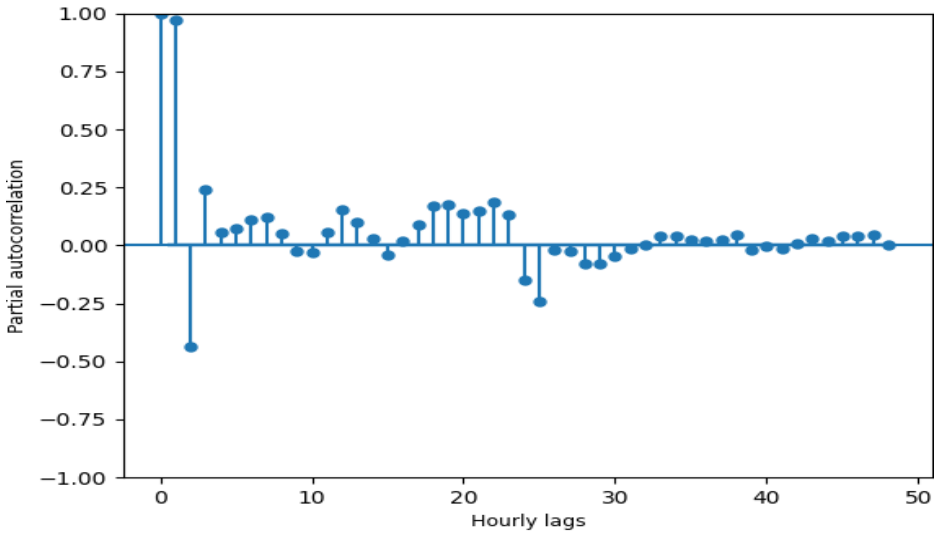

Figure 3.4: Displays the partial autocorrelation of the price time series with hourly lags up to 48 hours. It's evident that there is a higher absolute correlation for lags within 24 hours.

In Figure 3.4 we note that there is a higher absolute correlation for lags within 24 hours, suggesting a dependence between prices throughout the day. Especially noteworthy is the gap between the autocorrelation and the 95^{th} confidence bound for correlation significance, which it is too thin to see in the graph above.

The autocorrelation profile in Figure 3.4 suggests that the times series is non stationary. We test this by employing the Augmented Dickey Fuller test (see e.g. Mushtaq (2011)). The null hypothesis of this test evaluates the existence of a unit root in the time series. We obtain a p-value of 0.07, suggesting that the time series is affected by some dependence and is a non-stationary time series.

We aim to capture this dependence by examining the seasonal behavior of prices. In particular, we look at the 28 days average trend of the signal.

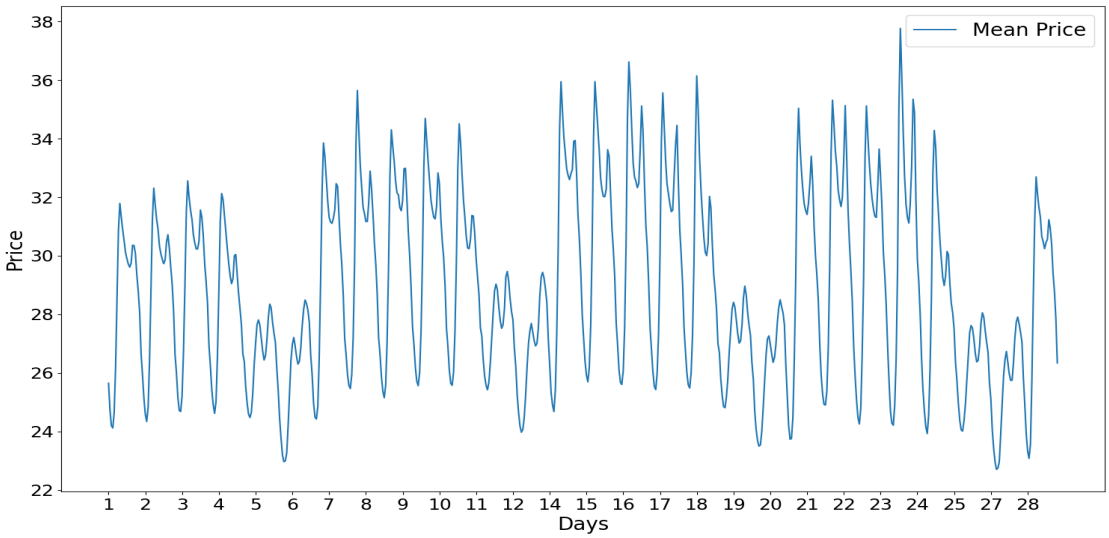

Figure 3.5: Displays the Nord Pool 28 days mean price built from the price time series by averaging observations with a lag of 28 days. It exhibits two prominent seasonal patterns, one on a daily basis and another on a weekly basis.

In Figure 3.5 the values are plotted hourly and the values on the x-axis underpin the day of the month. This process helps capture and visualize the monthly average trend in the data. We can observe two distinct seasonal components: a daily and a weekly one. The daily seasonality involves higher prices during the day, roughly from 8:00 AM to 5:00 PM, and lower prices in the evening and at night. In contrast, the weekly seasonality demonstrates higher prices during weekdays and lower prices on weekends.

When discussing seasonality, it's often beneficial to determine the seasonal period, as various effective methods for removing this component are available when the period is known. Obtaining a time series without seasonality can be advantageous, as it approximates an uncorrelated series that resembles white noise. This deseasonalized series allows for the application of a set of models whose assumptions might not hold when applied to the original price time series. To obtain the deseasonalized time series we apply a technique called **sesonal differencing** (see e.g. Li (1991)).

Definition 3.2.3
The **seasonal differencing** of a time series X_t in discrete time t given the seasonality's period d is the transformation of the series to a new time series S_t where the values are the differences between the value of X_t at time t and the the value of X_t a period d before.

$$S_t = X_t - X_{t-d} \tag{3.2}$$

We apply this technique with a period d of 24 hours to the price time series.

We visualize this in Figure 3.6 where the values are plotted hourly, and the x-axis values correspond to the days of the month. The daily periodic trend appears to have vanished, leaving only the weekly trend, with peaks noticeable during the weekends.

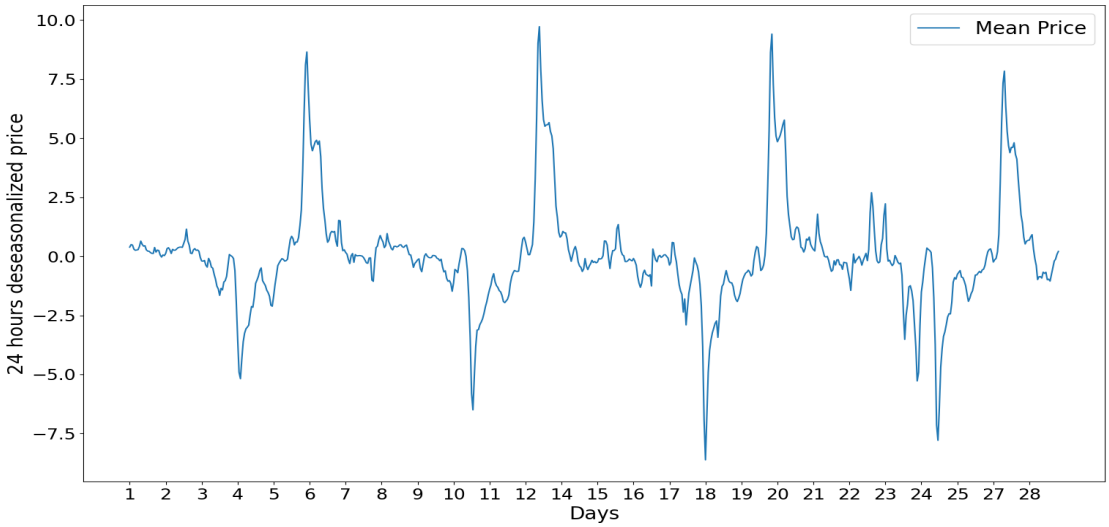

Figure 3.6: Nord Pool 28 days mean price after having removed the daily seasonality. We observe only a weekly periodic trend.

The analysis proceeds by eliminating the weekly periodicity from the signal we've recently acquired, employing seasonal differencing with a 7-day period. The resulting time series comprises the unexplained correlation component, which we have yet to identify and remove, as well as the random component that we must use for training our forecasting models.

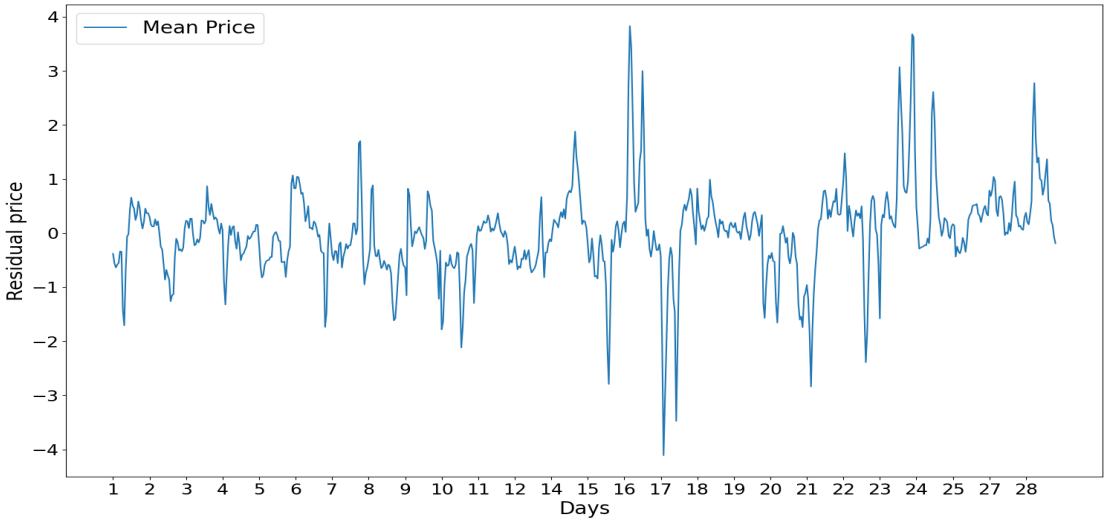

Figure 3.7: Nord Pool 28 days mean deseasonalized price. It seems to fulfill our criteria of obtaining a less autocorrelated time series that resembles white noise.

Even if visually Figure 3.7 presents a less correlated signal, we still aim to test our hypothesis by quantifying the seasonal influence of the original signal in comparison to the one obtained after eliminating seasonality.

We quantify the existence of periodic behaviors in the time series by analyzing its frequency spectrum. Specifically, we compute the discrete Fourier transform of the signal, which decomposes it into a representation that characterizes the frequencies within the original series. We subsequently assess the phase and amplitude distribution of each frequency component in relation to frequency. A spectrum displaying peaks at specific frequencies indicates that the original signal primarily consists of a periodic function at those frequencies, signifying seasonality. Conversely, a time series with no peaks in its frequency domain indicates a signal devoid of predominant periodic effects that more closely resembles a series of independent observations.

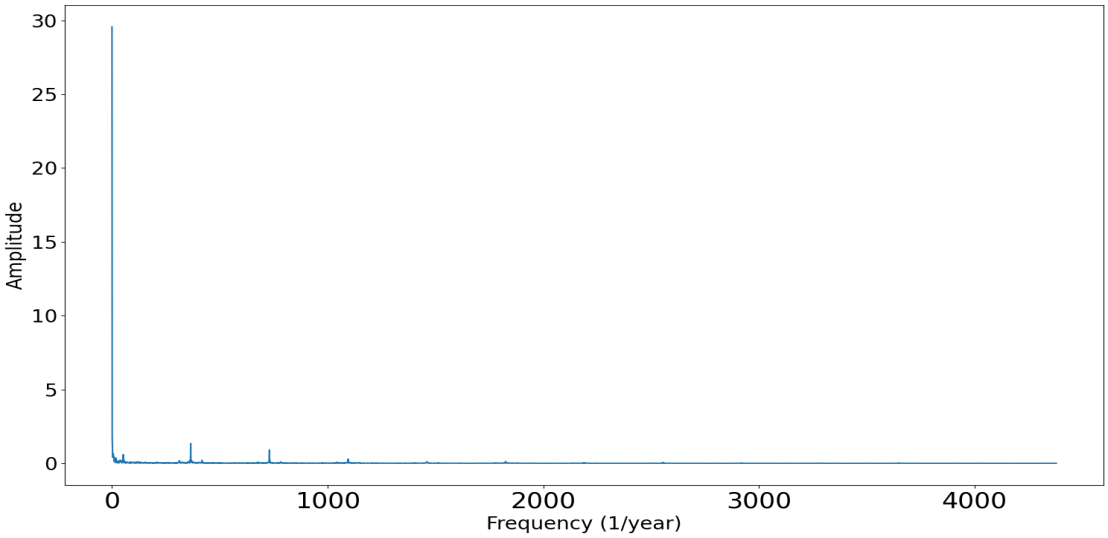

Figure 3.8: Nord Pool yearly mean price frequency domain, displays the frequency spectrum of the original price signal, the frequency is intended as 1/8670 hours, or equivalently 1/year

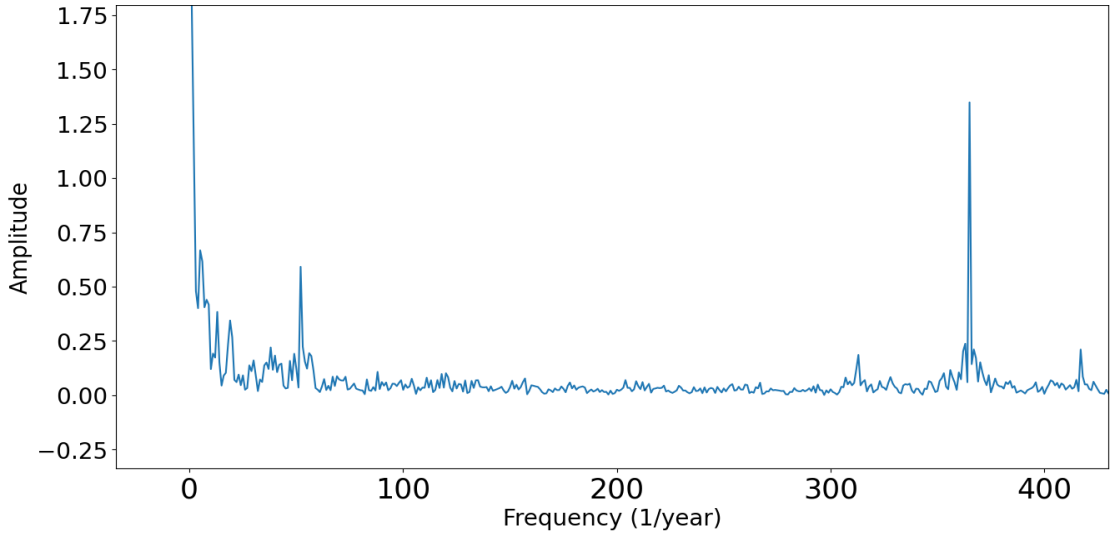

Figure 3.9: Displays a zoom on the origin of the frequency spectrum of the original price signal. We observe a trend, and a weekly and daily periodic patterns.

Now, let's examine and provide comments on the three significant amplitude peaks of Figure 3.9. The first and most prominent peak corresponds to the presence of a periodic function with zero annual cycles, which signifies the overall trend of the signal. The second peak, found at a frequency of 52, signifies a periodic function with 52 annual cycles, equating to 1 cycle per week; this is indicative of the contribution of weekly seasonality. The third peak, situated at a frequency of 365, denotes a periodic function with 365 annual cycles, reflecting the contribution of daily seasonality. Let's now observe how the frequency spectrum is affected when we eliminate the daily seasonality.

