Machine learning techniques for electricity price forecasting

Léonard Tschora

N° d'ordre NNT : 2024ISAL0007

Thèse de doctorat de l'Université de Lyon
opérée au sein de
L'INSA Lyon

Ecole Doctorale N° 512
Mathématiques et Informatique (InfoMaths)

Spécialité / discipline de doctorat : Informatique

À soutenir publiquement le 17/01/2024 par
Léonard Tschora

Machine Learning Techniques for Electricity Price Forecasting

Devant le jury composé de:

Pr. Siegfried Nijssen	Université Catholique de Louvain	Rapporteur
Pr. Massih-Reza Amini	Université Grenoble Alpes	Rapporteur
Mcf. Charlotte Laclau	Télécom Paris	Examinatrice
Pr. Elisa Fromont	Université Rennes 1	Examinatrice
Pr. Stéphane Canu	INSA Rouen	Examinateur
Dr. Erwan Pierre	BCMEnergy	Co-encadrant en entreprise
Pr. Céline Robardet	INSA Lyon	Directice
Pr. Marc Plantevit	EPITA	Co-directeur

Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
ED 206 CHIMIE	**CHIMIE DE LYON** https://www.edchimie-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr	**M. Stéphane DANIELE** C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne directeur@edchimie-lyon.fr
ED 341 E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** http://e2m2.universite-lyon.fr Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.e2m2@univ-lyon1.fr	**Mme Sandrine CHARLES** Université Claude Bernard Lyon 1 UFR Biosciences Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69622 Villeurbanne CEDEX e2m2.codir@listes.univ-lyon1.fr
ED 205 EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** http://ediss.universite-lyon.fr Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.ediss@univ-lyon1.fr	**Mme Sylvie RICARD-BLUM** Laboratoire ICBMS - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 sylvie.ricard-blum@univ-lyon1.fr
ED 34 EDML	**MATÉRIAUX DE LYON** http://ed34.universite-lyon.fr Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr	**M. Stéphane BENAYOUN** Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 stephane.benayoun@ec-lyon.fr
ED 160 EEA	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** https://edeea.universite-lyon.fr Sec. : Philomène TRECOURT Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr	**M. Philippe DELACHARTRE** INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 philippe.delachartre@insa-lyon.fr
ED 512 INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** http://edinfomaths.universite-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	**M. Hamamache KHEDDOUCI** Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 direction.infomaths@listes.univ-lyon1.fr
ED 162 MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** http://edmega.universite-lyon.fr Sec. : Philomène TRECOURT Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr	**M. Jocelyn BONJOUR** INSA Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69621 Villeurbanne CEDEX jocelyn.bonjour@insa-lyon.fr
ED 483 ScSo	**ScSo**[1] https://edsciencessociales.universite-lyon.fr Sec. : Mélina FAVETON Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr	**M. Bruno MILLY** (INSA : J.Y. TOUSSAINT) Univ. Lyon 2 Campus Berges du Rhône 18, quai Claude Bernard 69365 LYON CEDEX 07 Bureau BEL 319 bruno.milly@univ-lyon2.fr

[1] ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Thanks

First of all, I warmly thank Massih-Reza Amini, Professor at Université Grenoble Alpes, and Siegfried Nijssen, Professor at Université Catholique de Louvain, for agreeing to review this thesis. Their remarks, comments and fresh look at my work have been truely engaging. Also, I thank the jury of my thesis for their interest toward my work : Elisa Fromont, Charlotte Laclau, and my former university teacher Stephan Canu who contributed to my interset on Artificial Intelligence many years ago.

Then, I send my sincere appreciation to Céline Robardet and Marc Plantevit. Not only did they patiently directed my thesis, but they also taught me so much during those four years. The quality of their advices made me much more efficient in organizing my ideas, writing intelligibly and working rigorously. For I will keep these skills forever, I am greatly grateful to them. Additional credits goes to the rest of the DM2l team as well as Tias Guns, for providing fruitfull comments and guidance on my work.

Next, I deeply thank Erwan Pierre and Arthur Boidevezi, the opportunity of this thesis : I owe them both its creation and its success. They provided me the ideal work environment in BCMEnergy, in which I found more than colleagues: friends. Elie, Paul, Cécile, Juliette, Mathieu, thanks for bringing sunshine to my rainy days with your warm smiles. Extra thanks goes to my favorite fork companions, Matias, Valentin and Ege, who where always there to empty my head and fill my belly.

Those final words go to my family, that forged both my creativity and thoroughness. Apologies to my mom, for spending years of writing not a fantasy novel but a stodgy scientific thesis. At last, I dedicate those lines to my beloved Fiona, who always pushed me toward giving the best I could while bearing my silly jokes.

```
"Information is not knowledge.
Knowledge is not wisdom.
Wisdom is not truth.
Truth is not beauty.
Beauty is not love."
```

Frank Zappa.

Abstract

Electricity is essential for the energetic transition due to the diversity of greenhouse-gas free means of production and its potential to replace fossil fuels in transportation, heating and industries. However, it requires a constant balance between generation and consumption to maintain intensity in the network, and it can't be stored efficiently. It is then necessary to use Price Fixing Algorithm (PFA) for developing competitive markets. Daily, the European PFA EUPHEMIA determines the prices for the next day in Europe, called the Day-Ahead prices, that maximize the Social Welfare, while maintaining energy balance. Unlike other purely speculative markets, the Day-Ahead prices is algorithmically computed. Forecasting them is thus a unique and challenging task.

This introduces the problem of Electricity Price Forecasting (EPF) at the European scale, that consists in predicting the 24 hourly prices for each market before their fixation at 12am. The literature highlights two approaches: Expert models, that aim at replicating the PFA and computing the prices based on estimates of the inputs of EUPHEMIA, and Data-Driven methods that directly estimate prices using exogenous variables and past prices. Both approaches are incomplete: Expert models approaches are theoretically appealing but fail to produce accurate forecasts in practice. Conversely, Data-Driven approaches lack transparency, lowering the forecasts reliability. Also, the true relationship between variables and prices is only captured by EUPHEMIA, implicitly limiting the performances of Data-Driven approaches.

This thesis addresses those limitations. The first challenge is to produce accurate and explainable models for a given market. We achieve the former by extending methodologies from the literature, while we use Shap Values, a model-agnostic explainability tool, for the latter. Then, we build a multi-market forecasting model by representing the European network as a graph where each market is a node labeled with its prices. Graph edges are connection lines between markets, and we estimate the cross-market flows using an optimization problem prior to training. Lastly, we combine the EUPHEMIA algorithm with in a Neural Network (NN) that forecasts its inputs. To consider the price forecasting error in the NN's training, we compute the gradient of EUPHEMIA's output with respect to its input, by vanishing the derivative of the dual function using a dichotomy search.

We believe this thesis will be beneficial for the EPF practitioners and will contribute toward bridging the gap between Expert models and Data-Driven approaches. We also believe that our work on mixing optimization problems with machine learning models will benefit the broader scientific community.

Résumé

L'électricité est essentielle pour achever la transition énergétique grâce aux nombreux moyens d'en produire de manière décarbonée et à son potentiel pour remplacer les énergies fossiles dans les domaines des transports, du chauffage et de l'industrie. Cependant, afin de maintenir l'intensité dans le réseau, il faut constamment s'assurer que la production et la consommation soient égales. De plus, il est impossible de stocker efficacement de l'électricité. Il est donc nécessaire d'utiliser un mécanisme de fixation des prix pour développer des marchés compétitifs. Tous les jours, EUPHEMIA détermine les prix du lendemain en Europe, que l'on appelle les Prix Spot. Ces prix sont ceux qui maximisent le bien être social tout en maintenant l'équilibre dans le réseau. A l'inverse d'autres marchés purement spéculatifs, le prix de l'électricité est calculé ce qui rend son estimation indispensable pour beaucoup d'applications industrielles.

Le problème de l'estimation des prix de l'électricité à l'échelle Européenne consiste à prévoir les 24 prix horaires de chaque marché avant leur fixation à midi. De la littérature ressortent deux grandes familles d'approches : les modèles experts, qui ont pour but de répliquer EUPHEMIA et de calculer les prix en se basant sur des estimations des entrées d'EUPHEMIA, et les approches basées sur les données, qui utilisent les variables exogènes du marché pour directement estimer les prix. Les deux approches sont incomplètes : les modèles experts sont théoriquement intéressant mais très imprécis en pratique. A l'inverse, les approches se basant sur les données manquent de transparence, ce qui diminue la fiabilité de leurs résultats. De plus, la vraie relation entre les variables du marché et le prix n'est reflété que par EUPHEMIA, ce qui implicitement limite les performances des approches basées sur les données.

Cette thèse aborde ces limitations. Le premier défit est d'obtenir des prédictions suffisamment précises et transparentes pour un marché donné. La précision est obtenue en appliquant les méthodes basées sur les données de la littérature et la transparence en utilisant les valeurs de Shap, un outil d'explicabilité des modèles agnostiques. Ensuite, nous construisons un modèle de prévision multi-marché en représentant le réseau européen sous la forme d'un graphe, où chaque marché est un noeud qu'il faut labeliser avec ses prix. Les arrêtes du graphe sont les câbles connectant deux marchés, et nous estimons les flux d'énergie à l'aide d'un problème d'optimisation avant l'entraînement du modèle de prédiction des prix. Pour terminer, nous combinons l'algorithme EUPHEMIA avec un réseau de neurones qui estime ses entrées. Afin de considérer l'erreur de prédiction des prix durant l'entraînement du réseau de neurones, nous calculons le gradient du résultat d'EUPHEMIA par rapport à ses entrées, en trouvant le point où la dérivée de la fonction duale atteint zéro avec une recherche dichotomique.