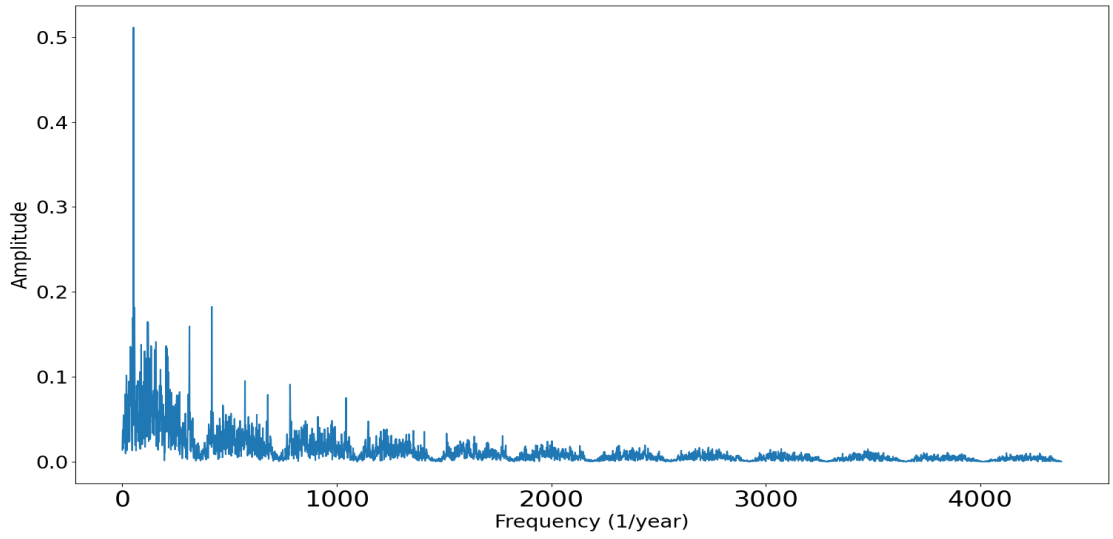

Figure 3.10: Frequency domain of the Nord Pool yearly mean prices daily seasonally adjusted. The contribution of the trend and daily periodicity has disappeared.

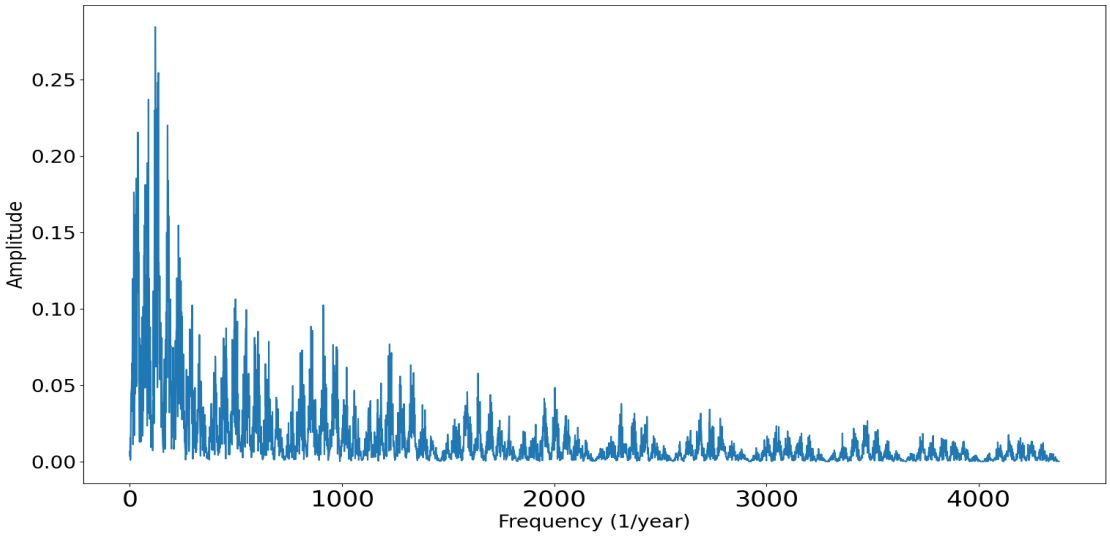

Figure 3.11: Frequency domain of the Nord Pool yearly mean prices after having removed both daily and weekly seasonal components. It shows a composition of sinusoids at various frequencies with similar amplitudes concentrated on small cycles.

The remaining amplitudes in Figure 3.11 indicate a composition of sinusoids at various frequencies with similar amplitudes. We have eliminated the most pronounced seasonal components. The residual components imply seasonality with a period longer than weekly (possibly monthly, quarterly, or annually), although they have a lesser impact.

We display the partial autocorrelation function for the deseasonalized prices and compare it with the one of the original price time series.

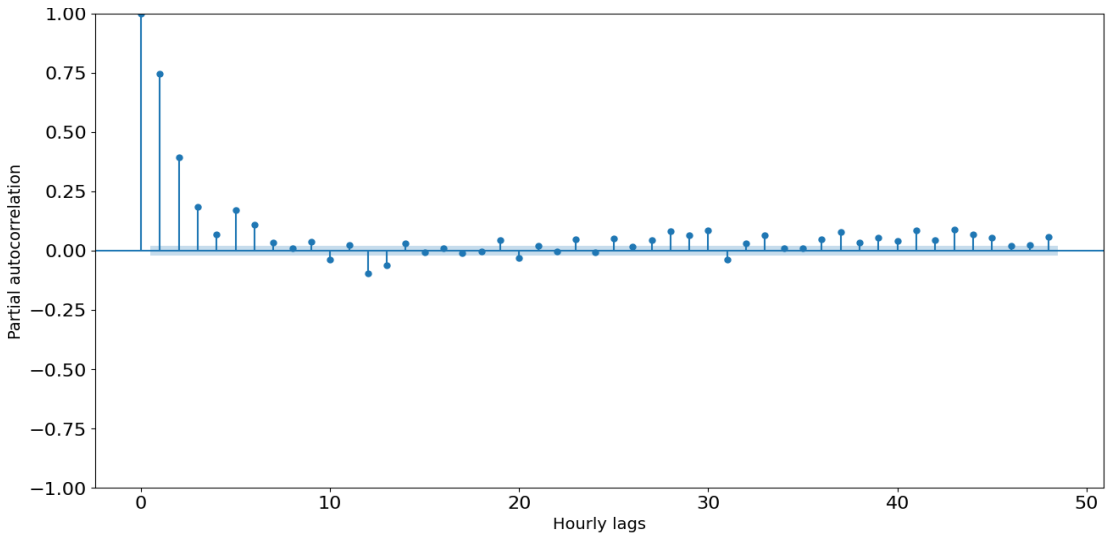

Figure 3.12: Partial autocorrelation of deseasonalized yearly mean prices, we observe some short lag correlation.

Compared to Figure 3.4, we observe that the correlation in Figure 3.12, in general, is lower. Some correlation remains between observations spaced 1 and 2 hours apart.

In concluding this subsection, we test the seasonally adjusted time series for stationarity using the Augmented Dickey-Fuller (ADF) test (see e.g. Mushtaq (2011)). The resulting p-value, approximately of the order of 10^{-8}, provides compelling evidence to reject the null hypothesis. Consequently, we can confidently consider the new time series as stationary.

3.3. EPEX France and EPEX Germany

Regarding these other two markets, the analysis performed is equivalent to the one presented above for Nord Pool, and in the current section, we highlight only the differences obtained compared to the study on Nord Pool. The data for EPEX DE spans from 09-01-2012 to 31-12-2017, and the descriptive statistics is for the period until 03-01-2016 (training and validation). EPEX FR, on the other hand, covers the period from 09-01-2011 to 31-12-2016, and we consider the data until 03-01-2015.

3.3.1. Dataset

The first difference lies in the selection of the feature that represents supply uncertainty, specifically the amount of generated electricity. For EPEX FR, this feature is the forecasted electricity generation from any source. This choice aligns with the characteristics of the French energy system, primarily reliant on nuclear energy, resulting in lower uncertainty in the quantity generated. In contrast, for EPEX DE, the feature is the forecasted generation from solar and wind sources, which is subject to uncertainty and have an impact on electricity prices.

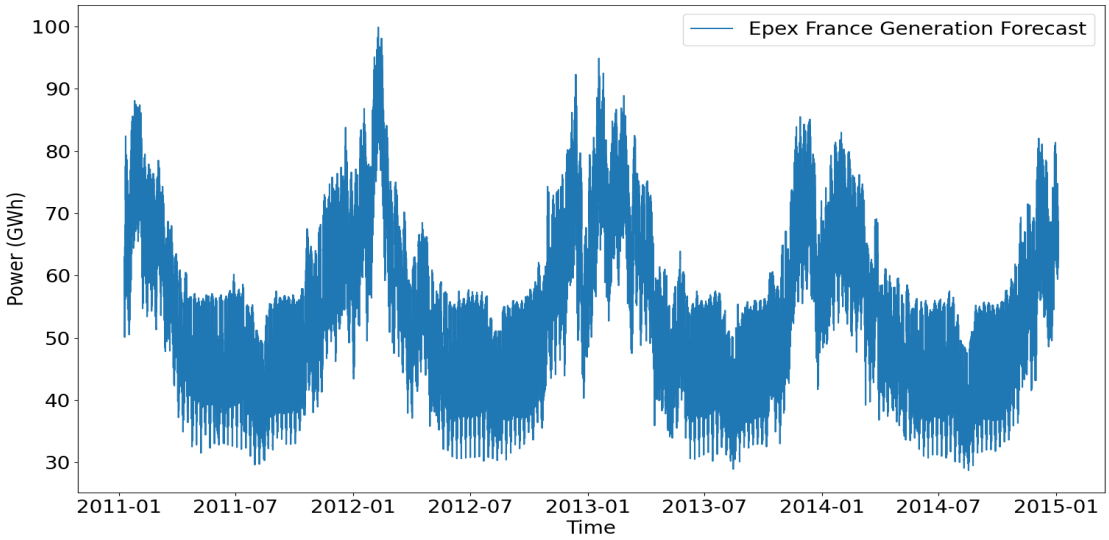

Figure 3.13: forecasted generation for EPEX FR in GWh, the generation profile has low variance because it comprises the total forecasted generation. Profiles exhibits annual seasonality.

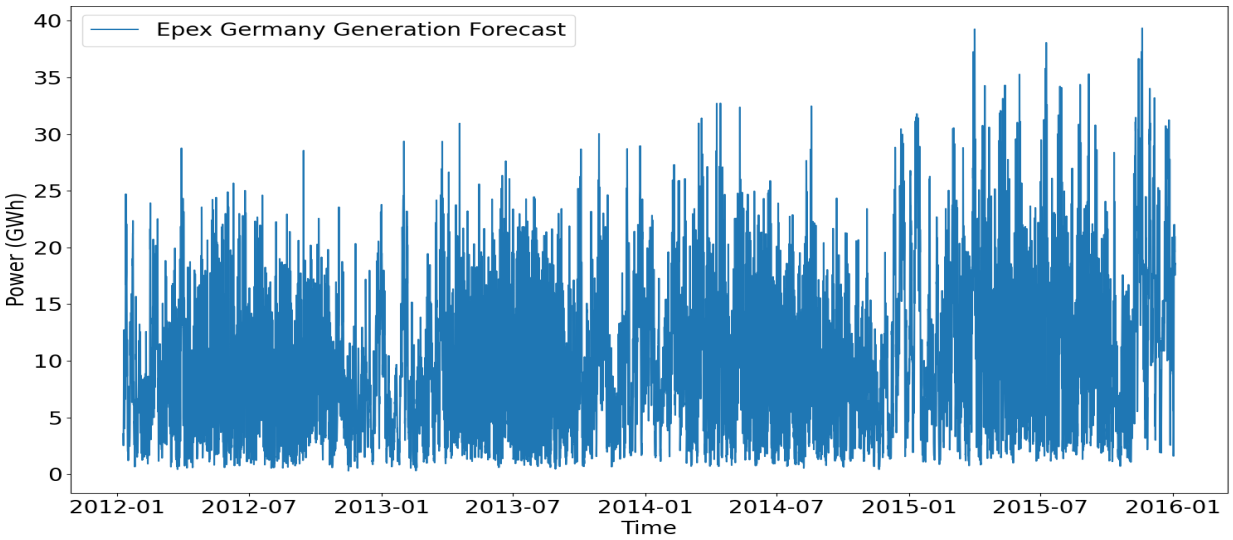

Figure 3.14: forecasted generation for EPEX DE in GWh. A high-variance profile is expected since solar and wind generation are subject to the uncertainty of weather conditions. Profiles exhibits also annual seasonality in supply.

Figure 3.13 and Figure 3.14 exhibit similar features in the Nord Pool profile (Figure 3.2), although the power magnitude is considerably larger in the case of EPEX France, and the profile displays less variance. However, for EPEX DE, we observe an unclear yearly seasonal pattern with very high variance.

We show now the price signals.

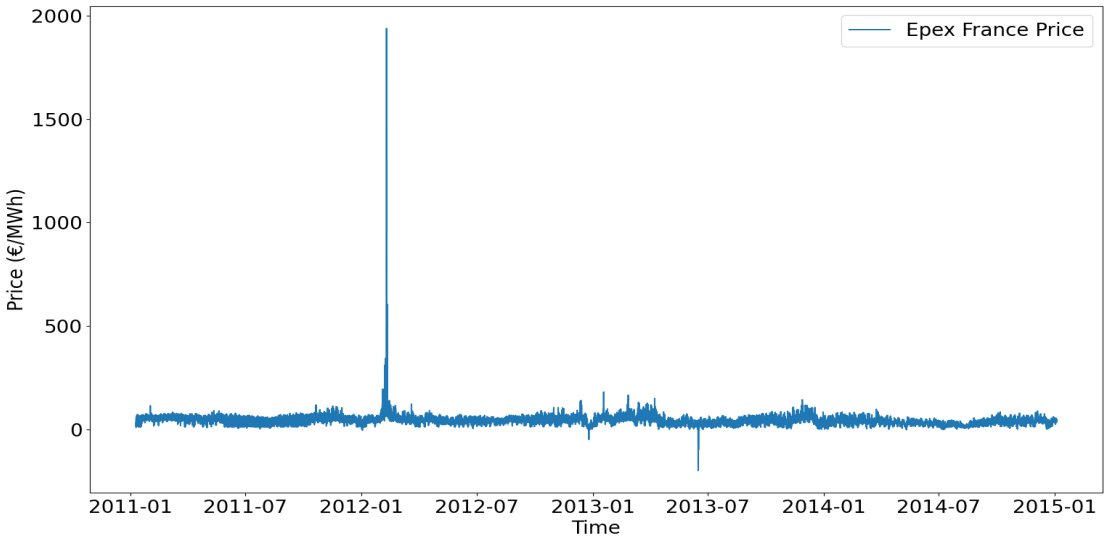

Figure 3.15: Price profile for EPEX FR, it includes negative prices and has strong outliers, although they are relatively few

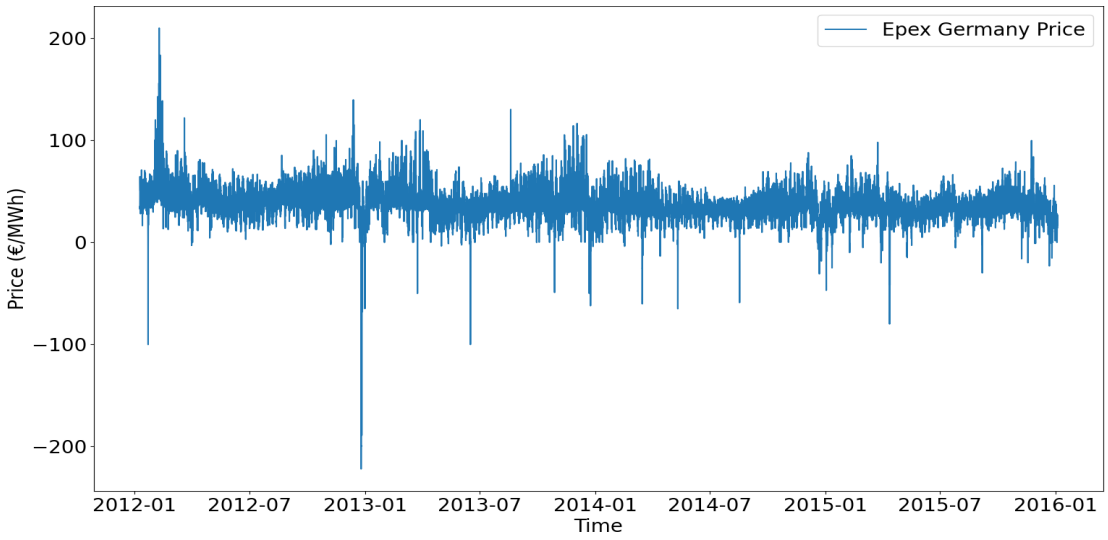

Figure 3.16: price profile of EPEX DE features many negative prices and frequent outliers, although they are less pronounced than those in EPEX FR

The price profiles also differ from those of Nord Pool. Looking at Figure 3.15, we can observe that the Price profile for EPEX FR includes negative prices and has strong outliers.