Nous pensons que cette thèse sera bénéfique pour les professionnels de l'énergie re-

quérant des prédictions de prix de l'électricité, et qu'elle contribue à franchir le fossé qui sépare modèles experts et méthodes basées sur les données. Nous pensons également que nos travaux sur le mélange de problèmes d'optimisation avec des modèles d'apprentissage machine seront bénéfiques pour toute la communauté scientifique en général.

Contents

Chapter 1

Introduction

Electricity is a cornerstone of our current society. Historically overshadowed by fossil fuels, it grew in popularity during the last decades due to its versatility. It is perfectly adapted to many everyday-life objects. Think about it, how many electrically-powered items are present in your house? Or upon your desk? Recently, energy transition politicies and the urgent need to reduce atmospheric pollution in big cities fostered its usage in many sectors. Industries, transportation, heat, cooling systems, the so-called *electricity fairy* 1.1 conquered the world. Unfortunately, there is nothing magical about electricity and its immateriality and invisibility are sources of many constraints. Maintaining the tension in the network, transforming the intensity for the end users, producing an electrical current from heat, wind, steam, photovoltaic cells, was an incredible gift to the society made by scientists and engineer over the 20th century. Nowadays, other stakes are rising. How can we produce increasing amount of electricity in a fair, sustainable, and greenhouse gas free manner? Can we free ourselves from fossil fuel dependency? How can we maintain an affordable price and eliminate energy precariousness? This thesis lies in this line of interrogations and aim to contribute in answering them. In particular, it focuses on forecasting electricity prices on deregulated markets.

1.1 The electricity market

In the past, electricity markets were regionally monopolistic: all means of production belonged to the same actor which was also the only supplier. However, since the early 90's, competitive markets were introduced to end the hegemony of government-controlled electricity sectors. Moreover, electricity usage is increasingly important: according to the International Energy Agency, a additional 25TWh of electricity is consumed every year, making a total of 3711TWh in 2020 [1]. Those figures are due to the huge increase in use of electrical transportation, industries, and the democratization of households heaters (Figure 1.2, (LEFT)). Those different activities have various volume needs and are subject to intermittency. To support such a demand, a plethora of plant types is available. In Europe, the energy mix is dominated by thermal and nuclear plants that represent almost two thirds of the generation (Figure 1.2, (RIGHT)) and whose prices are driven by the price of their respective commodity. However, the share of renewables is strongly increasing and future

[1] `https://www.iea.org/regions/europe`, 23-10-2023

Figure 1.1 – *La fée electricité*, a giant painting presented by Raoul Dufy for the Universal Exposition of 1937 in Paris. Already in the first half of the 20th century, the artist underlined the diversity of electricity generation sources and consumption means, as well as the benefits of electricity to society.

policies will follow this trend [Far and Youngs, 2015]. Wind and solar plants, that constitute the bulk of renewable plants, are subject to heavy intermittency.

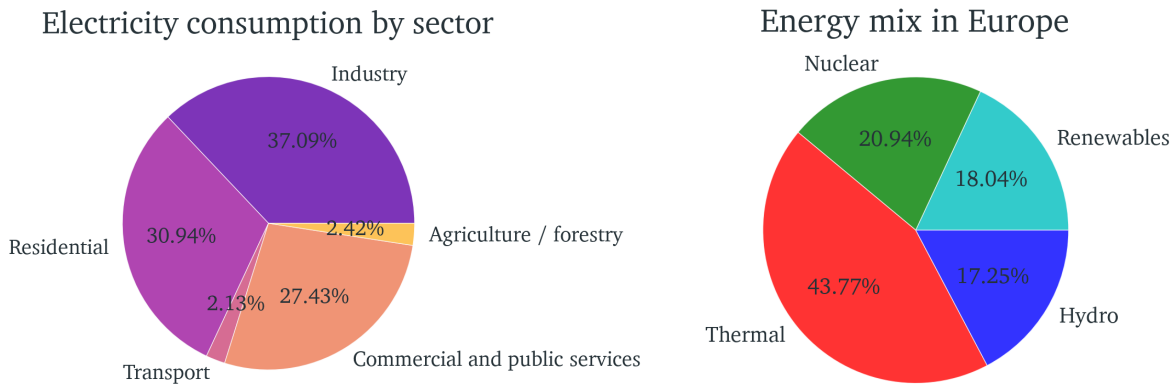


Figure 1.2 – (LEFT) The electricity consumption by sector in 2020 (RIGHT) The electricity sources in Europe for the year 2020.

The opening up to competition of energy generation and supply pushed forward the need for an open electricity market [IEA, 2001]. To satisfy the different business needs expressed by the market players, the European market uses a Price-Fixing Algorithm (PFA)

called EUPHEMIA [Committee et al., 2020]. Market actors can submit orders that describe their intention to buy or sell a fixed volume of energy for a given hour with a limit acceptable price. Orders are collected and an optimization problem is solved to determine the best possible price, by accepting or rejecting orders. This price is chosen to maximize the *social welfare* (supplier surplus and consumer surplus) and satisfy the *energy balance* (generated volume equals consumed volume). All accepted orders are executed using this determined price. One price per hour is computed everyday at noon for the 24 hours of the following day in every concerned country [2]. We denote the vector of those prices, called the *Day-Ahead Prices*, in zone z by $Y_z \in \mathbb{R}^{24}$. The task of forecasting $\widehat{Y}_z$ before price settlement is called Electricity Price Forecasting (EPF) [Weron, 2014]. Because of the regulation for submitting orders to the Price-Fixing Algorithm, the prices on the electricity market are not subject to speculation but aligned with the physical costs of generating plants. The Day-Ahead Price is also used to determine the price for exchanging energy at larger horizons. For instance, the cost of trading over a month is the average Day-Ahead Price over this month. The Price-Fixing Algorithm and Day-Ahead Price are therefore the pillars of electricity markets and understanding both is the key to many business opportunities.

1.2 Stakes and Applications

Energetic transition politicies brought new practices into the spotlight, aiming at favoring clean energy generation and responsible consumption, as well as increasing the quality of life of citizens. The Islander project [3], which aims to render the $31km^2$ island of Borkum (Germany) carbon-emission free by 2030, is a good example of such trends. The main challenge of EPF is that such applications are only profitable with an accurate price estimate. This thesis originated from the company BCMEnergy.

1.2.1 The Islander project

The island of Borkum (Germany), aims to become emission-free by 2030. Challenges of this project are multiple because of the consumption volatility due to summer tourism that multiply the population by five. The Islander project implements smart energy management systems for the island. The island possesses 2 wind turbines of 1.8MW each and a solar panel park of 1.67 MW, that provides around half of the yearly consumption. The other half is directly bought on the German market. The idea is, based on generation, consumption and price forecasts, to decide when it's best to sell the produced energy to the market or directly consuming it. To optimize these transactions, it's also possible to use batteries to buy when cheapest and sell when most expansive. Lastly, customers can be alerted about the cheapest times to use their devices. Price forecasts are therefore necessary for the success of such a project. The authors of [Hong, 2015] estimate that a small plant owner with a 5GW annual peak load can increase profits by 3 million USD by improvingthe accuracy of day-ahead price forecasts by 1%. Some of the findings of this thesis were used for the price prediction task of the Islander project.

[2] As some countries are divided into several areas with unique prices, we later use the term *zone* or *market*
[3] https://islander-project.eu/

1.2.2 BCMEnergy

To tackle these challenges, the company BCMEnergy was commissioned. Created in Lyon in 2015, BCMEnergy supplies green electricity to individual and professional consumers under the brand *elmy*. It stands out from other suppliers because of the variety of its commercial activities. For instance, it helps owners of generation plants maximizing their profits by bringing expertise in production forecasting, energy equilibrium and other aggregation services that make the transition between producers and transmission system operators. It also builds long-term partnerships between owners of fields, roofs or parkings by financing solar plants on those spaces. This upgrades wasted spaces, increases renewable capacities and provides profit to the owner. Additionally, it works with local communities to develop self-consumption projects. Such projects contribute to fighting energy precariousness and offer consumption tracking to reduce the electricity bill. Lastly, the activity that motivated this thesis is trading on the Day-Ahead market. The Day-Ahead prices are determined by the Price-Fixing Algorithm everyday at noon. Afterwards, all actors that submitted orders exchange energy at this fixed price. Players can also speculate on the Day-Ahead price on a separated platform. They can buy or sell energy between them, and at noon, if their energy balance is not equal to 0, they are settled using the Day-Ahead price. By estimating prices in advance, they can take advantageous positions on the market. An illustration is drawn in Figure 1.3, where using a price estimate $\widehat{Y}$ and a bid b, one can decide to sell (if $\widehat{Y} < b$) or buy (if $\widehat{Y} > b$). The reward is computed using the real price Y and is $Y - b$ in case the player bought, $b - Y$ if he sold.

The expertise on all links of the electricity value chain (from balance responsible to market access), the engagement in energetic transition and social concerns, as well as the aspiration to innovate and push boundaries made BCMEnergy an ideal environment to conduct this thesis.