In Figure 3.16, we notice that the price profile of EPEX DE features many negative prices and frequent outliers, although they are less pronounced than those in EPEX FR.

We can highlight these differences more effectively with a descriptive table of the two datasets.

EPEX FR					
Feature	*Mean*	*Max*	*Min*	*Median*	*Std.deviation*
Price (€/MWh)	41.45	1938	-200	41	23.1
Load forecast (GWh)	59.47	93.52	33.15	58.11	10.45
Generation forecast (GWh)	54.17	99.9	28.7	52.48	12.1

Table 3.2: EPEX France statistics table

Table 3.2 displays a set of descriptive statistics for the French electricity market. We immediately notice that the min-max range exceeds 2000 Euros, with an average price of 41. The standard deviation of the price surpasses 50% of the mean, indicating a high-variance time series. Additionally, we observe that, unlike Table 3.1, the order of magnitude of the forecasted generation is comparable to that of the demand.

EPEX DE					
Feature	*Mean*	*Max*	*Min*	*Median*	*Std.deviation*
Price (€/MWh)	34.68	210	-222	33.45	15.94
Load forecast (GWh)	21.36	35.5	10.72	21.29	3.74
Generation forecast (GWh)	10.71	48.59	0.3	8.9	7.64

Table 3.3: EPEX Germany statistics table

Table 3.3, on the other hand, provides key figures for EPEX DE. Here, we also see a high variance in prices. The generation covers approximately 50% of the demand, reflecting Germany's substantial utilization of solar and wind energy.

We have effectively summarized the main differences between the datasets, the autocorrelation analysis, seasonality removal, and frequency domain study yielded results equivalent to those of Nord Pool, and they are available upon request.

3.4. Chapter summary

In this chapter, we have examined the main characteristics of our data, and for each exchange, we have the tools to introduce and test forecasting models. Specifically, we have the time series of prices, affected by autocorrelation and seasonality, and the deseasonalized price time series, which are less autocorrelated and better represent the stochastic component of the price. We are going to explore the effects of training different forecasting models with conformal prediction intervals on both the price time series and the seasonally adjusted one.

4 | Linear models

Most forecasting studies in the power market primarily concentrate on price forecasting (see, e.g., Uniejewski et al. (2016)). However, in this work, we concentrate our efforts on short-term price forecasting with a specific emphasis on the ability to predict uncertainty, specifically through a conformal predictor. Notably, the literature on Electricity Price Forecasting (EPF) that delves into the validity and efficiency of prediction intervals is somewhat limited.

The day-ahead price characteristics we aim to model encompass several aspects. On one hand, we seek to capture the short-term trends and seasonal patterns, including both daily and weekly variations. On the other hand, we are interested in understanding daily autocorrelation within the prices and exploring the relationship between day-ahead prices, load forecasts, and generation forecasts.

In this chapter, we introduce two linear models that serve as a reference for comparing the performance of a more complex model. For each method, we provide details about the modeling process, hyperparameter selection, and the methodology applied to the specific dataset.

Each dataset comprises 6 years of observations, which we will divide as follows:

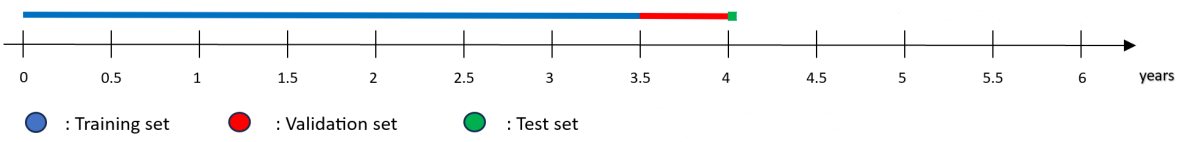

Figure 4.1: Dataset segmentation, the test set is the day that follows the last one on the validation set. We use 3.5 years of training set, 6 months of validation and 1 day of test.

The models are calibrated on the training set, which includes the first 3 and a half years of data. The following 6 months are used as the validation set, and the subsequent day is used as the test set. After choosing, training, and testing a model on the test set's date, we add the first day of the validation set to the training set and the test set's date to the validation set. By doing so, we train a model for each day of the test set, iteratively

increasing the size of the training set.

It's important to emphasize that these benchmark models use the validation set to optimize their hyperparameters. After selecting the model based on its performance on the validation set, but before testing it, we recalibrate it on the dataset that includes both the training set and the validation set.

The data is preprocessed using the inverse hyperbolic sine variance stabilizing transformation (see e.g. Uniejewski et al. (2018)). The transformation is $asinh(x) = log(x + \sqrt{x^2 + 1})$, where x denotes the price, and it is standardized by subtracting the median calculated from the in-sample data. Then, this value is divided by the median absolute deviation (MAD), which is adjusted by a factor to ensure asymptotically normal consistency with the standard deviation.

$$y = \frac{1}{b}(asinh(x) - a) \tag{4.1}$$

In equation (4.1), y represents the transformed price used to train the models, a is the median of $asinh(x)$ from the in-sample data, and b is the median absolute deviation of $asinh(x)$ divided by $z_{0.75}$, which is the 75^{th} quantile of the normal standard distribution. We choose this type of data scaling because it is well defined for negative values, smooth around zero, and is often preferred when dealing with spiky electricity prices, as highlighted in the paper of Uniejewski et al. (2018).

4.1. Notation

Let's introduce the notation used in this chapter.

- $p_{d,h}$ represents the price on day d at hour h.
- $l_{d,h}$ represents the load forecast on day d at hour h.
- $g_{d,h}$ represents the generation forecast on day d at hour h.
- $\mathbf{p}_d$, $\mathbf{l}_d$, $\mathbf{g}_d$ represent the vector of values for every hour of day d
- $\mathbf{D}$ is a dummy vector representing the day of the week. It's a vector of dimension 7 where all values are zero, except for the index that represents the day of the week, which is filled with a value of one. If the day is a holiday, the vector consists of all zero values.
- p_d^{min} is the minimum price observed on the hours of day d.
- p_d^{max} is the maximum price observed on the hours of day d.

- p_d^{mean} is the mean price observed on the hours of day d.

4.2. Full Autoregressive Model

Here, we delve into the full autoregressive model (see, e.g. Uniejewski et al. (2016)), with our goal being to predict $p_{d,h}$ given a set of features observable in $\mathbf{p}_{d-1}$.

$$\begin{aligned} p_{d,h} = {} & \boldsymbol{\beta}_1 \mathbf{p}_{d-1} + \boldsymbol{\beta}_2 \mathbf{p}_{d-2} + \boldsymbol{\beta}_3 \mathbf{p}_{d-3} + \beta_{4,h} p_{d-7,h} \\ & + \sum_{j=1}^{3} (\beta_{4+j,h} p_{d-j}^{min} + \beta_{7+j,h} p_{d-j}^{max} + \beta_{10+j,h} p_{d-j}^{mean}) \\ & + \beta_{14,h} l_{d,h} + \beta_{15,h} l_{d-1,h} + \beta_{16,h} l_{d-7,h} + \beta_{17,h} g_{d,h} \\ & + \sum_{j=1}^{7} (\beta_{17+j,h} D_j + \beta_{24+j,h} D_j l_{d,h} + \beta_{31+j,h} D_j p_{d-1,h}) + \epsilon_{d,h} \end{aligned} \tag{4.2}$$

Where $\boldsymbol{\beta}$ is a vector of dimension 24 representing regression coefficients for each hour, and $\epsilon_{d,h}$ are assumed to be independent and identically distributed Gaussian variables.

We have a total of 107 regressors.

4.2.1. Selection and shrinkage procedures

In this section, we present the methodology for both tuning the hyperparameters and train the model. Since our goal is to forecast the day-ahead price for a specific hour, and the set of features used varies with the hour of the day, we optimize the hyperparameters for each individual hour. This process leads to obtaining 24 distinct models for each new day in the test set.

Specifically, the set of potential models includes five different loss functions with a regularization parameter. Therefore, our hyperparameter selection include the selected loss function and its corresponding regularization parameter.

The loss functions we focus on are listed below:

- **Lasso Regression**: is a regularization method similar to least squares, except that the coefficients are estimated by minimizing the penalized residual sum of squares using a module factor for coefficients that are too large in magnitude. It is a commonly used method in EPF literature as it can estimate the coefficients of

non-influential regressors as zeros, essentially performing feature selection.

$$\hat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{d,h \in T} \left(p_{d,h} - \sum_{i=1}^{107} \beta_{h,i} X_{h,i} \right)^2 + \lambda \sum_{i=1}^{107} |\beta_{h,i}| \right\} \tag{4.3}$$

Where $X_{h,i}$ represent the i-th feature used for training the model to forecast $p_{d,h}$

- **Ridge Regression**: It is similar to lasso but enforces penalization using a quadratic factor. This method is not widely analyzed in EPF literature, mainly because it doesn't zero out the coefficients of non-influential regressors. We have included it nonetheless, and if it doesn't significantly improve performance, we won't select it for the forecasting procedure.

$$\hat{\boldsymbol{\beta}}^{ridge} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{d,h \in T} \left(p_{d,h} - \sum_{i=1}^{107} \beta_{h,i} X_{h,i} \right)^2 + \lambda \sum_{i=1}^{107} \beta_{h,i}^2 \right\} \tag{4.4}$$

- **Elastic Net**: It is a combination of both Lasso and Ridge. This loss function is also not commonly used in the EPF context, primarily for the same reason as Ridge.

$$\hat{\boldsymbol{\beta}}^{EN_\alpha} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{d,h \in T} \left(p_{d,h} - \sum_{i=1}^{107} \beta_{h,i} X_{h,i} \right)^2 + \alpha\lambda \sum_{i=1}^{107} |\beta_{h,i}| + (1-\alpha)\lambda \sum_{i=1}^{107} \beta_{h,i}^2 \right\} \tag{4.5}$$

The five possible loss function to choose for forecasting $p_{d,h}$ are *Lasso*, *Ridge*, $EN_{0.25}$, $EN_{0.50}$, $EN_{0.75}$ (respectively equal to $\alpha = 1, 0, 0.25, 0.5, 0.75$).

The first step is to find the best λ for each loss function, and we do this through leave-one-out cross-validation on the training set (see e.g. Wong (2015)). We set up a λ grid, specifically 20 possible values that span orders of magnitude from 10^{-9} to 100. To assess the performance of a specific λ, we train the model by minimizing the respective loss function on the entire training set, excluding a single observation, and then evaluate it's error in terms of squared distance by making it predicting the omitted observation. We iterate this process for each observation in the training set and evaluate the mean of the errors as a metric to assess the impact of that λ. We choose the one which produces the lower mean square error.

In this way, we have found the best parameters for the 5 models, and we choose one based on their performance on the validation set. Specifically, we use the 5 models to predict the observations on the validation set, and for each model, we calculate the mean absolute error.

$$MAE = \frac{1}{|T|} \sum_{d,h \in T} (|p_{d,h} - \hat{p}_{d,h}|) \tag{4.6}$$

Where T is the timeline of the validation set and $\hat{p}_{d,h}$ is the price forecasted on the validation set. At this point we select the model with the lowest error metric, and we train it again on the concatenation of both training and validation set.

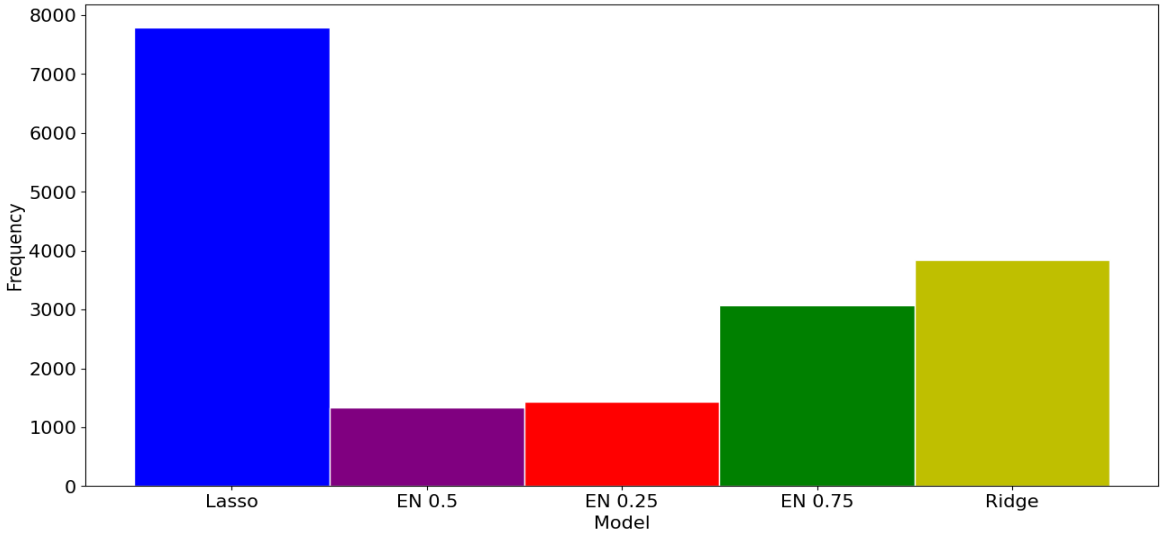

Figure 4.2: Frequency at which each loss function has been chosen training the model on Nord Pool prices, we can observe a higher frequency for the Lasso model.

In figure Figure 4.2, a histogram shows the frequency at which each loss function has been chosen, we can observe a higher frequency for the Lasso model.

The optimization method to find the minimum of the loss function is the coordinate descend method (see e.g. Wright (2015)).

4.3. Lasso estimated AutoRegressive

This model was originally presented as *LassoX* (see, e.g. Uniejewski et al. (2016)). This parameter-rich linear model closely resembles the autoregressive model previously described but comes with distinct characteristics and capabilities. The LEAR model for predicting $p_{d,h}$ is defined as follows:

$$\begin{aligned} p_{d,h} =& \boldsymbol{\beta}_1 \mathbf{p}_{d-1} + \boldsymbol{\beta}_2 \mathbf{p}_{d-2} + \boldsymbol{\beta}_3 \mathbf{p}_{d-3} + \boldsymbol{\beta}_4 \mathbf{p}_{d-7} \\ &+ \boldsymbol{\beta}_5 \mathbf{l}_d + \boldsymbol{\beta}_6 \mathbf{l}_{d-1} + \boldsymbol{\beta}_7 \mathbf{l}_{d-7} + \boldsymbol{\beta}_8 \mathbf{g}_d + \boldsymbol{\beta}_9 \mathbf{g}_{d-1} \\ &+ \boldsymbol{\beta}_{10} \mathbf{g}_{d-7} + \sum_{j=1}^{7} (\beta_{10+j} D_j) + \epsilon_{d,h} \end{aligned} \tag{4.7}$$

The first difference from equation (4.2) is that the features are specific to the day but not to the hour. Consequently, we use the same training set to predict every hour of day d. The model includes 247 regressors, more than double the number in the full autoregressive model. In this case, a robust feature selection process is mandatory due to the large number of parameters.