1.3 Challenges

The EPF task presents many challenges. Some are caused by the diversity of data sources and explanatory variables of electricity prices, or by the chosen business application. The most important challenge for EPF is the modeling of prices. The variability in consumption and generation, combined with the absence of storage lead to strong seasonal components, high volatility and price spikes [Janczura et al., 2013]. Price fixation mechanisms cause non-linearity between supply and demand, which can only be understood through solving a Mixed-Integer Quadratic Programming (MIQP) [Taha, 2014] problem comprising thousands of orders of different types. Lastly, this algorithm computes prices across a continent where network topology plays a determining role.

1.3.1 Electricity

Inherently, electricity is an unsteady energy. Electrical systems requires a constant balance between consumption and generation. Balancing supply and demand implies turning on or off expensive power plants at the last minute or cutting-off consumption. Those mechanisms heavily affect price determination and can lead to price spikes. On top of that, the solutions

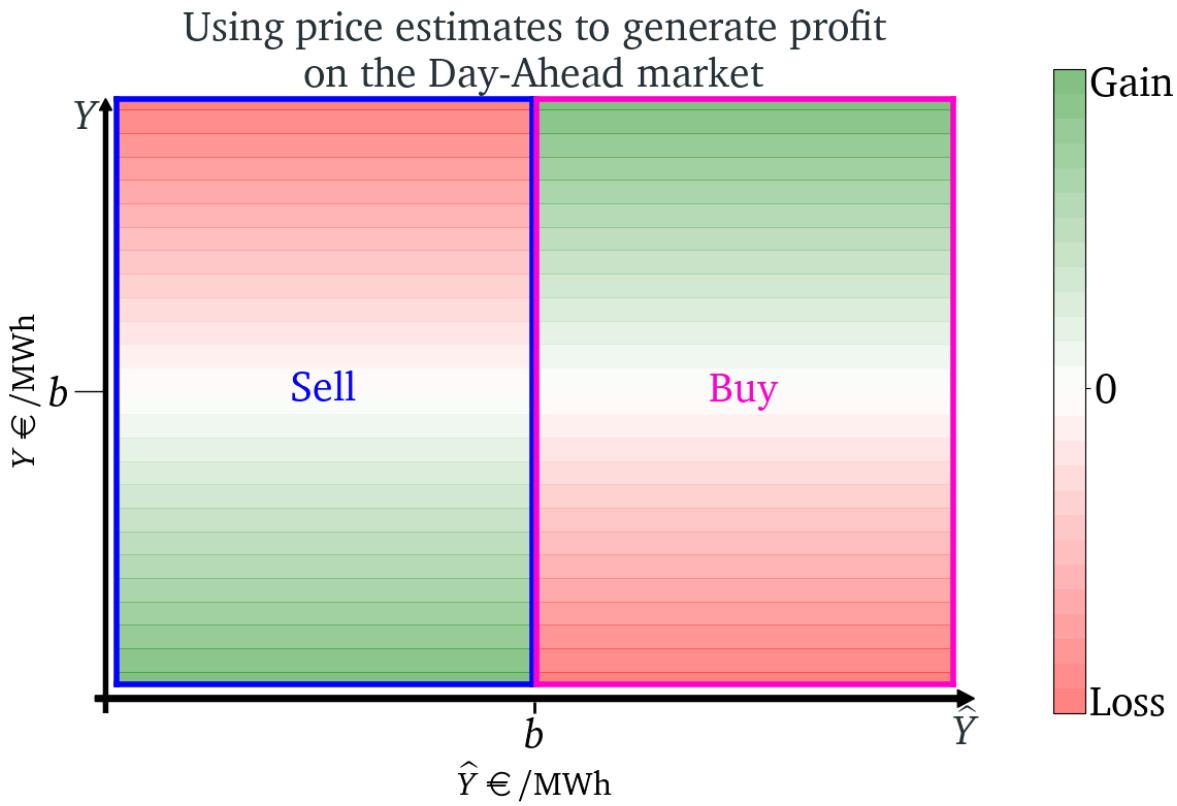


Figure 1.3 – Based on a price estimate $\widehat{Y}$ and a bid b, it's possible to decide between selling (if $\widehat{Y} < b$) or buying (if $\widehat{Y} > b$). Then, using the real price Y, one can compute the profit as the difference between b and Y.

for storing electricity (retention lakes, batteries) are not yet economically viable and the storage capacities are limited.

Unlike other commodities, electricity is always traded based on specific delivery period (hour, day, quarter, ...). Buying electricity for a given period t to sell it later at $t + \delta$ is not possible. It naturally follows that prices can highly diverge from one period to another since the price of period t is not used for pricing $t + \delta$. Prices are determined by matching supply and demand from the orders of market actors. The supply side reflects the marginal costs of power plants. In European countries where the energy mix is eclectic 1.2, this means that the electricity price is dependant on the prices of other commodities (gas, fuel, coal). Added to this fact, all generation means are subject to unpredictability such as material failures, maintenance operations or strikes. The demand side is mostly regulated by regional governments through tariff shields but legislation can differ across borders. Moreover, the demand volume, driven by weather conditions, is hard to foresee. The set of regulations that followed the Paris agreement [Agreement, 2015] are necessary to achieve energetic transition, but it leads to even more volatile consumption, generation and price of electricity. Usage of electric cars has been promoted and self consumption allows private individuals to directly use their own produced energy, making the consumption patterns more complex. Moreover, the part of renewables in the energy mix in Europe must reach one third in 2030.

To achieve it, the capacity of solar and wind plants is increasing, making the generation weather-dependent, and thus, fickle and unpredictable.

Generally, energy transition policies introduce regulations that exacerbate price volatility by reducing the number of tractable plants (nuclear, fossil plants) or introducing taxation of carbon dioxide emissions [4]. The price of carbon allowances, necessary to compute marginal costs of fossil plants, is another variable that explains electricity prices. Lastly, unpredictable worldwide events such as economic crises of pandemics can heavily affect prices. In particular, the last 3 years where marked by repeated lockdowns and economic recovery that caused prices to reach unprecedented levels [Narajewski and Ziel, 2020b; Suvarna et al., 2022].

The intermittency of generation and consumption, as well as the impossibility for efficient storage and the required balance between supply and demand make the spot prices prone to spikes, seasonal trends, and high volatility.

1.3.2 The Price-Fixing Algorithm

On the European market, Day-Ahead Prices are computed every day at noon by the EUPHEMIA algorithm. It is a MIQP algorithm that maximizes *social welfare* and ensures *energy balance* across concerned zones. Market actors (suppliers and producers) have to submit Order Books that describe how much they are willing to buy or sell and at which price. EUPHEMIA computes which orders are accepted and deducts the Day-Ahead Price. It is the price that generates the most *social welfare*, defined as the sum of consumers' surplus and suppliers' surplus. All accepted orders are carried out using this price.

Order Books are constituted of thousands of orders per trading hour. Orders can be of several levels of complexity. Simple orders are defined on a single hour, by a volume, and a limit price (step order) or a price range (linear order). Complex orders can span across several hours (block orders), have special activation conditions (minimum income conditions orders) or be linked to other orders (grouped orders). It results that the computed price has a complex relationship with consumption and generation because of the Order Book structure. In Figure 1.4, we only displayed step orders. The Day-Ahead Price $P^\star$ lies on the intersection of the supply and demand curves. If we focus on this intersection point, we see that demand could be lowered by 35 MWh without modifying the intersection price (red dotted line). On the other hand, a drop in generation of only 5 MWh would increase the price by 10€/MWh (blue dotted line).

Order Books enable market players to develop profitable business and implement their trading strategies, while the EUPHEMIA algorithm ensures a fair price for every actor and that energy balance is satisfied. However, the diversity of orders, the size of the Order Books, and the complexity of EUPHEMIA algorithm make the price-fixation mechanism hard to grasp. On the other hand, correctly understanding it is the key to mastering electricity price forecasting.

1.3.3 The European network

EUPHEMIA operates on 39 European markets, with some countries like Italy or Norway having multiple markets as shown in Figure 1.5. To compute optimal prices, the *social welfare* is maximized and the *energy balance* ensured across all zones. Each market has its

[4] https://climate.ec.europa.eu/eu-action/eu-emissions-trading-system-eu-ets_en 23-10-

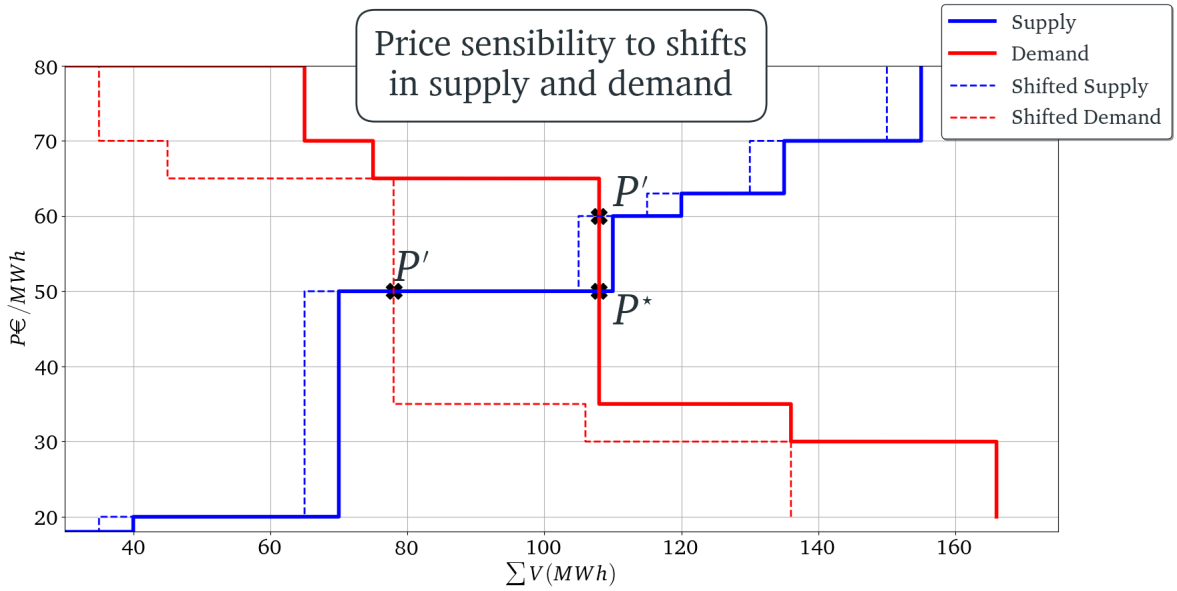


Figure 1.4 – $P^\star$ is the price resulting of the intersection of the supply (RED) and demand (BLUE) plain curves. An important shift in Demand can occur (red dotted line) without modifying the price, whereas a small supply shift drastically increases the price.

own specific characteristics and is subject to different external conditions. This results in important differences in supply, demand and prices between the different markets. More importantly, differences in the relationships between characteristics and prices can occur. Then, EUPHEMIA assigns energy flows from cheaper countries to the more expansive. This usually helps balancing energy for a cheaper price, but leads to several complexifications. First, the energy has to be balanced at the scale of a continent, that involves managing energy flows. The flows are themselves constrained by the Available Transfer Capacities (ATC), that is the maximal volume that can be sent from one zone to another for a given time period. Then, the prices of a given zone becomes dependant on variables from other zones. Also, many pairs of zones don't have electrical lines that directly connect them. For instance, Spain and Germany are not linked. To send energy from Spain to Germany, the connections Spain France and France Germany can be used. The topology of the European network (the map of all zones and their connections), is another characteristic of the market to be taken into account. Lastly, this increases the number of orders by considering all 39 Order Books and adds another layer of complexity to the Euphemia algorithm. In practice, this also makes the prices from a country dependant on its neighbors characteristics, potentially to all considered zones.

1.3.4 Applications of price forecasts

The industrial context in which electricity price forecasts are used should be taken into account when designing an EPF model. Electricity prices are dependent on many factors, and huge quantities of data from several providers are available. An appropriate tool for market players should handle data collection, processing of missing data, and forecast

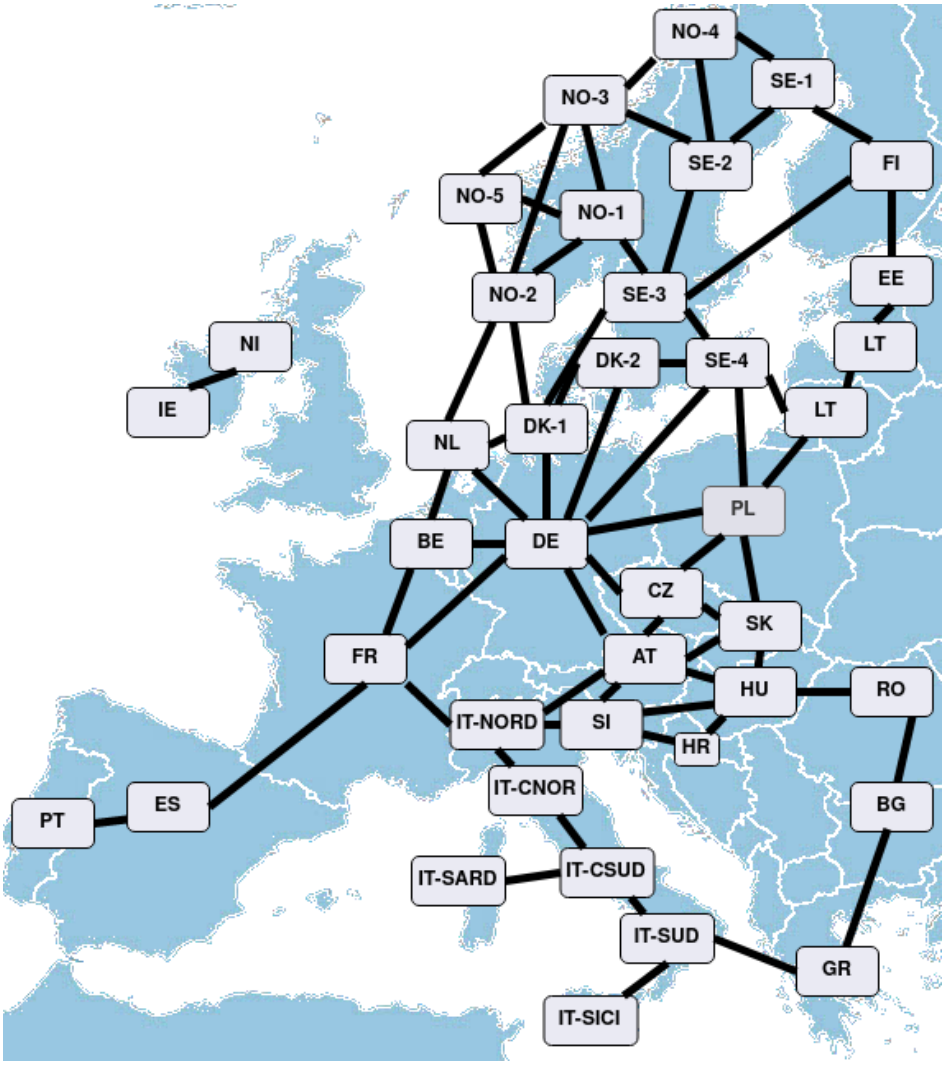


Figure 1.5 – The map of the markets considered by EUPHEMIA. Countries can be divided into several price zones (Italy, Norway). Power lines between zones are summarized by black lines.

publication. Next, the business application in which price forecast are going to be used introduces constraints. For instance, if the objective is to optimize the use of a battery, it's better to forecast the shape of the hourly price curve rather that the actual values. Indeed, the aim is to buy when cheapest and sell when most expensive. A plant owner would benefit from a prediction for several days in advance to adjust his generation plan.

The use case that motivated this thesis is trading on the day-ahead market. For this, only the daily average price is needed, but its accuracy is paramount. Moreover, traders must react quickly to generate more profit, which constrains the response time of the models. Traders also benefit from detailed explanations about a forecast to take a more refined decision. Knowing which input variable led to such forecast allows them to consider data that wasn't part of the forecasting process due to lack of historical datasets or unavailability at

prediction time (impromptu unavailability of a unit, sudden change in weather, unreported strike, etc...). It also helps the trader to be more confident when taking a position on the market, with as consequence bidding more often with a growing volume, increasing the potential profit.

1.4 Contributions

The abundance of data collection tools and computing power has brought Artificial Intelligence to an unprecedented level of efficiency since the 90's. Rapidly during the early 2000's, research interest in this field exploded and thousands of papers, models, architectures were explored. However, by going further into Machine Learning research, scientists and engineers collided with its limitations. In this thesis, we propose methods that tackle challenges induced by the EPF task that lies at the edge of the ML methods. Our contribution to the literature is based on three articles, with two of them presented in the scope of international conferences.

1.4.1 Explaining the forecasts

One of the main drawbacks of Machine Learning tools is their black-box characteristics. The output is hard for users to understand, and sometimes a transparent solution is preferred even at the expense of performance. The EPF problem can be reasonably addressed using simple methods such as Linear Regression or Auto-Regressive models, that offer the advantage of producing understandable results. Once fitted, the coefficients of a Linear Regression give a direct value of the weights of each feature in the forecast and users have a simple mathematical formula to compute the model's output. This is sufficient to make businesses practitioners choosing a simpler model over a ML model.

In this thesis, we address the problem of explaining electricity price forecasts. We make use of findings from the eXplainable Artificial Intelligence (XAI) field that aims at lightening the opacity of ML models. Particularly, we use the Shapely values [Lundberg and Lee, 2017] which are based on well-known game-theory ideas of [Shapley, 1953]. The forecast is the reward of a N-person collaborative game where each feature is a player. Computing the exact contribution of each feature is too expensive as it requires computing 2^N model predictions. [Lundberg and Lee, 2017] thus propose a sampling method to estimate them with reduced complexity. We show how a feature contribution analysis is able to bridge the gap between model performance and user confidence. We identify features responsible for performance increase and discard the less important ones. By analysing the impact of the Covid crisis on the model performance, we show that when using an appropriate recalibration method, EPF models can adapt to new situations even in extreme market conditions. This work has been published in the journal Applied Energy [Tschora et al., 2022].

1.4.2 An optimize-then-predict Graph Neural Network

ML models have to be adapted to work with non-tabular data. For example, Convolutional Neural Networks, which where first introduced in [LeCun et al., 1995], have been extended to graph structures in [Gilmer et al., 2017]. Graphs have been widely used for a variety of problems (assignation, optimal path, etc...) and combining them with ML models has led to

significant progress in applications such as drug discovery or traffic forecasting. Formally, a graph is a set of nodes (or vertices) that can be linked together by one or more edges. Each node and edge usually contains information (time series, tabular data). A common framework in traffic forecasting [Epelbaum et al., 2017] is to consider a sequence of graphs. Each graph describes the state of the road network at a given time.