4.3.1. Selection and shrinkage procedures

While the training set is applicable for all hours of a specific day, we still opt for a distinct model to forecast a specific hour of the following day. This decision aligns with our goal to use these models as benchmarks, ensuring their complexity remains consistent. As previously noted, this model possesses a wealth of parameters and demands a strong feature selection process. Hence, we choose to estimate the parameters using L1 regularization, i.e Lasso, which has the capability to nullify coefficients associated with unimportant regressors.

Returning to the model in equation (4.2) and recalling our methodology for hyperparameter selection, we observe that the full autoregressive model optimizes the parameters of 5 loss functions, selecting them from a grid of 20 possible values that vary in magnitude. This process involves calibrating a total of 100 models for each hour. Here, since we decide to utilize only L1 regularization, to maintain complexity comparability we can search for the lambda hyperparameter on a larger grid.

The option that seems most promising involves calibrating the lambda parameter through cross-validation, using a grid equally spaced from 10^{-9} to 100 with at least 1000 values. While this method yields good results, it comes with a high computational cost. Further-

more, this strategy would result in a model with higher complexity compared to the full *autoregressive* model (equation (4.2)). The solution we adopt is to change the optimization method to reach the minimum of the loss function. Instead of relying on coordinate gradient descent, we optimize using least angle regression (see e.g. Efron et al. (2004)), which is computationally comparable to the least squares solution when the number of covariates is the same. With a grid of specific λ, we assess the performance of a λ through leave-one-out cross-validation on the training set using the AIC information criterion (see e.g. Bozdogan (1987)) as metric for comparisons, which introduces a penalty parameter for overly complex models (in this case, for a high number of covariates), thus providing additional protection against overfitting.

Like previously, after selecting the best regularization parameter for the Lasso loss, we concatenate the training set and the validation set to retrain the model, this time using the coordinate descend optimization method (see e.g. Wright (2015)).

4.4. Chapter summary

In this chapter, we have introduced two linear models: the full autoregressive model (fARX) and the Lasso estimated autoregressive model (LEAR). Since our goal is to explore the utilization of conformal prediction intervals, we will test the results to draw valid conclusions, using both simple models that employ straightforward hyperparameter tuning techniques and more complex models that we will introduce in the next chapter.

5 | Deep neural network tuned with a bayesian optimizator

The use of deep neural networks in the field of electricity price forecasting is quite common (see e.g Kuo and Huang (2018) or Ugurlu et al. (2018)). It is indeed known that this model is the best when the goal is to obtain accurate point predictions. However, due to its non-linear structure, obtaining reliable predictions regarding uncertainty in the forecast remains a fertile research ground. Furthermore, most studies seem to propose fixed neural structures, and utilizing a method for generalizing hyperparameter tuning can be interesting for our purposes.

This model (see e.g. Lago et al. (2021)), in particular, utilizes a Bayesian optimization technique called tree-structured Parzen estimator (see e.g Watanabe (2023)), which, upon convergence, provides the optimal structure of the deep neural network for the specific dataset.

We must begin by redefining the data timeline, how we split the data into training, validation, and test sets, and the number of distinct models employed for each date in the test set.

- **Hyperparameters tuning**: It's the initial phase for model tuning, which involves selecting the hyperparameters of the deep neural network and the relevant features to consider. This process is carried out using data spanning the initial four years, while the last two years are reserved for our test set and are not considered in the hyperparameter optimization.

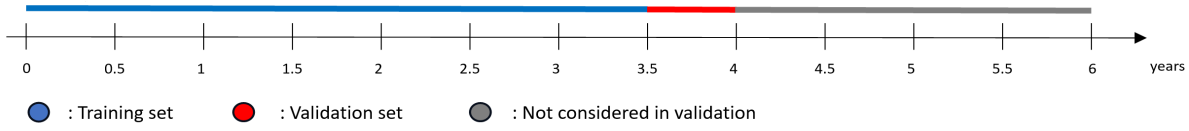

Figure 5.1: Dataset segmentation for tuning the deep neural network hyperparameters.

In Figure 5.1, the dataset segmentation for tuning the deep neural network hyperparameters, we consider the first 3 years of data as training set and the fourth one as validation set, the rest of the data is considered as unobservable future data.

- **Model training and testing**: Once the tuning phase has converged, resulting in the optimal hyperparameters and feature selection for the dataset in question, we obain a set of hyperparameters which defines a fixed DNN model that remains constant throughout the subsequent steps. At this point, we follow a similar data splitting strategy to the one used for the benchmark models. We designate the first 3 years of data as the training set and the fourth year as the validation set. The validation set's purpose in this phase is to facilitate early stopping, which triggers if the performance on the validation set doesn't improve for 40 epochs in a row.

  Once the model is trained, we evaluate it on the first available date in the test set. Subsequently, we incorporate the first date of the validation set into the training set and the date from the test set into the validation set. By applying this iterative approach, we incrementally expand the size of the training set while continuously considering the most recent data.

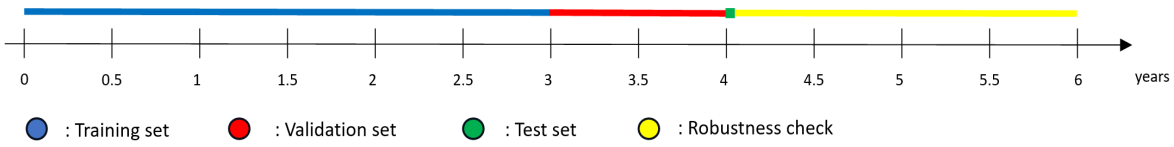

Figure 5.2: Dataset segmentation for training and testing the model

In Figure 5.2 we display the data segmentation for training and testing the tuned model. We consider the first 3 years of data as training set and the fourth one as validation set, the test set is the first available date after the validation set and the rest of dates will serve as robustness checks.

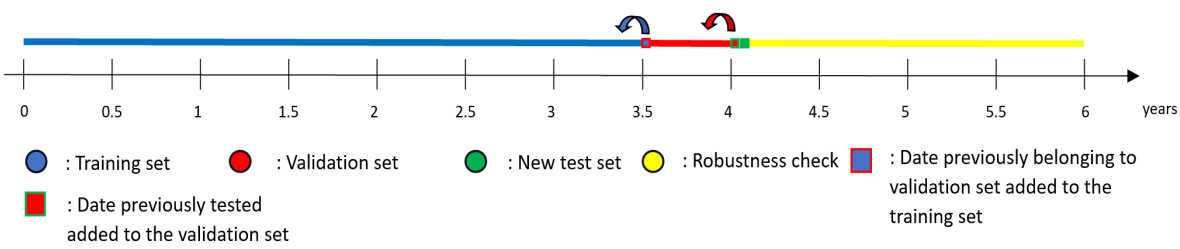

Figure 5.3: Dataset segmentation process after having tested a date

In Figure Figure 5.3, we illustrate the splitting process at the next iteration. The

first date of the validation set is merged with the training set and the date previously tested is added to the validation set.

5.1. Tree-structured Parzen Estimator

In this section, we delve into the Bayesian optimization technique considered for tuning the hyperparameters of the deep neural network. This method (see e.g Watanabe (2023)) has been used in the EPF context for the first time in the paper of Lago et al. (2021).

5.1.1. Hyperparameters space

As we have already pointed out, the optimization method selects the best hyperparameters and features for the model. Therefore, we need to provide it with the space of possible hyperparameters from which each random draw represents a neural network with its corresponding features to include. Only two aspects of the neural network remain constant: the number of layers, which is 2, and the number of epochs, set at 1000. The choice of the number of layers was based on the work of Lago et al. (2021), where structures with 3 or 4 layers were tested. However, it was observed that the model's performance did not improve significantly enough to justify the use of greater complexity.

In Figure 5.4, we illustrate a visual representation of the hyperparameters space, where each attribute corresponds to a hyperparameter of the DNN. Additionally, we introduce the "feature selection" attribute, which contains the potential features to consider. Every attribute shown in Figure 5.4 is essentially a probability distribution from which we can sample to obtain the hyperparameter value, we list those distributions below.

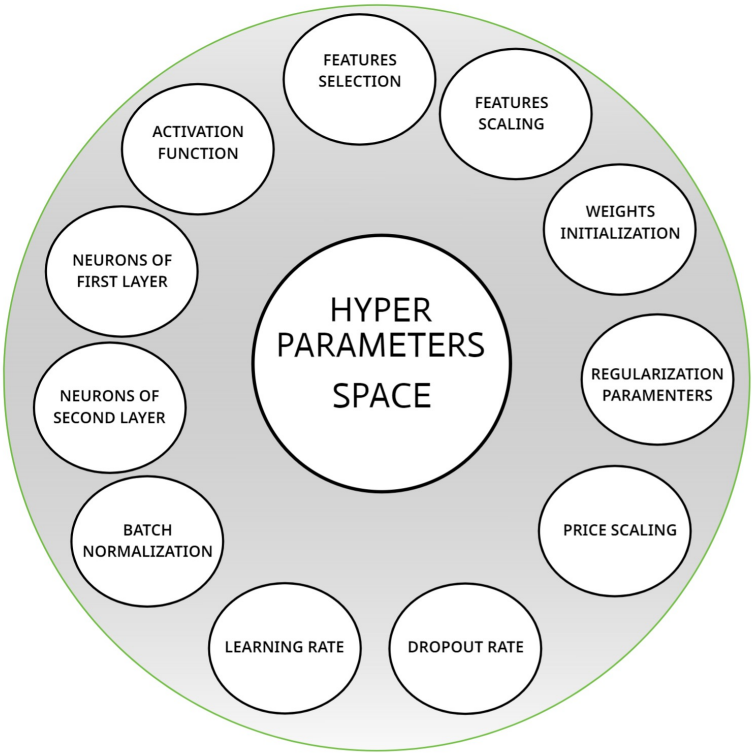

Figure 5.4: Visual representation of the hyperparameter space

- **Activation Function**: is a discrete uniform distribution over $\{relu, softplus, tanh, selu, LeakyRelu, PReLU, sigmoid\}$ (for details see e.g. Ramachandran et al. (2017)).
- **Neurons of layers**: describes the number of neurons for each layer, is a discrete uniform distribution over $\{50, ..., 500\}$ for the first layer and over $\{25, ..., 400\}$ for the second one.
- **Batch normalization**: it's a boolean that toggles whether or not to perform standard data scaling in the input layer, the probability of each outcome is 50%.
- **Learning rate**: is a continuous uniform distribution over $[5 * 10^{-4}, 0.1]$
- **Dropout rate**: represents the percentage of nodes to skip in the layer. It's a factor used as a safeguard against overfitting and follows a continuous uniform distribution in the range $[0, 1]$.
- **Price scaling**: It defines the transformation to be applied to the time series of electricity prices. This transformation follows a discrete uniform distribution over the set $\{No, Z-score, Min-Max, Invariant\}$ (for details see e.g Cao et al. (2016)), the 'Invariant' data scaling method is the same used for the linear models introduced

in the previous chapter.

- **Feature scaling**: same distribution as the price scaling attribute but for the features instead.
- **Regularization parameter**: It is a two-step distribution: first, a boolean is sampled with a 50% probability, representing the use or non-use of regularization in the loss function. If regularization is chosen, the corresponding penalty parameter lambda is then selected by sampling from a continuous uniform distribution in the range $\{10^{-5}, 1\}$.
- **Weights initialization**: It describes the weight initialization method for interconnecting neurons. It's a discrete uniform distribution over $\{Orthogonal, lecun\_uniform, glorot\_uniform, glorot\_normal, he\_uniform, he\_normal\}$ (for details see e.g. Glorot and Bengio (2010)).
- **Feature Selection**: It's a multi-step distribution where we sequentially decide whether to include the feature in the model or not. Each feature is a binary variable with a 50% probability for each outcome. We include for the decision the set of features: $\boldsymbol{p}_{d-1}$, $\boldsymbol{p}_{d-2}$, $\boldsymbol{p}_{d-3}$, $\boldsymbol{p}_{d-7}$, $\boldsymbol{l}_d$, $\boldsymbol{l}_{d-1}$, $\boldsymbol{l}_{d-7}$, $\boldsymbol{g}_d$, $\boldsymbol{g}_{d-1}$, $\boldsymbol{g}_{d-7}$, $\boldsymbol{D}$.

5.1.2. Methodology

Here, we introduce the methodology of the tree-structured Parzen estimator. We start by fixing some notation.

- $\boldsymbol{\chi}$: The hyperparameters space
- $\boldsymbol{x} \in \boldsymbol{\chi}$: A hyperparameters configuration.
- $f(\boldsymbol{x})$: An objective function to minimize $f : \boldsymbol{\chi} \to \mathbb{R}$
- $y = f(\boldsymbol{x}) + \epsilon$: an observation of the objective function with a noise ϵ
- $D := \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$: a set of observations
- $D^{(l)}, D^{(g)}$: a better and a worse group in D (with sizes $N^{(l)}N^{(g)}$, respectively)
- $\gamma \in (0, 1]$: a top quantile for the better group $D^{(l)}$
- $y^{\gamma} \in \mathbb{R}$: the top-γ quantile objective value in D
- $p(\boldsymbol{x}|D^{(l)}), p(\boldsymbol{x}|D^{(g)})$: the probability density functions for the better and worse group.

We need to address a minimization problem of the following form:
Find $\boldsymbol{x_{opt}} \in \boldsymbol{\chi}$ such that $\boldsymbol{x_{opt}} = \arg\min_{\boldsymbol{x}\in\boldsymbol{\chi}} f(\boldsymbol{x})$
Where $f(\boldsymbol{x})$ is the **mean absolute error** of the forecast on the validation set. Instead of directly minimizing $f(\boldsymbol{x})$ we find it's minimum by maximizing another function: the expected improvement.

$$EI_{y^\gamma}(\boldsymbol{x}) = \int_{-\infty}^{\infty} max(y^\gamma - y, 0)p(y|\boldsymbol{x})dy = \int_0^{y^\gamma} (y^\gamma - y)p(y|\boldsymbol{x})dy \tag{5.1}$$

The lower bound of the integral becomes zero because the objective function is positively defined. We observe that the maximum of this function is found when y (an observation of the objective function) is equal to zero, and finding **x** that maximizes the expected improvement means concentrating $p(y|\boldsymbol{x})$ around zero.

Lemma 5.1 Under the hypothesis that $p(\boldsymbol{x}|y, D) = \begin{cases} l(\boldsymbol{x}) \ if \ y \leq y^\gamma \\ g(\boldsymbol{x}) \ if \ y > y^\gamma \end{cases}$

we have that $EI_{y^\gamma}(\boldsymbol{x})$ is directly proportional to $\dfrac{l(\boldsymbol{x})}{g(\boldsymbol{x})}$

The hypothesis is stating that the distribution of $\boldsymbol{x}$ given y depends on y only in terms of its relationship with y^γ. We provide the proof below (see e.g. Bergstra et al. (2011)).