We employ an analoguous representation for the European electricity network. Each zone is represented by a node, connections between zones are edges, and the model is trained to label each node by the Day-Ahead Prices. Different values of edge attributes are assessed and we show that using estimated cross-market flows with an optimization problem prior to learning significantly contributes towards increasing the model's performances. Up to our knowledge, this approach is the first to use a GNN model for EPF, and the only one that considers the European Network integrally. Moreover, the optimize-then-predict approach shows that even powerful ML models can benefit from optimization and suggests that interaction between optimization and model training should be delved into. This work [Tschora et al., 2023a] has been presented during the IDA 2023 conference [5] where it was awarded the Frontier Prize.

1.4.3 Considering EUPHEMIA in a *Decision-Focused Learning* framework

ML models are very efficient modeling tools, but a vast domain of real-world problems requires the enforcement of constraints that those models can't guarantee. Constrained Optimization is typically used for such problems, but sometimes, coefficients of the optimization problem are unknown and have to be estimated. Some researchers are trying to link both worlds [Amos and Kolter, 2017] and fit coefficient forecasting models that yield the best solution with respect to the optimization problem.

Electricity prices are computed by EUPHEMIA, which essentially solves a MIQP problem. The *social welfare* which is maximized is a function of the Order Books (which are coefficients of the problem) and the acceptance of the orders (which are the decision variables). In the EPF task, Order Books are not available prior to price publication : they have to be estimated. We build an Order Book forecasting Neural Network (NN), followed by the EUPHEMIA algorithm. EUPHEMIA is solved using Order Books estimates and as a way to ensure that the model lowers the price forecasting error overall, we take into account the solution of EUPHEMIA in the gradient descent. This requires differentiating through a quadratic optimization problem. We achieve the latter by considering the dual problem that we solve by setting its derivative to 0. The search for an optimal solution is performed using a dichotomy search where the gradients of each step are retained. This approach is the first to consider EUPHEMIA along with a NN model and constitutes an important tool for EPF practitioners because of its consistency with respect to the Price-Fixing Algorithm. Moreover, our approach to differentiable optimization using a dichotomy search is a novelty that contributes in the objective of mixing optimization problems and Machine Learning models. This work was presented during the DSAA 2023 conference [Tschora et al., 2023b] [6].

[5] https://ida2023.org/

[6] https://conferences.sigappfr.org/dsaa2023/

1.5 Thesis structure

This thesis articulates around 4 chapters. In the first Chapter 2, we present the state of the art in electricity price forecasting literature. Particularly, we focus on the limitations of the two main approaches. Expert models aim at replicating Price-Fixing Algorithm and Data-Driven methods that learns relationships between the Day-Ahead Prices and the input data. We conclude the Chapter by overviewing the existing methods for combining an optimization problem and a Machine Learning model.

In Chapter 3, we focus on tuning standard Machine Learning models for 3 datasets of interest: France, Germany and Belgium. We compare our results to the state of the art models and find that simpler models (Support Vector Regressors) can outperform more sophisticated ones (Deep Neural Network) if using the appropriate features. Using Shap Values, we also discuss the importance of each feature with respect to the market characteristics and analyze the effects of the Covid crisis in the model accuracy.

Chapter 4, is dedicated to bringing the problem to the European scale by using a graph structure to capture the topology of the network. We investigate how to enrich the edge attributes by designing a cross-market flow estimation problem using four different optimization problems. We thus use an optimize-then-predict framework that first computes optimal flows between markets, then predicts prices for all considered zones. By analyzing metrics and feature contributions, we explain the price forecasting error differences between flow estimation methods but also depending on the position of the node in the graph.

Then, in Chapter 5, we introduce our differentiable optimization approach for EPF. We start by solving EUPHEMIA using a differentibale dichotomy search, which is placed on top of an Order Books forecasting Neural Network. Because of the differentiability of the dichotomy search, the NN is fit by considering the price forecasting error rather than the Order Books error. We analyse our results by varying the weights of the optimization loss in the training process, and, using Shap values, identify cases where the differentiable optimization approach outclass the standard approaches.

Chapter 6 summarizes the work accomplished during this thesis. We then forecast our own future work by introducing several research tracks.

1.6 Publication list

Peer-reviewed international conferences with proceedings:

- **Léonard Tschora**, Erwan Pierre, Marc Plantevit, Céline Robardet. *Forecasting Electricity Prices: An Optimize Then Predict-Based Approach*. In: Crémilleux, B., Hess, S., Nijssen, S. (eds) Advances in Intelligent Data Analysis XXI. IDA 2023. Lecture Notes in Computer Science, vol 13876. Springer, Cham. `https://doi.org/10.1007/978-3-031-30047-9_35`

- **Léonard Tschora**, Tias Guns, Erwan Pierre, Marc Plantevit and Céline Robardet, *Electricity Price Forecasting based on Order Books: a differentiable optimization approach*, 2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA), Thessaloniki, Greece, 2023, pp. 1-10, doi: 10.1109/DSAA60987.2023.10302542

International journals:

- **Léonard Tschora**, Erwan Pierre, Marc Plantevit, Céline Robardet. *Electricity price forecasting on the day-ahead market using machine learning* Applied Energy 313 (2022): 118752.

Chapter 2

State of the Art

This chapter introduces the technical and literature background of this thesis. The first part introduces general notations, formalizes the EPF problem, and explains the EUPHEMIA optimization problem. The second section presents the most popular EPF methods from the literature. We distinguish between the Data-Driven methods, which learn relationships between price and variables from historical datasets, and Expert models, where the relationships are handcrafted by practitioners who approximate the Price Fixing algorithm (PFA). In a third section, we motivate the need for developing more sophisticated approaches that combine both approaches by showing the limits of the literature. To this aim, we present *Constrained Optimization Learning* (COL) methods in the fourth section. They consist in solving optimization problems with coefficients estimated by a ML model. The general idea is to forecast the input of EUPHEMIA using a ML model before solving the optimization problem.

2.1 Preliminaries

Before deepening the electricity price forecasting methods of the literature, we specify the notations that are used throughout this document. Also, we describe the most important parts of the EUPHEMIA algorithm.

2.1.1 The Electricity Price Forecasting Problem

The Day-Ahead price of a day d, an hour h, and a zone z is denoted by $Y_z^{(d,h)} \in \mathbb{R}$. $Y_z^{(d)} \in \mathbb{R}^{24}$ is the vector of the 24 hourly prices. Everyday $d-1$ at noon, $Y_z^{(d)}$ are computed by the EUPHEMIA algorithm. The EPF problem is a multivariate time series task that consists of forecasting $Y_z^{(d)}$ for one or more z, before the price computation the day before at noon $(d-1, 12)$. To achieve this, it's possible to use past day-ahead prices $Y_z^{(d-1)}$, thus considered as input features. A graphical illustration of the Day-Ahead price forecasting task is available in Figure 2.1. Other usual exogenous features mainly consist of estimates of the fundamentals of a day d, available in $d-1$: consumption forecasts $C_z^{(d)} \in \mathbb{R}^{24}$, renewable generation forecasts $R_z^{(d)} \in \mathbb{R}^{24}$, programmable generation forecasts $G_z^{(d)} \in \mathbb{R}^{24}$. The set of variables used to predict a day d of a zone z is noted X_z^d. For clarification, if only one market is considered, the notations Y^d and X^d are preferred. Lastly, when referring to a collection of days of size n, letters X and Y are simply used.

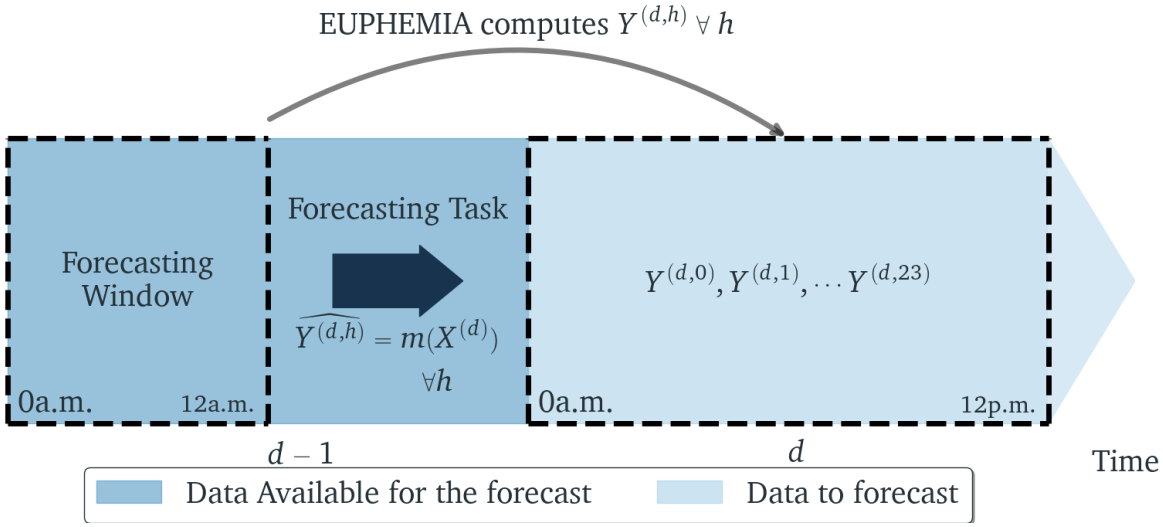


Figure 2.1 – The Day-Ahead prices prediction task. The 24 hourly prices of a day d must be forecast on $d-1$ before 12 a.m.