Proof: By applying Bayes' theorem, we obtain

$$EI_{y^\gamma}(\mathbf{x}) = \int_0^{y^\gamma} (y^\gamma - y)p(y|\boldsymbol{x})dy = \int_0^{y^\gamma} (y^\gamma - y)\frac{p(\boldsymbol{x}|y)p(y)}{p(\boldsymbol{x})}dy \tag{5.2}$$

By construction, $\gamma = p(y < y^\gamma)$ and $p(\boldsymbol{x}) = \int_{\mathbb{R}} p(\boldsymbol{x}|y)p(y)dy = \gamma l(\boldsymbol{x}) + (1-\gamma)g(\boldsymbol{x})$. The latter equality emerges from the decomposition of the integral and the application of the hypothesis outlined in the lemma. Therefore

$$\int_0^{y^\gamma} (y^\gamma - y)p(\boldsymbol{x}|y)p(y)dy = l(\boldsymbol{x})\int_0^{y^\gamma} (y^\gamma - y)p(y)dy = \gamma l(\boldsymbol{x})y^\gamma - l(\boldsymbol{x})\int_0^{y^\gamma} p(y)dy \tag{5.3}$$

So that finally $EI_{y^\gamma}(\boldsymbol{x}) = \dfrac{\gamma y^\gamma l(\boldsymbol{x}) - l(\boldsymbol{x})\int_0^{y^\gamma} p(y)dy}{\gamma l(\boldsymbol{x}) + (1-\gamma)g(\boldsymbol{x})} \propto \left(\gamma + \frac{g(\boldsymbol{x})}{l((x))}(1-\gamma)\right)^{-1}$. Having provided the building blocks for this method we present the TPE algorithm below:

Algorithm 5.1 Tree-structured Parzen estimator

$D \leftarrow \emptyset$ ▷ initializing empty set
$N_{init} \leftarrow 50$ ▷ dimension of the first sample collected
$N_{iter} \leftarrow 1000$ ▷ number of iterations for the optimization
$\gamma \leftarrow 0.25$ ▷ Quantile used for splitting D
for $n = 1, ..., N_{init}$:
    Sample $\mathbf{x}_n$
    $y_n = f(\mathbf{x}_n)$
    Append $\{(\mathbf{x}_n, y_n)\}$ to D ▷ Collecting data
for $dummy = 1, ..., N_{iter}$: ▷ Optimization
    Split D into $D^{(l)}$ and $D^{(g)}$ according with γ
    Build $p(\mathbf{x}|D^{(l)})$ and $p(\mathbf{x}|D^{(g)})$ ▷ see equation 5.2 for the details
    Sample a set $S = \{\mathbf{x}_s\} \sim p(\mathbf{x}|D^{(l)})$
    Pick $\mathbf{x}_{N+1} = \arg\max_{\boldsymbol{x} \in S} \frac{p(\boldsymbol{x}|D^{(l)})}{p(\boldsymbol{x}|D^{(g)})}$ ▷ Moving towards the maximum of EI
    $y_{N+1} = f(\mathbf{x}_{N+1})$
    Append $\{\mathbf{x}_{N+1}, y_{N+1}\}$ to D

Here we show the methodology for building the probability density functions $p(\boldsymbol{x}|D^{(l)})$ and $p(\boldsymbol{x}|D^{(g)})$:

$$\begin{aligned} p(\mathbf{x}|D^{(l)}) &= \omega_0 p_0(\mathbf{x}) + \sum_{n=1}^{N^{(l)}} \omega_n k(\mathbf{x}, \mathbf{x}_n|b^{(l)}) \\ p(\mathbf{x}|D^{(g)}) &= \omega_0 p_0(\mathbf{x}) + \sum_{n=N^{(l)}+1}^{N} \omega_n k(\mathbf{x}, \mathbf{x}_n|b^{(g)}) \end{aligned} \tag{5.4}$$

Where:

- $w_n = \begin{cases} \frac{1}{N^{(l)}+1} & n = 0, ..., N^{(l)} \\ \frac{1}{N^{(g)}+1} & n = N^{(l)}+1, ..., N+1 \end{cases}$
- $b^{(l)}$ and $b^{(g)}$ are constants parameters of $k(\mathbf{x}, \mathbf{x}_n|b^{(g)})$ called bandwidth, and determine the amount of exploration we wish to do, a large bandwidth leads to more exploration.
- $k(\mathbf{x}, \mathbf{x}_n|b) = \begin{cases} 1 - b & x = x_n \\ \frac{b}{C-1} & otherwise \end{cases}$ for categorical variables, C is the number of choices in the categorical set. It's called Aitchison-Aitken kernel.

- $k(\mathbf{x}, \mathbf{x}_n|b) = \frac{1}{\sqrt{2\pi b^2}} exp\left(-\frac{1}{2}\left(\frac{x - x_n}{b}\right)^2\right)$ it's a Gaussian kernel for numerical variables.

- $p_0(\mathbf{x}) \sim N\left(\frac{R+L}{2}, (R-L)^2\right)$ for numerical variables defined on [R,L]

- $p_0(\mathbf{x}) \sim U\left(\{1, ..., C\}\right)$ for categorical variables

5.1.3. Experiments

In this section, we present some empirical analyses of TPE, specifically an example of the convergence profile and an observation on the impact of the feature selection.

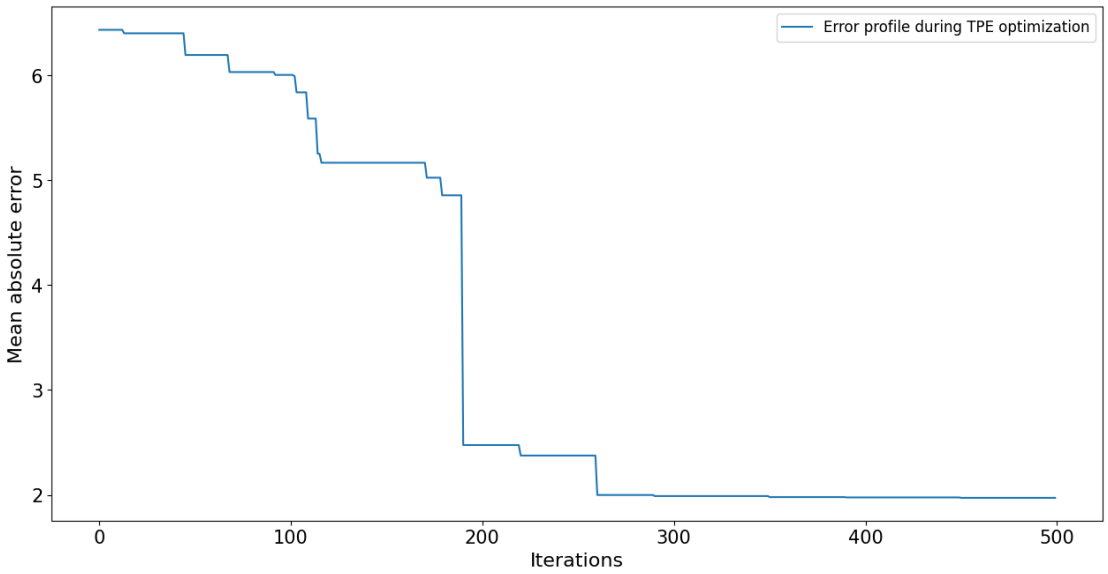

Figure 5.5: Convergence of the TPE algorithm for the Nord Pool dataset, on the x-axis the number of iterations computed and the MAE error on the validation on the y-axis

Examining Figure 5.5, we observe a visual representation of convergence of the TPE algorithm for the Nord Pool dataset. On the y-axis the objective function and the iterations on the x-axis. The optimizator converges after 280 iterations, it becomes apparent that the algorithm is highly proficient at selecting the optimal set of hyperparameters for the neural network. The convergence profile is consistently non-increasing, which aligns with the algorithm's tendency to select the argmax of the expected improvement. Convergence is achieved after roughly 280 iterations, resulting in a nearly 70% reduction of the initial error.

Now, let's examine the impact of feature selection within the TPE framework. As seen in Figure 5.5, after approximately 280 iterations, the optimizer discovers a set of hyperparameters that achieve a mean absolute error of around 1.9 on the validation set. We aim to evaluate the convergence profile while keeping all the hyperparameters chosen by TPE fixed and leaving the feature selection unrestricted. This approach allows us to compare the effect of feature selection versus that of hyperparameter tuning.

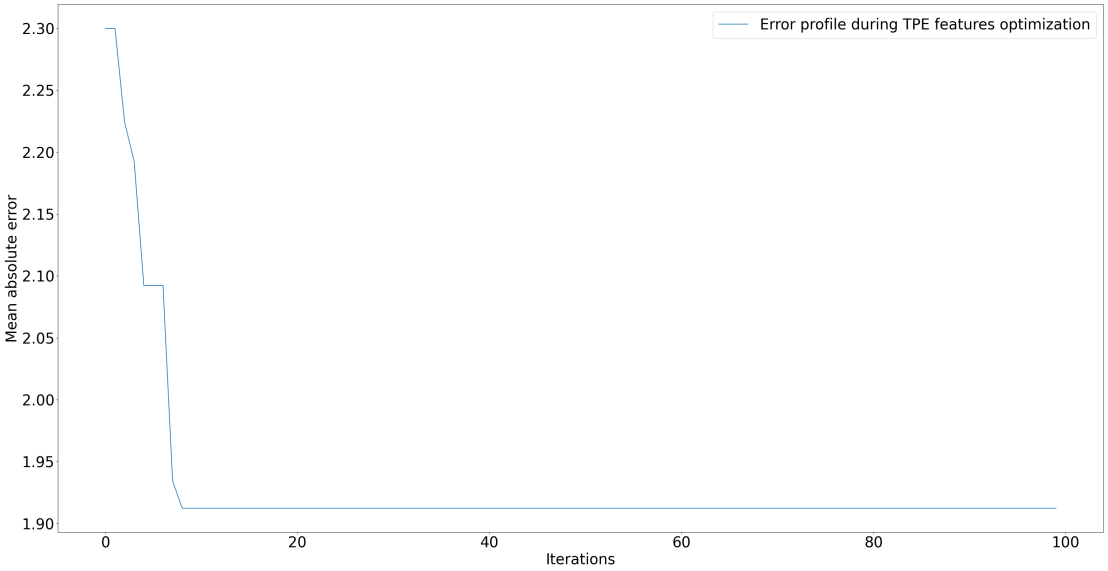

Figure 5.6: Convergence profile of the mean absolute error on the validation set by fixing all hyperparameters chosen by the previous tuning except for the features.

In Figure 5.6, a visual representation of the convergence profile of the tree-structured Parzen estimator with fixed neural network hyperparameters and unfettered feature selection optimization. The impact on the convergence is about an 18% reduction in error, which, while not extremely substantial, is still significant. Convergence occurs quite quickly, after 7-8 iterations, a feature that isn't surprising considering the significantly smaller number of possible feature combinations compared to the potential hyperparameter choices.

5.2. DNN Stability

In this section, we provide a brief analysis of the stability of the resulting deep neural network concerning the seed used. This is to gain a good understanding of the model's prediction capabilities in a generalized stochastic context. We aim to obtain a predictor that is close to being independent of the seed.

DNN performance statistics			
Mean	*Min*	*Max*	*Std.deviation*
1.99	1.88	2.26	0.15

Table 5.1: Statistical table of the variation of the mean absolute error on the test set in relation to the seed used, in a sample of 6 runs

We observe in Table 5.1 the statistics on the variation in mean absolute error on the test set in relation to the seed used. It's the mean absolute error after having fixed the hyperparameters of the deep neural network, changing with respect to the chosen seed, on the Nord Pool dataset. We see an average error of 1.99 with a standard deviation of 0.15 in a sample of 6 runs. From this analysis, we deduce that the impact of the seed on the error should be around 7%, although a sample of 6 seeds might not be sufficient, especially considering that the min-max range is 0.38, approximately 20% of the mean.

This analysis answers to the natural question whether the DNN, being a more complex model than the benchmark models, achieves more accurate predictions regardless of the seed used. As we will show in the result's section, this is indeed the case.

5.3. Chapter summary

In this chapter, we have explored the functioning of the more complex model that we experimented with. At this point, we have three models to test on six different datasets, three containing day-ahead prices and three with the seasonal component removed. Our objective is to test the performance of conformal prediction intervals in the field of EPF, assessing the ability to predict uncertainty with both simple and complex models. We have constructed the necessary building blocks to conduct this analysis.

6 | Prediction intervals

In this chapter, we introduce two techniques for predicting uncertainty in day-ahead electricity price forecasting. We implement and test these techniques on various datasets and models to assess their feasibility. Similar analyses have been presented in some recent papers (Kath and Ziel (2021) and Kowalczewski (2019)), they typically apply these techniques to linear models optimized with various techniques. However, an analysis that compares models with different levels of complexity is still relatively rare in the literature. Moreover, we aim to compare the performances not only across different models but also between the original data and data with the seasonality removed.

When referring to prediction intervals, it's important to focus on two key concepts, which together represent the important characteristics of a prediction interval.

- **Validity**: $C^\alpha(x_1, ..., x_n)$ is a **valid** prediction set for y_{n+1} if

$$P(y_{n+1} \in C^\alpha(x_1, ..., x_n)) \geq 1 - \alpha \tag{6.1}$$

  Of course, it is essential to construct valid prediction intervals because, without this foundation, the very meaning of confidence level would be lost. It is interesting to observe that in the definition of validity, we do not rule out unbounded intervals.

- **Efficiency**: $C_1^\alpha(x_1, ..., x_n)$ is more **efficient** than $C_2^\alpha(x_1, ..., x_n)$ if

$$\mathbb{E}[|C_1^\alpha(x_1, ..., x_n)|] < \mathbb{E}[|C_2^\alpha(x_1, ..., x_n)|] \tag{6.2}$$

  Efficiency refers to the ability of the interval to maintain its validity with a limited width.

Our work indeed focuses on exploring these two characteristics in the world of EPF, using various forecasting techniques and different types of data.

6.1. Conformal prediction intervals

Here, we introduce the first category of prediction intervals we examined: the conformal prediction intervals. They have previously been applied in few studies (see e.g. Kath and Ziel (2021) and Kowalczewski (2019)), focusing on the original prices and employing straightforward forecasting models.

6.1.1. General features

Let's present the main pros and cons of using this model.

- **Pros**:
  - generates prediction intervals that maintain the specified confidence level of $1-\alpha$, which means they respect validity. The user sets the desired confidence level in advance.
  - The only assumption is on exchangeability, while no specific assumptions are needed concerning the underlying distributions.
  - is model-agnostic and can be combined with any individual prediction model because it relies solely on the final prediction generated by a classification or regression model.
- **Cons**:
  - demands a larger amount of historical data compared to other models. This is because it involves fitting a point prediction model, generating out-of-sample forecasts, and subsequently deriving prediction intervals from these forecasts.
  - it may not be well-suited for time series data with significant structural breaks. In cases where a regime switch or substantial structural change occurs, the assumption of exchangeability, which is fundamental to conformal prediction, may no longer hold.
  - it can be significantly affected by outliers, potentially leading to a degradation in its efficiency.

We briefly focus on discussing the assumption of data exchangeability, which doesn't hold in the context of time series, particularly if they exhibit seasonality. This forms the core of our study: adapting this assumption to the context of EPF. If we succeeded in this endeavor, we would have a powerful tool capable of predicting uncertainty, regardless of the proposed model, all with a low computational cost.

6.1.2. Methodology

The method commences with the division of the available dataset into a training set, calibration set, and test set. We followed the same splitting procedure as employed in calibrating and training the models. The "calibration set" mentioned here corresponds to what we previously denoted as the "validation set" in the models section.

- $\mathbf{Z}_{train} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_M, y_M)\}$ is the training set
- $\mathbf{Z}_{calib} = \{(\mathbf{x}_{M+1}, y_{M+1}), ..., (\mathbf{x}_L, y_L)\}$ is the calibration set
- $\mathbf{Z}_{test} = \mathbf{x}_{L+1}$ is the test set

We employ the training set to train any point prediction model and make the forecast for the observations in the calibration set, resulting in a collection of point predictions. Using the actual values from the calibration set, we can compute the non-conformity scores, which are determined by evaluating the absolute differences between predicted prices and realized prices.

$$\lambda_i = |y_i - \hat{y}_i| \quad i = M+1, ..., L \tag{6.3}$$

The non-conformity threshold, which is utilized to construct the prediction interval, is essentially the empirical quantile of the distribution of non-conformity scores based on the realized values.

$$\begin{aligned} r(\lambda) &= \frac{\#\{i \in \{M+1, ..., L\} : \lambda_i < \lambda\} + 1}{\#Z_{calib} + 1} \\ \lambda^{\alpha}_{L+1} &= min\{\lambda \in \{M+1, ..., L\} : r(\lambda) \geq 1 - \alpha\} \end{aligned} \tag{6.4}$$

The final step involves concatenating the training set and the calibration set, retraining the point prediction model, and obtaining predictions for the test set date. To this predicted value, we add and subtract the non-conformity threshold to obtain the prediction interval.

$$y^{\alpha}_{L+1} = \hat{y}_{L+1} \pm \lambda^{\alpha}_{L+1} \tag{6.5}$$

6.1.3. Theory framework

Here, we summarize the concept behind this method, emphasizing its validity given exchangeable observations (see e.g. Fontana et al. (2023) for the full discussion).

The underlying idea of this method is that if the observations are exchangeable, we can define a rank variable distributed as a discrete uniform distribution.

$$R_i = \sum_{j=M+1}^{L+1} \mathbb{1}(\lambda_j - \lambda_i) \tag{6.6}$$

R_i is called **rank** of the i-th observation, and if the assumption of data exchangeability holds, it follows a uniform distribution.

$$R_i \sim U(0, ..., L - M) \tag{6.7}$$

Then

$$P(R_{L+1} \leq \lceil (1-\alpha)(L-M) \rceil) = \frac{\lceil (1-\alpha)(L-M) \rceil}{(L-M)} \geq (1-\alpha) \tag{6.8}$$

Restating the event $R_{L+1} \leq \lceil (1-\alpha)(L-M) \rceil$ as a function of $\{\mathbf{x}_{M+1}, ..., \mathbf{x}_L\}$ allows us to define the prediction interval.

$$C^{\alpha}(\mathbf{x}_{M+1}, ..., \mathbf{x}_L) = \{\mathbf{x}_{L+1} \in \mathbb{R} : R_{L+1} \leq \lceil (1-\alpha)(L-M) \rceil\} \tag{6.9}$$

We have stated here that if the **exchangeability** of observations holds, the conformal prediction interval is **valid**.

6.1.4. A test on price data

Here, we provide a concise overview of some results obtained by applying this method to the Nord Pool electricity price time series.

Since the observations are interdependent and influenced by seasonality, the exchangeability assumption is not valid. Consequently, we observe a substantial decrease in the efficiency of prediction interval for the LEAR model, while for the DNN and fARX we can't even assure validity. These intervals can have widths that are orders of magnitude larger than the point predictions generated by the model, or very small, underestimating the uncertainty.

In the table below, we present two metrics obtained by applying conformal prediction intervals to the price time series, using 4 different quantile levels: 95%, 90%, 80%, 70%. The first metric describes the average width of the prediction interval. The second one, named delta unconditional coverage (Δ_{UC}), is the difference between the expected percentage of points that should fall outside the prediction interval (which is 1-α for an α

confidence interval) and the actual observed percentage of points that fall outside the interval predicted by the model.

$$\Delta_{UC} = (1-\alpha) - \frac{number\ of\ points\ outside\ the\ interval}{number\ of\ observations} \tag{6.10}$$

Conformal predictions on Nord Pool								
Model	*Mean width 95th*	*Delta UC 95th*	*Mean width 90th*	*Delta UC 90th*	*Mean width 80th*	*Delta UC 80th*	*Mean width 70th*	*Delta UC 70th*
LEAR	45.8	-3.82%	40.2	-7.62%	35.9	-15.56%	32.7	-22.15%
fARX	39.76	2.93%	13.9	4.65%	8.2	6.32%	7.1	7.56%
DNN	16.91	5.13%	11.23	8.61%	7.45	12.67%	4.98	15.75%

Table 6.1: Conformal prediction interval performance on Nord Pool price series

In Table 6.1 we report the conformal prediction interval performance on Nord Pool price series. We observe an efficiency degradation for the LEAR model, while prediction intervals are not valid for fARX and DNN, observing an uncertainty underestimation. The conformal prediction interval tends to overestimate the prediction uncertainty for LEAR, leading to wide and inefficient intervals, and to underestimate the uncertainty for the DNN and fARX.

This is why we present a modified version of the model that considers the dependence of observations and aims to restore the exchangeability assumption.

6.2. Normalized conformal prediction intervals

Given the evident inefficiency of conformal predictors applied to non-exchangeable data, we seek an alternative solution to the problem. One solution proposed in the literature (see e.g. Kath and Ziel (2021)) is the normalized conformal prediction interval.

The main difference compared to the previous model is the non-conformity score, which is defined as:

$$\lambda_i = \frac{|y_i - \hat{y}_i|}{|\hat{\epsilon}_i|} \tag{6.11}$$

where $|\hat{\epsilon}_i|$ is the module of the error predicted by a second model. This modification aims to bring the empirical distribution of scores closer to an independent distribution. By considering the ratio of the observed error on the calibration set and the error predicted by the model as a new score measure, we provide a smoothing that reduces the impact of data dependency.

The non conformity threshold is computed the same way as equation (6.4), and the final prediction will be:

$$y^{\alpha}_{L+1} = \hat{y}_{L+1} \pm \lambda^{\alpha}_{L+1}|\hat{\epsilon}_{L+1}| \tag{6.12}$$

Adding the error predicted by the point prediction model serves to rescale the final prediction, considering the smoothing applied earlier. We provide in Figure 6.1 the scheme of the method.

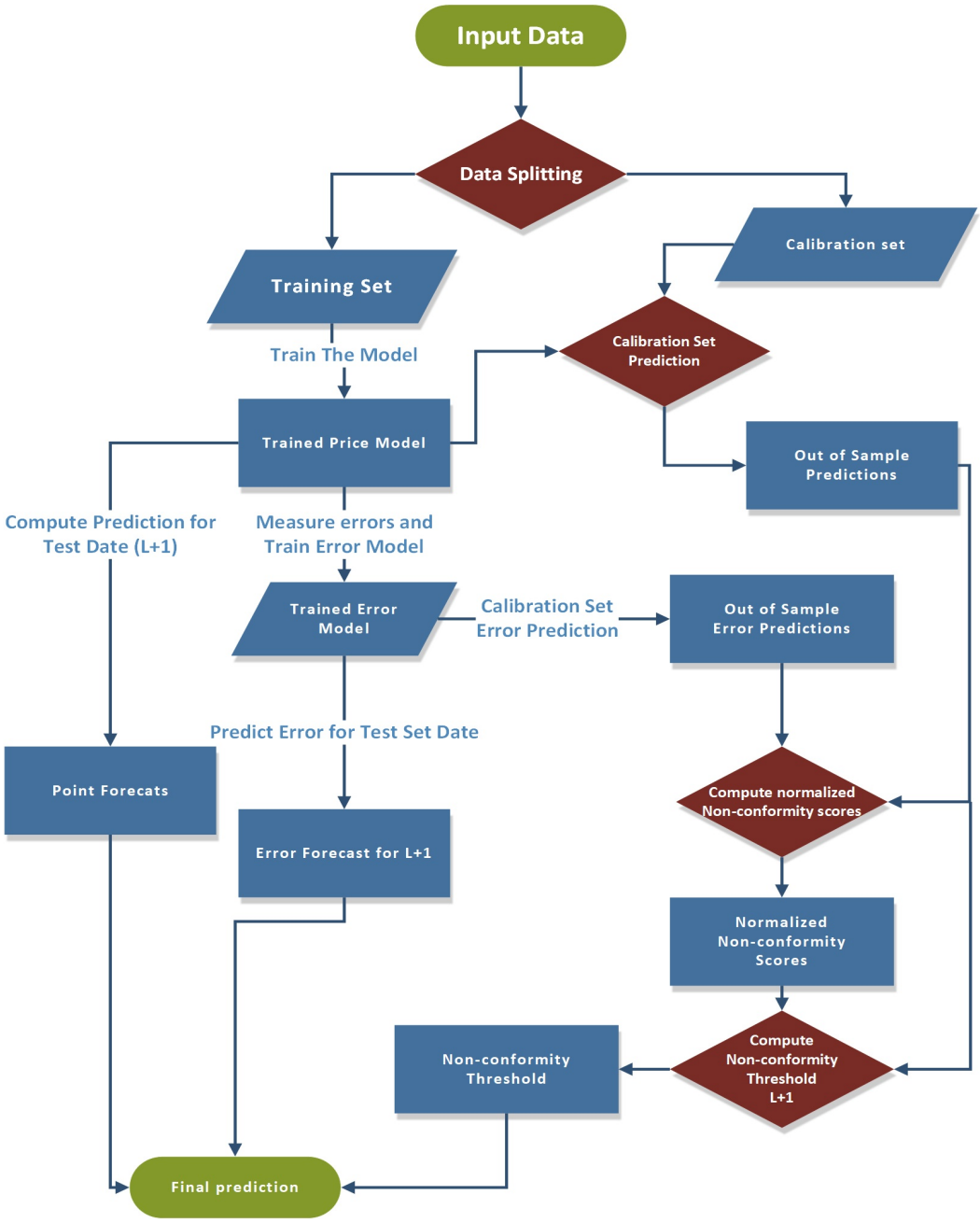

Figure 6.1: Normalized conformal prediction

In Figure 6.1, we observe that the main distinction from conformal prediction intervals is the inclusion of a secondary model dedicated to forecasting the prediction error. In our study, this model, is the one selected by the hyperparameters optimization. After choosing and training the model for price forecasting, we calculate prediction errors on the training set. Subsequently, we retrain the same model, minimizing the loss function with respect to these computed residuals. This process results in two trained models on the same input set (training set): one for forecasting day-ahead prices, and another for predicting the associated forecasting errors. The latter model is employed to predict errors on the calibration set, generating normalized non-conformity scores (see equation (6.11)). After having computed the chosen quantile from this score distribution we can predict the forecasting error on the test set, and construct the prediction interval (see equation (6.12)). The Python code utilized for implementing this methodology has been included in Appendix B.

As we show in the next table, this approach proves to be more effective, producing narrower widths, thus showing greater efficiency compared to the conformal prediction model even if it is more complex than a pure point forecasting model. This is the reason why we opt to test this method on the datasets. To be more specific, we will evaluate it using the three introduced models, across the three available datasets, and on both the original prices and the deseasonalized time series.

6.2.1. Experiments

Here we show the same metrics as in Table 6.1, but with the intervals computed using the normalized conformal prediction intervals methodology.

Normalized conformal predictions on Nord Pool								
Model	*Mean width 95th*	*Delta UC 95th*	*Mean width 90th*	*Delta UC 90th*	*Mean width 80th*	*Delta UC 80th*	*Mean width 70th*	*Delta UC 70th*
LEAR	38.8	-2.45%	34.2	-6.14%	29.8	-13.22%	26.8	-20.93%
fARX	40.1	2.09%	14.8	3.13%	8.4	4.28%	6.27	5.38%
DNN	17.1	4.94%	11.56	7.61%	7.4	12.22%	5.35	15.3%

Table 6.2: Normalized Conformal prediction interval performance on Nord Pool price

We observe in Table 6.2, the same characteristics of Table 6.1, but less prominent. The efficiency loss characteristics for the LEAR model and the underestimation of uncertainty for the DNN and fARX models persist. However, we have obtained results that are slightly more in line with the ideal performance, which would be a delta UC of zero

with the smallest possible interval width. Since we still observe an improvement in each metric for every model, we decide to continue the analysis by shifting our focus to the normalized conformal prediction intervals and observing their performance on seasonally adjusted prices.

In Figure 6.2, we show a graphical representation of a forecast on the Nord Pool test set. It incorporates normalized conformal prediction intervals at the 95th confidence level.

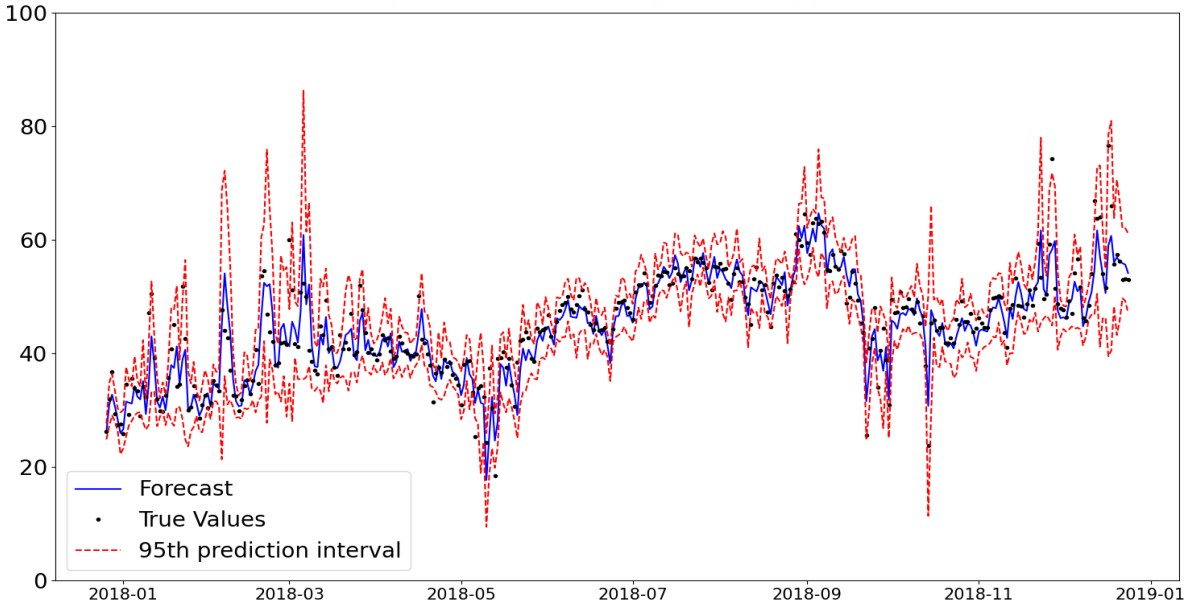

Figure 6.2: DNN forecast of Nord Pool prices for hour 12:00, with normalized conformal prediction interval at the 95th confidence level. The x-axis shows the time, and the y-axis the prices.

6.3. Chapter summary

In this chapter, we have introduced and briefly compared two different methodologies for predicting uncertainty in forecasting. We have observed how the conformal prediction interval technique loses both the efficiency and validity when applied to the historical time series of electricity day-ahead prices, as the assumption of exchangeability breaks down. Then we have presented a modified version that takes into account a model for the point forecast of the error, which smoothes the empirical distribution of scores. After observing that this method consistently improves the critical metrics, we chose it to explore how it performs when applied to different versions of the time series and various models.

7 | Results

One of the most commonly used interval prediction technique in the EPF domain is Quantile Regression Averaging (QRA) (see e.g. Maciejowska et al. (2016)) and a comparison between this model and the normalized conformal prediction interval has already been presented in the literature (see e.g. Kath and Ziel (2021)). In this work, we aim to delve into conformal prediction intervals and investigate their quality when applied to the deseasonalized price time series, regardless of the chosen model or dataset.

We present the results by showcasing the relevant error metrics, which we can categorize into two groups: those related to accuracy in point forecasting and those related to the quality of prediction intervals. Then, we show the results obtained for Nord Pool. Additionally, the results for EPEX FR and EPEX DE are provided in Appendix A.

7.1. Error Metrics

We begin presenting the error metrics used to compare both the performance of point prediction models and the performance of the normalized conformal prediction intervals.