2.1.2 EUPHEMIA

The full documentation of the EUPHEMIA algorithm is available in [Committee et al., 2020]. In this section, to give an overview of the complexity of this algorithm, we detail its major components. The prices $Y_z^{(d,h)}$ are computed by the EUPHEMIA algorithm for all zones z and

hours h of a day d at once, using daily Order Book OB_z^d as input. Order Books are done on a daily basis as they contain orders that span over several hours. We summarize EUPHEMIA by a function $\mathcal{E}$ that maps elements from the Order Books domain Φ to the Day-Ahead prices:

$$\mathcal{E}: \Phi^{n_z} \mapsto \mathbb{R}^{24\times n_z}$$
$$OB^d \to \mathcal{E}\left(OB^d\right) = Y^{(d)}$$
$$\begin{pmatrix} OB_1^d \\ \cdots \\ OB_{n_z}^d \end{pmatrix} \to \begin{pmatrix} Y_1^{(d,1)} \cdots Y_1^{(d,24)} \\ \cdots \\ Y_{n_z}^{(d,1)} \cdots Y_{n_z}^{(d,24)} \end{pmatrix}$$

The general idea behind EUPHEMIA is to maximize all zones' *social welfare* (SW) while satisfying the constraints induced by the network or by the player's orders, by selecting which orders to accept using decision variables $A^d \in [0,1]^{n_z \times |OB_z^d|}$ and by determining the necessary energy exchanges between zones $F^d \in \mathbb{R}^{24\times n_z \times n_z}$. For simplifying the notations, we employ OB, A and F to designate the Order Books, Acceptance variables and Cross-Zonal Flows of an arbitrary day.

$$\underset{A,F}{argmax}\ SW(OB,A,F)$$
$$\text{u.c.} \begin{cases} \text{Network Constraints}(OB,A,F) \\ \text{Order Constraints}(OB,A,F) \end{cases}$$

The *social welfare* is defined as the sum of consumer and supplier economic surplus, which can be written as the sum of SW of each order of OB. In the *social welfare* computation, the profit realized by energy line owners (Congestion Rents CR) is also taken into account. This profit is simply the amount of energy $F_{z,z'}^h$ sent through the line times the tariffs $T_{z,z'}^h$ of the line.

$$SW(OB,A,F) = \underbrace{\sum_{O\in OB} SW(O,A)}_{\text{Consumer and Supplier economic surplus}} + \underbrace{\sum_h \sum_z \sum_{z'} F_{z,z'}^h T_{z,z'}^h}_{\text{Congestion Rents}}$$

On the market, several players can be found: suppliers, owners of small renewable plants, owners of important thermal or nuclear power plants, etc. Each player has its own requirements and constraints that need an appropriate bidding structure. For instance, an owner of a coal-fired power-plant has to be sure that he will be able to sell its production above its marginal cost for several consecutive hours, enough to cover its start-up and shut-down costs. Also, the order should account for the physical requirement of the plant such as the start-up time, the maximal generation capacity or the possible volume variation each hour. As such, several types of orders are available and we give the expression of SW and constraints of the four most common ones: Step s, Linear l, Block b and Complex c Orders. We identify each order by its type $t \in \{s,l,b,c\}$, ID i, zone z and hour h.

- STEP ORDERS s are defined by a fixed volume $V_{s,i,z}^h$ and a limit price $P_{s,i,z}^h$ on a given hour h. They can be fully accepted, fully rejected or partially accepted: $A_{s,i,z}^h \in [0,1]$.

$$SW(O_{s,i,z}^h) = A_{s,i,z}^h V_{s,i,z}^h P_{s,i,z}^h$$

- Linear Orders l express the amount of volume to exchange as a linear function of the price, on a fixed price range $[Po^h_{l,i,z}, Po^h_{l,i,z} + P^h_{l,i,z}[$ on a given hour h. They can be fully accepted, fully rejected or partially accepted: $A^h_{l,i,z} \in [0,1]$:

$$SW(O^h_{l,i,z}) = A^h_{l,i,z} V^h_{l,i,z} \left(Po^h_{l,i,z} + A^h_{l,i,z} P^h_{s,i,z} \right)$$

  It is important to notice that the SW expression of Linear Orders is quadratic of the Optimization variable $A^h_{l,i,z}$.

- Block Orders have a volume $V^h_{b,i,z}$ that can vary each hour h over a period H with a fixed price $P_{b,i,z}$. The acceptance of such an order is defined as the sum of accepted volume $A_{b,i,z} = \sum_{h \in H} A^h_{b,i,z}$, that must be higher than a submitted Acceptance Ratio $AR_{b,i,z}$: $A_{b,i,z} \geqslant AR_{b,i,z}$. Players that submit a Block Orders are usually nuclear or thermal plants owners that can't shut down their plants once turned on. As such the order that they submit must ensure them that turning on the plant will be profitable, i.e. that they can sell energy above the marginal cost during several hours. They can also submit Linked Blocks (LB) that are a set of Block Orders with Acceptance of elements of the set dependent on the Acceptance of others. Lastly, Exclusive Groups EG are also sets of Block Orders, with the constraint $A_{b,i,z} \leqslant 1$. The Social Welfare is defined as

$$SW(O_{b,i,z}) = A_{b,i,z} P_{s,i,z} \sum_{h \in H} V^h_{b,i,z}$$

- Complex Orders are a set CO of Step and Linear orders whose *social welfare* is the sum of SW of all orders of the set:

$$SW(O_{c,i,z}) = \sum_{o \in CO} SW(o)$$

  Players submit Complex Orders with a Minimum Income Condition (MIC) constraint that stipulates that its profit should exceed a certain value, usually a fixed term (start-up cost C_{start}) and a variable term linear of the sold volume (marginal cost C_{marg}) :

$$\sum_{o \in CO} SW(o) \geqslant C_{start} + C_{marg} \sum_{o \in CO} A_o V_o$$

  Scheduled Stop conditions modify the MIC so that plant owners don't have to abruptly shut down their plants on the first hour of a new auction. Lastly, Load Gradient constraints make sure that volume differences from an hour to another do not exceed a threshold $\delta_{b,i,z}$: $A^h_{b,i,z} V^h_{b,i,z} \in [A^h_{b,i,z} V^h_{b,i,z} - \delta_{b,i,z}, A^h_{b,i,z} V^h_{b,i,z} + \delta_{b,i,z}]$.

The network constraints are the following:

- Available Transfer Capacities (ATC) are the maximum amount of energy that can be sent through a line :

$$F^h_{z,z'} \leqslant \Gamma^h_{z,z'}$$

for all connections (z, z') and hours h, with $\Gamma^h_{z,z'}$ the capacity of exchange between z and z' at hour h.

- Flow-Based Capacities (FBC) are another way of expressing the constraints of energy flows between countries, by considering critical branches of the network. Critical branches are significantly influenced by the cross-border allocation and have a risk of being constrained due to network security reasons. Rather than considering connections separately, a Power Transmission Distribution Factor (PTDF) matrix is used to describe how a commercial exchange would affect physical flows on a given critical branch of the network. Each critical branch is a row of the PTDF matrix, and each country is a column: $PTDF \in \mathbb{R}^{n_{cb} \times n_z}$. The determined energy flows $F \in \mathbb{R}^{n_z}$ shall respect the following constraint : $PTDF.F \leqslant RAM$ with the Remaining Available Margin RAM is the maximal capacity of the critical branch minus a safety flow reliability margin. This method is only employed for specific bidding zones of the European Network and increases *social welfare* overall by determining more precise flows.

- Energy Balance stipulates that supply and demand should match. In other words, the sum of accepted volumes from all order types plus the sum of incoming and outgoing energy flows should be null.

$$\sum_{t \in \{s,l,b,c\}, i \in t} A^h_{t,i,z} V^h_{t,i,z} + \sum_{z'} F^h_{z,z'} - \sum_{z'} F^h_{z',z} = 0$$

  for all zones z and hours h. Note that we consider the volumes of supply orders as positive and of demand orders as negative to simplify the notations.

Let's summarize. euphemia has to determine the decision variables:

- Acceptance of Orders A
- Cross-Zonal flows F

that best maximize the sum of :

- SW from Step Orders
- SW from Linear Orders
- SW from Block Orders
- SW from Complex Orders
- SW from Congestion Rents

under the constraints:

- Acceptance Ratio of Block Orders
- Linked Block Orders
- Exclusive groups of Block Orders
- Minimum Income Condition of Complex Orders

- Scheduled stop of Complex Orders
- Load Gradient of Complex Orders
- Available Transfer Capacities
- Energy Balance

Once A and F are computed, it's possible to determine the Day-Ahead Prices Y by taking the most expensive accepted supply order or the least expensive accepted demand order $Y_z^h = \max_{t\in\{s,l,b,c\},i\in t} P_{t,i,z}^h * sign(V_{t,i,z}^h)$. Accepted orders also tell plant owners when to turn on, how many volume to produce, and individual suppliers at which price they buy. Note that once Y_z^h is computed, all exchanges are conducted using Y_z^h, not $P_{t,i,z}^h$ (i.e. all suppliers sell at Y_z^h whatever their marginal costs and all consumers buy at Y_z^h).

2.2 Electricity Price Forecasting Models

The subject of EPF has attracted numerous scientists since the successive market deregulation in the 90s. The complexification of price-fixation algorithms, the increase of renewable share in the energy mix and the opening up to competition of individual supply explain the growing popularity of EPF. A plethora of methods have been studied, that can be split in two categories. Traditionally, the same models were used for any commodity market. They consisted in using expert domain-knowledge to simulate the market's rules and logic. Oppositely, Data-Driven approaches are based on using historical datasets to extract relationships between prices and market characteristics. In this section an overview of the most important EPF models found in the literature is given.

2.2.1 Data-Driven methods

Once sufficient amount of data became available, Data-Driven methods revealed to be the most efficient for modeling electricity prices [Weron, 2014]. They consist in learning relationships between the prices and its past occurrences or exogenous variables using an historical dataset. Only a few variables are necessary to fit a model that forecasts prices with more accuracy than naive benchmarks, and Auto-Regressive (AR) methods can perform well using only the price time series as input [Lago et al., 2018a].

2.2.1.1 Naive methods

The most common benchmark is to use the previous day's price as the forecast $Y_z^{(d-1)}$ for the current day. Variants capture the weekly seasonality of the prices by forecasting $Y_z^{(d-1)}$ if d is a working day (except Monday), and $Y_z^{(d-7)}$ if d is a week-end day or a Monday.

Similar day methods [Nogales et al., 2002; Shahidehpour et al., 2003] seek a day with similar characteristics to the one to predict based on external variables (generation forecasts, weather, etc...) and uses the prices of this day as forecasts. They constitute a solid naive regressor [Contreras et al., 2003; Nogales et al., 2002]. In combination, some contributions used exponential smoothing where the prediction is an exponentially weighted average of past observations [Cruz et al., 2011].

Lastly, for the German market, the EXAA auction can be used as a benchmark [Ziel et al., 2015b]. This exchange is a secondary market in Germany whose prices are published at 10am. They are thus possible to include in a price forecasting model and give a good overview of the German price.

2.2.1.2 Statistical models

Regression models consist of expressing the target as a parameterized mathematical expression of the predictive variable. The simplest expression is the Linear Regression:

$$\widehat{Y^{(d,h)}} = \beta_0 + \beta_1 \cdot \begin{pmatrix} X_1^{(d,h)} \\ X_2^{(d,h)} \\ \cdots \\ X_n^{(d,h)} \end{pmatrix} + \epsilon^{(d,h)}$$

with n input variables X and white noise $\epsilon^{(d,h)}$. Coefficients $\beta_1 \in \mathbb{R}^n$ and the bias β_o is estimated by minimizing the differences between predicted and observed prices (least square methods) [Karakatsani and Bunn, 2008]:

$$\widehat{\beta} = \underset{\beta}{argmin} ||Y - \widehat{Y}||^2$$

The most common variant of regression models is the Auto-Regressive (AR) model, that takes into account the time correlations between prices. The input variables are past occurrences of the prices $X^{(d,h)} = (Y^{(d-1,h)}, Y^{(d-2,h)}, \cdots, Y^{(d-w-1,h)})$ (with a window length of w). A Moving Average term $\beta_2 \cdot \epsilon$ can be added to this expression, where $\epsilon \in \mathbb{R}^q$ is a white noise term errors and β_2 new parameters to estimate. AutoRegressive Moving Average (ARMA) models have been used for EPF in [Cuaresma et al., 2004; Misiorek et al., 2006].

When $X^{(d,h)}$ contains both past prices and other variables, the term Auto-Regressive using exogenous characteristics (AR-X) is employed [Contreras et al., 2003; Lira et al., 2009; Nogales and Conejo, 2006; Nogales et al., 2002; Zareipour et al., 2006].

Threshold Auto-Regressive (TAR) models [Bunn, 1985; Gonzalez et al., 2011; Misiorek et al., 2006] include a regime-switching logic. The regimes are switched if an observable variable $X_t^{(d,h)}$ exceeds a threshold T:

$$\widehat{Y^{(d,h)}} = \begin{cases} \beta_1 \cdot X^{(d,h)} + \epsilon^{(d,h)} \text{ if } X_t^{(d,h)} > T \\ \beta_2 \cdot X^{(d,h)} + \epsilon^{(d,h)} \text{ otherwise} \end{cases}$$

Lastly, The Generalized AutoRegressive Conditional Heteroskedastic models (GARCH) [Diongue et al., 2009; Knittel and Roberts, 2005] aim to estimating residuals $\epsilon^{(d,h)}$ by representing the conditional variance σ^2 of the time series by an autoregressive process. It addresses the problem of heteroskedasticity (that occurs when variables have different variance) :

$$\epsilon^{(d,h)} \sim N(0, \sigma^2)$$

$$\sigma^{(d,h)^2} = \beta_1 \cdot \begin{pmatrix} \epsilon^{(d,h)^2} \\ \cdots \\ \epsilon^{(d-\omega_1,h)^2} \end{pmatrix} + \beta_2 \cdot \begin{pmatrix} \sigma^{(d,h)^2} \\ \cdots \\ \sigma^{(d-\omega_2,h)^2} \end{pmatrix} + \omega$$

with different calibration windows ω_1 and ω_2, and a fixed term $\omega > 0$.

The simplicity and straightfowardness of these approaches explain their popularity. Moreover, statistical methods are attractive to price forecasters because of the possibility to link regression coefficients to market characteristics. While this provides explainability tools for the users, it also introduces a drawback. Forecasters heavily rely on expert knowledge to select the variables [Misiorek et al., 2006; Ziel and Weron, 2018]. A well-known outcome is the so-called *Expert model* [1], defined as :

$$\widehat{Y^{(d,h)}} = \beta_1 \cdot \begin{pmatrix} Y^{(d-1,h)} \\ Y^{(d-2,h)} \\ Y^{(d-7,h)} \end{pmatrix} + \beta_2 \cdot \begin{pmatrix} Y^{(d-1,24)} \\ Y^{(d-1,max)} \\ Y^{(d-1,min)} \end{pmatrix} + \beta_3 \cdot X^{(d,h)} + \beta_4 \cdot D$$

with auto-regressive effects ($Y^{(d-1,h)}$, $Y^{(d-2,h)}$, $Y^{(d-7,h)}$) associated to coefficients $\beta_1 \in \mathbb{R}^3$, the last, maximum and minimum prices from the day before $Y^{(d-1,24)}$, $Y^{(d-1,max)}$ and $Y^{(d-1,min)}$ associated to $\beta_2 \in \mathbb{R}^3$, the exogenous variables $X^{(d,h)}$ to β_3, and D date dummies to β_4. Solving the problem of automatic regressor selection for EPF has been tackled after R. Weron 2014's review in the papers [Ludwig et al., 2015; Uniejewski et al., 2016; Ziel, 2016; Ziel et al., 2015a]. It consists in adding a regularization term of order q to the regression : $\lambda \sum_{i=1}^{n} |\beta_i|^q$. For $q = 1$, some β can be set to 0 and eliminates regressors from the model. This feature grapping method is rightfully called the LASSO for Least Absolute Shrinkage and Selection Operator [Tibshirani, 1996] and studies such as [Uniejewski et al., 2016] claim that it significantly outperforms regularization with $q = 2$ (Ridge Regression [Hoerl and Kennard, 1970]) and the Expert model. The Elastic Net [Zou and Hastie, 2005] uses a regularization term constructed using $q = 1$ and $q = 2$ and yields even more accurate predictions. LASSO models were extensively studied in EPF during several years [Janke and Steinke, 2019; Marcjasz, 2020; Narajewski and Ziel, 2020a; Özen and Yıldırım, 2021; Uniejewski and Weron, 2018; Ziel and Weron, 2018] and they constituted the state of the art statistical model until its upgrade in the LASSO-Estimated Auto Regressive model (LEAR) [Lago et al., 2021]. This model renders the estimated price $Y^{(d,h)}$ dependant not only on data from the corresponding hour h as in the Expert model, but on all hours of the day. The expression for $\widehat{Y^{(d,h)}}$ becomes:

$$\widehat{Y^{(d,h)}} = \sum_{i=1}^{24} \left[\beta_1^{(i)} \cdot \begin{pmatrix} Y^{(d-1,i)} \\ Y^{(d-2,i)} \\ Y^{(d-3,i)} \\ Y^{(d-7,i)} \end{pmatrix} + \beta_2 \cdot X^{(d,i)} \right] + \beta_3 \cdot D$$

with the exogenous features $X^{(d,i)} \in \mathbb{R}^2$.

2.2.1.3 Machine Learning models

Many ML models have been studied and this section provides a list of the most popular ones. Detailing the subtleties of all ML models is beyond the scope of this thesis. Instead, links with the EPF task are given.

[1] This term used in the literature, denotes the necessary expertise to select the variables, and has to be differentiated from our class of approaches that we also called *Expert*

Support Vector Machines (SVM) [Sansom et al., 2003; Zhao et al., 2008] map data to a higher-dimensions space using kernel functions, before computing decision boundaries using linear functions. Usually, the Radial Basis Function kernel is used [Niu et al., 2010; Shiri et al., 2015], but it's possible to use a predefined kernel that better separates the data. SVM are efficient models that work well with small datasets with numerous features. This is usually the case with EPF datasets where $X \in \mathbb{R}^{n_d \times 24 n_f}$. Historical datasets are sometimes hard to constitute for some variables (small n_d) and if considering lagged prices $Y^{(d-1,h)}, Y^{(d-2,h)} \dots Y^{(d-l-1,h)}$ as variables, n_f quickly grows. They have the disadvantage of outputting only one scalar value. Hence, 24 models per zone to forecast have to be trained.