7.1.1. Point predictions

The error metrics used to evaluate the performance of the point prediction model for electricity prices are as follows:

- **Mean absolute error**: is a measure that quantifies the error in tearms of the prices distance, it is very infomative if used for comparing different results in the same dataset, but not for comparing different dataset with different magnitudes. $MAE = \frac{1}{|T|} \sum_{d,h \in T} (|p_{d,h} - \hat{p}_{d,h}|)$

- **Root mean square error**: RMSE is an error metric with properties similar to MAE. It can be higher, suggesting the existence of errors of great magnitude. $RMSE = \sqrt{\frac{1}{|T|}\sum_{d,h\in T}(p_{d,h} - \hat{p}_{d,h})^2}$

- **Mean absolute percentage error**: The MAPE aims to provide information where MAE and RMSE fall short. It represents the average of the relative errors compared to the actual price magnitude. It is useful for comparing the performance of different datasets, provided that the price magnitude to be predicted is not too small. In such cases, the MAPE may present very high values, approaching infinity if there are zero prices. $MAPE = \frac{1}{|T|}\sum_{d,h\in T}\frac{|p_{d,h} - \hat{p}_{d,h}|}{|p_{d,h}|}$

- **Symmetric mean absolute percentage error**: The sMAPE is an error measure with a similar objective to MAPE, but it aims to resolve the issue of extremely low or zero prices. Rather than calculating the error in relation to the price, it calculates the error as a percentage of the sum of the actual price and the predicted price. This helps alleviate the problem of extremely high errors when dealing with zero prices. $sMAPE = \frac{1}{|T|}\sum_{d,h\in T}\frac{|p_{d,h} - \hat{p}_{d,h}|}{|p_{d,h}| + |\hat{p}_{d,h}|}$

- **Relative mean absolute error**: rMAE is an error metric that assesses the performance of a point forecast model by comparing it to the performance of a naive model. The naive model in this context is a predictor that forecasts the next day's price to be the same as the previous day's price. This metric helps determine how well the forecasting model performs in comparison to a simple, baseline approach. $rMAE = \frac{1}{|T|}\sum_{d,h\in T}\frac{|p_{d,h} - \hat{p}_{d,h}|}{|p_{d,h} - p_{d,h}^{naive}|}$

7.1.2. Prediction interval

The error metrics used to compare the performance of prediction intervals are as follows:

- **Mean interval length**: The mean of the difference between the upper bound and lower bound, or the width of the prediction interval, is representative of the efficiency of the prediction interval.

- **Delta coverage**: It is a metric that assesses the validity of the prediction interval. This metric quantifies the difference between the expected percentage of points that should fall outside the prediction interval (which is 1-α for an α confidence interval) and the actual observed percentage of points that fall outside the interval as predicted by the model.

- **Pinball loss**: The pinball loss, also known as quantile loss, is a metric used to evaluate the accuracy of quantile predictions in forecasting, particularly when constructing prediction intervals. It quantifies the difference between the predicted quantile and the actual value of the target variable. This metric provides insights into both the efficiency and validity of the prediction interval. In our analysis, it serves as the primary metric for comparison since comprises informations on both validity and efficiency.

  Let α be the quantile target, $p_{d,h}$ the realized price and $C^{\alpha}_{d,h} = [Q^{1-\alpha}_{lower}, Q^{\alpha}_{upper}]$ the α prediction interval for $p_{d,h}$. Then

$$Pinball(Q^{\alpha}_{upper}, p_{d,h}) = \begin{cases} (1-\alpha)(Q^{\alpha}_{upper} - p_{d,h}) \ if \ Q^{\alpha}_{upper} > p_{d,h} \\ \alpha(p_{d,h} - Q^{\alpha}_{upper}) \ if \ Q^{\alpha}_{upper} \leq p_{d,h} \end{cases}$$

$$Pinball(Q^{1-\alpha}_{lower}, p_{d,h}) = \begin{cases} \alpha(Q^{1-\alpha}_{lower} - p_{d,h}) \ if \ Q^{1-\alpha}_{lower} \geq p_{d,h} \\ (1-\alpha)(p_{d,h} - Q^{1-\alpha}_{lower}) \ if \ Q^{1-\alpha}_{lower} < p_{d,h} \end{cases}$$

$$Pinball(C^{\alpha}_{d,h}, p_{d,h}) = \frac{1}{2}\big(Pinball(Q^{\alpha}_{upper}, p_{d,h}) + Pinball(Q^{1-\alpha}_{lower}, p_{d,h})\big)$$

  After computing all the pinball losses between the price and the prediction interval for each $d, h \in T$, we calculate the mean of these pinball losses over the dataset's timeline to obtain the final metric that characterizes the model's pinball loss on that dataset for a specific α level.

7.2. Nord Pool results

In this section, we present the results obtained by applying the normalized conformal prediction intervals to the Nord Pool dataset. We test them in six different scenarios. First, we apply them to the original prices using the three introduced models, and then we repeat the process on the seasonally adjusted prices. For each scenario, we evaluate the metrics at four different alpha levels: 95%, 90%, 80%, and 70%.

Normalized conformal predictions on Nord Pool - $\alpha = 0.95$								
Model	*MAE*	*RMSE*	*MAPE*	*sMAPE*	*rMAE*	*Mean width*	*Delta UC*	*Pinball loss*
LEAR - price	3.1	5.62	9.5%	8.6%	0.95	38.8	-2.45%	1.03
LEAR - residuals	2.68	5.23	8.8%	7.8%	0.89	28.95	1.81%	0.68
fARX - price	2.06	3.72	6.3%	5.9%	0.72	40	2.09%	3
fARX - residuals	2.7	5.2	8.4%	7.8%	0.94	21	0.98%	0.63
DNN - price	1.92	3.65	6.2%	5.3%	0.69	23.2	1.03%	0.76
DNN - residuals	3.1	5.74	9.5%	8.8%	0.92	17.64	2.52%	0.62

Table 7.1: Normalized Conformal prediction interval performance on Nord Pool with a 95% confidence level, we observe that for the fARX and DNN models trained on deseasonalized prices, the point forecast accuracy decreases, while the prediction intervals metrics improve. We also observe the fARX-price model to produce very wide intervals.

Normalized conformal predictions on Nord Pool - $\alpha = 0.9$								
Model	*MAE*	*RMSE*	*MAPE*	*sMAPE*	*rMAE*	*Mean width*	*Delta UC*	*Pinball loss*
LEAR - price	3.1	5.62	9.5%	8.6%	0.95	34.2	-6.14%	1.78
LEAR - residuals	2.68	5.23	8.8%	7.8%	0.89	23.2	2.39%	0.88
fARX - price	2.06	3.72	6.3%	5.9%	0.72	14.8	3.13%	0.86
fARX - residuals	2.7	5.2	8.4%	7.8%	0.94	13.5	1.37%	0.86
DNN - price	1.92	3.65	6.2%	5.3%	0.69	18.4	0.63%	1.17
DNN - residuals	3.1	5.74	9.5%	8.8%	0.92	12.2	3.54%	0.9

Table 7.2: Normalized Conformal prediction interval performance on Nord Pool with a 90% confidence level, the use of residuals for the LEAR model drammatically increases the prediction interval performance, while for the fARX model the pinball losses are the same even if the version fitted on residuals produces tighter intervals with a Delta UC metric closer to zero. The use of residuals for the DNN also improves the prediction interval metrics

Normalized conformal predictions on Nord Pool - $\alpha = 0.8$								
Model	*MAE*	*RMSE*	*MAPE*	*sMAPE*	*rMAE*	*Mean width*	*Delta UC*	*Pinball loss*
LEAR - price	3.1	5.62	9.5%	8.6%	0.95	29.7	-14.2%	3.08
LEAR - residuals	2.68	5.23	8.8%	7.8%	0.89	19.3	2.53%	1.17
fARX - price	2.06	3.72	6.3%	5.9%	0.72	8.4	4.28%	1.03
fARX - residuals	2.7	5.2	8.4%	7.8%	0.94	8.4	1.37%	1.18
DNN - price	1.92	3.65	6.2%	5.3%	0.69	13.8	1.4%	1.7
DNN - residuals	3.1	5.74	9.5%	8.8%	0.92	7.8	5.54%	1.26

Table 7.3: Normalized Conformal prediction interval performance on Nord Pool with a 80% confidence level, the use of residuals for the LEAR model drammatically increases the prediction interval performance, while for the fARX model the widths are the same, the residual version has better Delta UC performances but worse Pinball losses. Regarding the DNN, we observe that the version with residuals produces tighter intervals with better pinball losses but with more 'missed' values in the interval with respect to the full price DNN.

Normalized conformal predictions on Nord Pool - $\alpha = 0.7$								
Model	*MAE*	*RMSE*	*MAPE*	*sMAPE*	*rMAE*	*Mean width*	*Delta UC*	*Pinball loss*
LEAR - price	3.1	5.62	9.5%	8.6%	0.95	26.7	-21.9%	4.16
LEAR - residuals	2.68	5.23	8.8%	7.8%	0.89	11.1	3.1%	1.35
fARX - price	2.06	3.72	6.3%	5.9%	0.72	6.27	5.38%	1.2
fARX - residuals	2.7	5.2	8.4%	7.8%	0.94	6.16	1.23%	1.38
DNN - price	1.92	3.65	6.2%	5.3%	0.69	11	0.51%	2.2
DNN - residuals	3.1	5.74	9.5%	8.8%	0.92	6	5%	1.5

Table 7.4: Normalized Conformal prediction interval performance on Nord Pool with a 70% confidence level, the use of residuals for the LEAR model drammatically increases the prediction interval performance, while for the fARX model the widths are the same, the residual version has better Delta UC performances but worse Pinball losses. Regarding the DNN, we observe that the version with residuals produces tighter intervals with better pinball losses but with more 'missed' values in the interval with respect to the full price DNN.

For the Nord Pool dataset, we can summarize the results by stating that training models on the deseasonalized price time series reduces the accuracy of point forecasts but produces narrower and more reliable prediction intervals.

8 | Conclusions and future developments

8.1. Conclusions

As seen in Chapter 7, the results obtained depend on the specific dataset and the model considered. The use of normalized conformal prediction intervals on models trained on seasonally deseasonalized prices significantly increases the reliability and efficiency of prediction intervals when applied to the Nord Pool dataset. This dataset, as discussed in Chapter 3, is characterized by greater regularity, fewer outliers, and does not contain negative or zero prices. Furthermore, it features a time series that is easier to predict, as confirmed by the error metrics of point forecasts.

Results differ for the EPEX France dataset, where the intervals improve in terms of quality when calibrated on deseasonalized prices only for the LEAR model. For the fARX model, interval performances remain similar, while a degradation is observed when using residuals to train the deep neural network.

In the case of the German dataset, the most challenging to predict due to high variance and the presence of negative prices, a significant deterioration in the performance of prediction intervals is observed when models trained on deseasonalized prices are used.

Another important conclusion, regardless of the dataset considered, is that normalized conformal prediction intervals produce slightly higher-quality intervals for more complex models with greater predictive power.

8.2. Limits of this work

Given the evident variability of results depending on the dataset and model under consideration, it would be advisable to further the analysis on new datasets and models to confirm the observations made. Furthermore, the analysis was conducted on data preceding 2019, in a period of stable geopolitical and climatic conditions, which is not reflective of the current market conditions. An analysis on more recent data might yield different conclusions. Lastly, it would also be beneficial to compare these results with those obtained by applying other uncertainty prediction techniques, as done by Kath and Ziel (2021), such as quantile regression averaging. Performing a further analysis that includes this technique could lead to more reliable conclusions.

8.3. Direction of future work

It is indeed an interesting avenue of research, one that has the potential to enhance the industry's predictive capabilities for uncertainty. Further exploration of the effects of deseasonalization on prediction intervals could provide valuable insights. The claims we have attempted to make are theoretically grounded, as deseasonalized price time series exhibit significantly less autocorrelation and bring the data closer to the exchangeability hypothesis, which is crucial for conformal prediction intervals. As mentioned in the section above, conducting this research on different sets of more recent data and with various models could lead to more robust conclusions. Additionally, a comparison with other prediction interval methodologies may yield different observations and insights.

Bibliography

[1] K. L. Anaya, M. Giulietti, and M. G. Pollitt. Where next for the electricity distribution system operator? Evidence from a survey of european DSOs and national regulatory authorities. *Competition and Regulation in Network Industries*, 23(4): 245–269, 2022.

[2] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

[3] H. Bozdogan. Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika*, 52(3):345–370, 1987.

[4] D. W. Bunn. Modelling prices in competitive electricity markets. *U.S. Department of Energy Office of Scientific and Technical Information*, 2004.

[5] X. H. Cao, I. Stojkovic, and Z. Obradovic. A robust data scaling algorithm to improve classification accuracies in biomedical data. *BMC bioinformatics*, 17(1):1–10, 2016.

[6] A. Cruz, A. Muñoz, J. L. Zamora, and R. Espínola. The effect of wind generation and weekday on spanish electricity spot price forecasting. *Electric Power Systems Research*, 81(10):1924–1935, 2011.

[7] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[8] M. Fontana, G. Zeni, and S. Vantini. Conformal prediction: a unified review of theory and new challenges. *Bernoulli*, 29(1):1–23, 2023.

[9] J.-M. Glachant, P. L. Joskow, and M. G. Pollitt. Handbook on electricity markets. *Edward Elgar Publishing*, 2021.

[10] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[11] C. Graf and D. Wozabal. Measuring competitiveness of the epex spot market for electricity. *Energy Policy*, 62:948–958, 2013.

[12] C. Kath and F. Ziel. Conformal prediction interval estimation and applications to day-ahead and intraday power markets. *International Journal of Forecasting*, 37(2): 777–799, 2021.

[13] J. Kowalczewski. Normalized conformal prediction for time series data. *Kth Royal Institute Of Technology*, pages 41–49, 2019.

[14] P.-H. Kuo and C.-J. Huang. An electricity price forecasting model by hybrid structured deep neural networks. *Sustainability*, 10(4):1280, 2018.

[15] J. Lago, G. Marcjasz, B. De Schutter, and R. Weron. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293:116983, 2021.

[16] W. K. Li. Some Lagrange multiplier tests for seasonal differencing. *Biometrika*, 78 (2):381–384, 1991.

[17] M. Liu and F. F. Wu. Risk management in a competitive electricity market. *International Journal of Electrical Power & Energy Systems*, 29(9):690–697, 2007.

[18] S. Luo and Y. Weng. A two-stage supervised learning approach for electricity price forecasting by leveraging different data sources. *Applied energy*, 242:1497–1512, 2019.

[19] K. Maciejowska, J. Nowotarski, and R. Weron. Probabilistic forecasting of electricity spot prices using factor quantile regression averaging. *International Journal of Forecasting*, 32(3):957–965, 2016.

[20] G. Marcjasz, B. Uniejewski, and R. Weron. On the importance of the long-term seasonal component in day-ahead electricity price forecasting with narx neural networks. *International Journal of Forecasting*, 35(4):1520–1532, 2019.

[21] R. Mushtaq. Augmented Dickey Fuller test. *Université Paris, Panthéon-Sorbonne*, pages 9–13, 2011.

[22] J. Nowotarski and R. Weron. Computing electricity spot price prediction intervals using quantile regression and forecast averaging. *Computational Statistics*, 30:791–803, 2015.

[23] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[24] F. L. Ramsey. Characterization of the partial autocorrelation function. *The Annals of Statistics*, pages 1296–1301, 1974.

[25] U. Ugurlu, I. Oksuz, and O. Tas. Electricity price forecasting using recurrent neural networks. *Energies*, 11(5):1255, 2018.

[26] B. Uniejewski, J. Nowotarski, and R. Weron. Automated variable selection and shrinkage for day-ahead electricity price forecasting. *Energies*, 9(8):621, 2016.

[27] B. Uniejewski, R. Weron, and F. Ziel. Variance stabilizing transformations for electricity spot price forecasting. *IEEE Transactions on Power Systems*, 33(2):2219–2229, 2017.

[28] L. Wang, Z. Zhang, and J. Chen. Short-term electricity price forecasting with stacked denoising autoencoders. *IEEE Transactions on Power Systems*, 32(4):2673–2681, 2016.

[29] S. Watanabe. Tree-structured Parzen estimator: Understanding its algorithm components and their roles for better empirical performance. *arXiv preprint*, 2023.

[30] R. Weron. Modeling and forecasting electricity loads and prices: A statistical approach. *John Wiley & Sons*, 2007.

[31] T.-T. Wong. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern recognition*, 48(9):2839–2846, 2015.

[32] S. J. Wright. Coordinate descent algorithms. *Mathematical programming*, 151(1): 3–34, 2015.

[33] W. Zhang, F. Cheema, and D. Srinivasan. Forecasting of electricity prices using deep learning networks. In *2018 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pages 451–456. IEEE, 2018.

[34] J. H. Zhao, Z. Y. Dong, Z. Xu, and K. P. Wong. A statistical approach for interval forecasting of the electricity price. *IEEE Transactions on Power Systems*, 23(2): 267–276, 2008.

[35] M. Zhou, Z. Yan, Y. Ni, G. Li, and Y. Nie. Electricity price forecasting with confidence-interval estimation through an extended arima approach. *IEE Proceedings-Generation, Transmission and Distribution*, 153(2):187–195, 2006.

A | Appendix

A.1. EPEX France results

In this section, we present the results obtained by applying the normalized conformal prediction intervals to the EPEX France dataset. The structure of the results is the same as Nord Pool.

Normalized conformal predictions on EPEX FR - $\alpha = 0.95$								
Model	*MAE*	*RMSE*	*MAPE*	*sMAPE*	*rMAE*	*Mean width*	*Delta UC*	*Pinball loss*
LEAR - price	4.8	11.5	0.167	0.15	0.66	55.5	-2.53%	1.45
LEAR - residuals	5.5	14	0.162	0.154	0.75	46.2	0.38%	1.05
fARX - price	4.33	11.4	0.139	0.126	0.59	25.9	-0.5%	1
fARX - residuals	5.46	13.45	0.16	0.15	0.75	31	0.6%	1.02
DNN - price	4.28	11.45	0.136	0.124	0.58	33.21	0.21%	0.96
DNN - residuals	4.95	13.1	0.152	0.145	0.7	35.2	-1.2%	1.05

Table A.1: Normalized Conformal prediction interval performance on EPEX France with a 95% confidence level, the use of residuals for the LEAR model dramatically increases the prediction interval performance, while for the fARX model produces larger intervals with a grater percentage of points 'missed', for the DNN the use of residuals generally degradates the prediction intervals quality.

Normalized conformal predictions on EPEX FR - $\alpha = 0.9$								
Model	*MAE*	*RMSE*	*MAPE*	*sMAPE*	*rMAE*	*Mean width*	*Delta UC*	*Pinball loss*
LEAR - price	4.8	11.5	0.167	0.15	0.66	40.5	-5.07%	2.12
LEAR - residuals	5.5	14	0.162	0.154	0.75	37.6	-0.13%	1.58
fARX - price	4.33	11.4	0.139	0.126	0.59	20.3	-1.32%	1.42
fARX - residuals	5.46	13.45	0.16	0.15	0.75	23.8	0.57%	1.54
DNN - price	4.28	11.45	0.136	0.124	0.58	21.2	-0.28%	1.28
DNN - residuals	4.95	13.1	0.152	0.145	0.7	21.9	-1.47%	1.35

Table A.2: Normalized Conformal prediction interval performance on EPEX France with a 90% confidence level, the use of residuals for the LEAR model dramatically increases the prediction interval performance, while for the fARX model produces larger intervals with a grater percentage of points 'missed', for the DNN the use of residuals generally degradates the prediction intervals quality.

Normalized conformal predictions on EPEX FR - $\alpha = 0.8$								
Model	*MAE*	*RMSE*	*MAPE*	*sMAPE*	*rMAE*	*Mean width*	*Delta UC*	*Pinball loss*
LEAR - price	4.8	11.5	0.167	0.15	0.66	29	-9.86%	3
LEAR - residuals	5.5	14	0.162	0.154	0.75	27.2	-2.3%	2.3
fARX - price	4.33	11.4	0.139	0.126	0.59	15	-3.55%	2.31
fARX - residuals	5.46	13.45	0.16	0.15	0.75	17.3	-0.83%	2.27
DNN - price	4.28	11.45	0.136	0.124	0.58	14.8	-2.26%	1.86
DNN - residuals	4.95	13.1	0.152	0.145	0.7	15.5	-3.12%	1.99

Table A.3: Normalized Conformal prediction interval performance on EPEX France with a 80% confidence level, the use of residuals for the LEAR model dramatically increases the prediction interval performance, same for the fARX model trained on deseasonalized prices, for the DNN the use of residuals generally degradates the prediction intervals quality.

Normalized conformal predictions on EPEX FR - $\alpha = 0.7$								
Model	*MAE*	*RMSE*	*MAPE*	*sMAPE*	*rMAE*	*Mean width*	*Delta UC*	*Pinball loss*
LEAR - price	4.8	11.5	0.167	0.15	0.66	23.1	-13.7%	3.7
LEAR - residuals	5.5	14	0.162	0.154	0.75	20.7	-3%	2.8
fARX - price	4.33	11.4	0.139	0.126	0.59	11.9	-5.55%	2.79
fARX - residuals	5.46	13.45	0.16	0.15	0.75	13.5	-1.46%	2.78
DNN - price	4.28	11.45	0.136	0.124	0.58	11.4	-4.37%	2.26
DNN - residuals	4.95	13.1	0.152	0.145	0.7	12.1	-5.95%	2.47

Table A.4: Normalized Conformal prediction interval performance on EPEX France with a 70% confidence level, the use of residuals for the LEAR model dramatically increases the prediction interval performance, same for the fARX model trained on deseasonalized prices, for the DNN the use of residuals generally degradates the prediction intervals quality.

On the EPEX France dataset, we observe that using the residuals for training the LEAR model leads to significant improvements in the error metrics of the prediction intervals. As for the fARX model, the prediction intervals have similar performances when compared to using full prices. However, the effect of using residuals for the DNN degrades the error metrices.

A.2. EPEX Germany results

In this section, we present the results obtained by applying the normalized conformal prediction intervals to the EPEX Germany dataset. The structure of the results is the same as Nord Pool.

Normalized conformal predictions on EPEX DE - $\alpha = 0.95$								
Model	*MAE*	*RMSE*	*MAPE*	*sMAPE*	*rMAE*	*Mean width*	*Delta UC*	*Pinball loss*
LEAR - price	4.76	7.57	inf	0.196	0.58	32.5	-0.7%	0.88
LEAR - residuals	5.5	9.2	inf	0.211	0.69	42.5	1.18%	0.87
fARX - price	4.22	6.54	inf	0.171	0.49	31.5	-1.1%	1.11
fARX - residuals	5.4	8.9	inf	0.2	0.64	28.8	1.38%	1.12
DNN - price	3.57	6.02	inf	0.14	0.43	35.2	0.37%	0.57
DNN - residuals	5.3	8.2	inf	0.21	0.63	38.1	-1.5%	0.98

Table A.5: Normalized Conformal prediction interval performance on EPEX Germany with a 95% confidence level, the use of residuals for the LEAR model produces larger intervals, while for the fARX model we observe tighter intervals with a grater percentage of points 'missed', for the DNN the use of residuals generally degradates the prediction intervals quality.

Normalized conformal predictions on EPEX DE - $\alpha = 0.9$								
Model	*MAE*	*RMSE*	*MAPE*	*sMAPE*	*rMAE*	*Mean width*	*Delta UC*	*Pinball loss*
LEAR - price	4.76	7.57	inf	0.196	0.582	24.5	-1%	1.37
LEAR - residuals	5.5	9.2	inf	0.211	0.69	28.5	0.82%	1.47
fARX - price	4.22	6.54	inf	0.171	0.49	31.5	-1.1%	1.11
fARX - residuals	5.4	8.9	inf	0.2	0.64	28.8	1.38%	1.12
DNN - price	3.57	6.02	inf	0.14	0.43	29.3	0.35%	0.91
DNN - residuals	5.3	8.2	inf	0.21	0.63	28.9	1.38%	1.34

Table A.6: Normalized Conformal prediction interval performance on EPEX Germany with a 90% confidence level, the use of residuals for the LEAR model produces larger intervals with agreater Delta UC metric, while for the fARX model we observe tighter intervals with a grater percentage of points 'missed', for the DNN the use of residuals generally degradates the prediction intervals quality.

Normalized conformal predictions on EPEX DE - $\alpha = 0.8$								
Model	*MAE*	*RMSE*	*MAPE*	*sMAPE*	*rMAE*	*Mean width*	*Delta UC*	*Pinball loss*
LEAR - price	4.76	7.57	inf	0.196	0.582	18	-2.9%	2.09
LEAR - residuals	5.5	9.2	inf	0.211	0.69	18.9	0.82%	1.99
fARX - price	4.22	6.54	inf	0.171	0.49	31.5	-1.1%	1.11
fARX - residuals	5.4	8.9	inf	0.2	0.64	28.8	1.38%	1.12
DNN - price	3.57	6.02	inf	0.14	0.43	16.1	-0.8%	1.42
DNN - residuals	5.3	8.2	inf	0.21	0.63	20.4	-5.5%	1.98

Table A.7: Normalized Conformal prediction interval performance on EPEX Germany with a 80% confidence level, the use of residuals for the LEAR model improves the prediction interval error metrics, while for the fARX model we observe tighter intervals with a grater percentage of points 'missed', for the DNN the use of residuals generally degradates the prediction intervals quality.

Normalized conformal predictions on EPEX DE - $\alpha = 0.7$								
Model	*MAE*	*RMSE*	*MAPE*	*sMAPE*	*rMAE*	*Mean width*	*Delta UC*	*Pinball loss*
LEAR - price	4.76	7.57	inf	0.196	0.582	14.3	-4.1%	2.6
LEAR - residuals	5.5	9.2	inf	0.211	0.69	14.2	0.21%	2.52
fARX - price	4.22	6.54	inf	0.171	0.49	31.5	-1.1%	1.11
fARX - residuals	5.4	8.9	inf	0.2	0.64	28.8	1.38%	1.12
DNN - price	3.57	6.02	inf	0.14	0.43	10.2	-2.9%	1.75
DNN - residuals	5.3	8.2	inf	0.21	0.63	15.2	-8%	2.1

Table A.8: Normalized Conformal prediction interval performance on EPEX Germany with a 70% confidence level, the use of residuals for the LEAR model improves the prediction interval error metrics, while for the fARX model we observe tighter intervals with a grater percentage of points 'missed', for the DNN the use of residuals generally degradates the prediction intervals quality.

For the EPEX DE dataset, the results of comparing the prediction intervals between training on the original price series and training on residuals show better error metrics for the models trained on the original price series. Nevertheless, it's essential to emphasize that the accuracy of point forecasts decreases when using residuals

B | Appendix B

B.1. Normalized conformal prediction intervals

In the provided snippet of code, we show the computation of normalized conformal prediction intervals for a deep neural network (DNN) model.

```
# We are inside a class which handles the train - validation - test
# routine. We initialize the DNN model passing to the DDN class the
# hyperparameters tuned with the TPE
self.model = DNNClass(neurons, n_features, dropout, batch_normalization,
                      learning_rate, verbose, optimizer,
                      activation_function, epochs_early_stopping,
                      scaler, loss, regularization,
                      lambda_reg, initializer, epochs)

# fitting the model minimizing the loss with respect to the prices, the
# validation set is used for early-stopping
self.model.fit(Xtrain, Ytrain, Xval, Yval)

# computing the absolute value of the errors on the training set
residuals = abs(Ytrain - self.model.predict(Xtrain))

# creating a deep copy of the model previously fitted, this is the
# model to predict the forecasting error
residuals_predictor = copy.deepcopy(self.model)

# fitting this model minimizing the loss with respect to the residuals
residuals_predictor.fit(Xtrain, residuals)

# forecasting prices and residuals on the validation set and computing
# normalized non-conformity scores
score = abs(Yval - self.model.predict(Xval)) / abs(residuals_predictor.
    predict(Xval))


```

```
# initializing the container for the quantiles of a discrete uniform
# distribution over the length of the validation set, the chosen
# quantiles are 0.95, 0.9, 0.8 and 0.7
quantile = np.zeros(4)

# filling the container with the respective quantiles, we use the
# ceil function because we need the realized quantiles
quantile[0] = int(np.ceil(np.quantile(np.arange(1, Xval.shape[0], 1),
    0.95)))
quantile[1] = int(np.ceil(np.quantile(np.arange(1, Xval.shape[0], 1),
    0.90)))
quantile[2] = int(np.ceil(np.quantile(np.arange(1, Xval.shape[0], 1),
    0.80)))
quantile[3] = int(np.ceil(np.quantile(np.arange(1, Xval.shape[0], 1),
    0.70)))

# computing the predicted residuals on the test set
errors = abs(residuals_predictor.predict(Xtest))

# saving the price forecasting prediction
Yp = self.model.predict(Xtest)

# computing the prediction interval for each hour
for h in range(24):
    # sorting the scores relative with the specific hour
    score_h = np.sort(score[:, h])
    # extracting the realized quantiles of the scores distribution
    quantile_95 = score_h[quantile[0]]
    quantile_90 = score_h[quantile[1]]
    quantile_80 = score_h[quantile[2]]
    quantile_70 = score_h[quantile[3]]
    # filling the containers. Each one represents a bound (lower or
    # upper) and an alpha confidence level
    Yp_ub_95[h] = Yp[0, h] + quantile_95 * errors[0, h]
    Yp_lb_95[h] = Yp[0, h] - quantile_95 * errors[0, h]
    Yp_ub_90[h] = Yp[0, h] + quantile_90 * errors[0, h]
    Yp_lb_90[h] = Yp[0, h] - quantile_90 * errors[0, h]
    Yp_ub_80[h] = Yp[0, h] + quantile_80 * errors[0, h]
    Yp_lb_80[h] = Yp[0, h] - quantile_80 * errors[0, h]
    Yp_ub_70[h] = Yp[0, h] + quantile_70 * errors[0, h]
    Yp_lb_70[h] = Yp[0, h] - quantile_70 * errors[0, h]
```

List of Figures

List of Tables

Acknowledgements

Ringrazio questi anni, anni intensi, di gioie e difficoltà, anni che costituiscono il miglior bagaglio personale che potessi acquisire per affrontare ogni sfida futura.

Vorrei esprimere la mia più profonda gratitudine ai miei genitori, due figure straordinarie che non solo mi hanno fornito il sostegno necessario per intraprendere questo cammino, ma che, nonostante le difficoltà e le preoccupazioni, mi hanno sempre incoraggiato e compreso con amore incondizionato.

Un ringraziamento speciale va a Tiziana, per il prezioso supporto e la costante presenza, che hanno fatto di questo viaggio un'esperienza ancor più gratificante. Grazie per aver creduto in me e riconosciuto qualità che non sempre riesco a cogliere da solo.

Ringrazio sinceramente il mio gruppo di amici. Il mio successo è un trionfo condiviso con voi, che fate del bene di uno il bene di tutti. La vostra amicizia è una costante fonte di ispirazione e forza.

Desidero inoltre ringraziare il Politecnico di Milano e i rispettivi professori, per avermi concesso l'opportunità di far parte di una delle migliori università del mondo. Un ringraziamento, in particolare, va al Professor Roberto Baviera, per avermi fatto da guida durante il percorso di tesi, e per la passione che dimostra e trasmette nel suo corso di Ingegneria Finanziaria. I suoi insegnamenti hanno influenzato in modo significativo il mio percorso e hanno contribuito a delineare con maggior chiarezza i miei obiettivi futuri.