Random Forests (RF) [Mei et al., 2014] consist of a collection of Decision Trees trained on different subsets of the input data. For each tree, the data is split using decision rules that best discriminate the labels. By summing the number of times that a feature is used to split the data across all trees of the forest, RF provide a feature importance evaluation that is valuable for practitioners [Díaz et al., 2019]. The drawback of RF for EPF is that each tree can only forecast a value that it has already seen. The averaging over numerous trees can interpolate new values, but unseen price spikes can't be extrapolated.

The first Neural Networks (NN) to be used for EPF contained a single linear layer, with sigmoid or RBF activation functions [Garcia-Ascanio and Maté, 2010; Pindoriya et al., 2008] that use Gaussian kernels [Amjady and Hemmati, 2009; Chen et al., 2012; Cruz et al., 2011; Garcia-Ascanio and Maté, 2010]. More complex NN were then used in [Keles et al., 2016] and [Lago et al., 2018a] that introduced a model that they qualified as Deep Neural Network (DNN). This model consists of two dense hidden layers, followed by a 24-neurons output layers whose output values are $\widehat{Y^{(d)}}$. The difficulty in training DNNs for EPF is the number of trainable weights n_w, that is significantly larger to the number of training samples n_d. For instance, [Lago et al., 2018a]'s DNN has $n_d \approx 1200$ and $n_w = 750 * 239 * 162 * 24 = 700M$ weights as illustrated in Figure 2.2. As a result, performances are heavily dependent on weight initialization and in a follow-up study, [Lago et al., 2021] observes large performance variations for different runs.

Convolutional Neural Networks (CNN) have also been used by [Cheng et al., 2020; Khan et al., 2020] to forecast prices. CNNs extract features from a n-dimensional input by moving filters along the dimensions. Each filter can be viewed as a local feature extractor. Typically for EPF, convolutions are performed over the time dimension that has a length of 24 and sometimes the variable dimension. The variable dimension is made of the employed features (generation forecasts, consumption forecasts, etc...) and is thus usually small, compared to the Imagenet dataset (222×222) that made CNNs very popular [Krizhevsky et al., 2012]. Also, results can vary based on how the features are ordered in the first place since it changes the value of kernels. [Lago et al., 2018a] decide to only perform 1-dimensional convolutions over the time dimension, bypassing the problem of feature ordering but ignoring interactions between variables during kernel feature extraction. A summary of those problems is displayed in Figure 2.3.

Recurrent Neural Networks (RNN) add memory to the network. Each neuron's output is

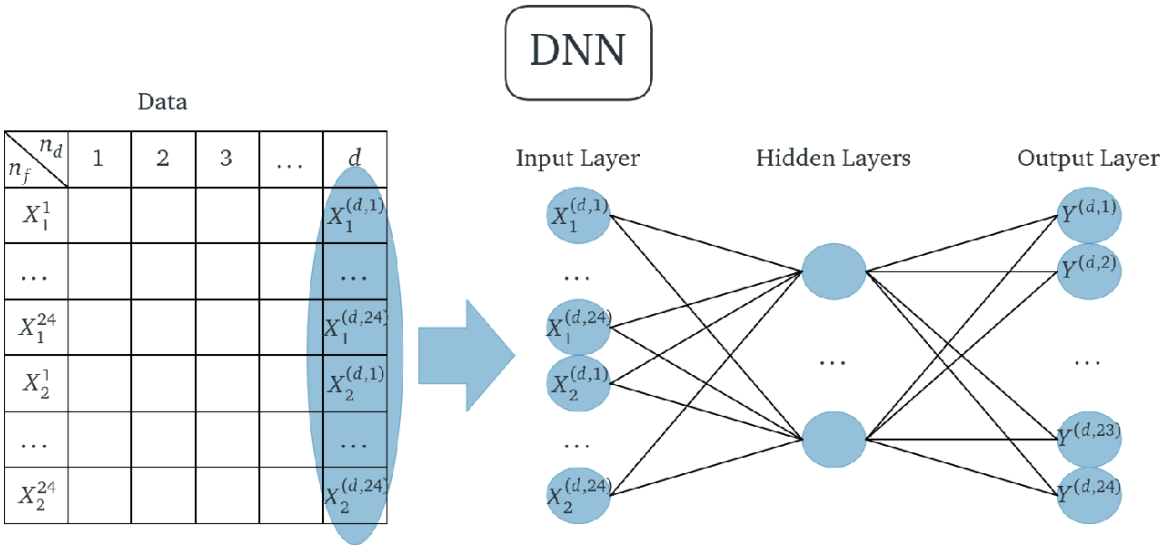


Figure 2.2 – Using a DNN for Day-Ahead price prediction. Dense layers with n_1 and n_2 neurons require $n_1 * n_2$ trainable weights.

now a direct input of the same neuron, for the next training iteration. The Long Short-Term Memory (LSTM) extend the principle of the RNNs [Anbazhagan and Kumarappan, 2012; Chen and Zhang, 2021; Sharma and Srinivasan, 2013] and combine time-dependencies from different horizons. LSTM are capable of modeling long-term and short-term time dependencies at the same time. Lastly, a Neural basis expansion (NBEATs) model introduced in [Oreshkin et al., 2019] to the EPF task [Olivares et al., 2023]. This model consists in decomposing the input signal using predefined projections V. Similar to a Convolutional Neural Network with residual connection, the input data goes through a series of independent blocks. Each block is made of several dense layers, followed by the projection on 2 separate predefined vectors V^{back} and V^{for}. V^{back} is subtracted from the block's input to constitute the next block's input. The idea is that this quantity reaches 0 in the last block. V^{for} is summed with all other block's forecasts to constitute the model's forecast. Similarly to LSTM, each block can then capture a specific seasonality. This sounds ideal for EPF where there are annual, weekly and daily seasonality components in the price. However, we must keep in mind that market participants can't buy electricity for a period (d, h) and sell it later at (d', h'). Hence, there is no reason why past prices should be the cause of a next day's prices. Even though, some studies that use LSTM [Andalib and Atry, 2009; Sharma and Srinivasan, 2013] claim that those models are superior to traditional Neural Networks.

2.2.1.4 Milestone Papers

A significant milestone for EPF is the work of [Lago et al., 2018a]. It is a good introduction to EPF in general as it gives an overview of the appropriate procedure for forecasting prices (data processing, data splitting, hyper-parameters selection, model evaluation) and an overview of the most popular models. The authors compare 24 Statistical and Machine Learning models that were used in previous contributions in the EPF field. Among those

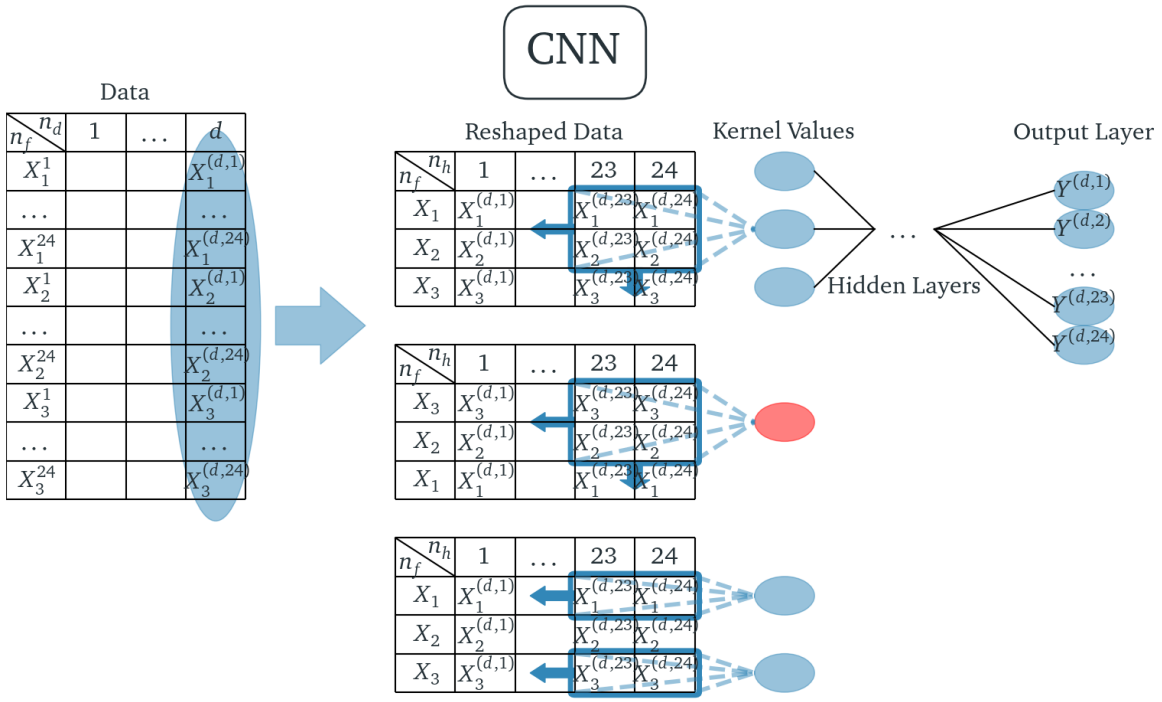


Figure 2.3 – Using a CNN for Day-Ahead price prediction. (TOP) and (MIDDLE) Modifying the order of the variables X in the reshaping process can cause a difference in the value of the computed kernel. (BOTTOM) [Lago et al., 2018a]'s convolutions are only performed on the time dimension.

models were AR models and their variants ARX, TARX, ARIMA, GARCH, as well as standard ML models (RF, SVR) and ANN models such as MLP, DNN, LSTM, CNN. The authors concluded that the DNN model significantly outperforms other models on the Belgian dataset. In a follow-up study [Lago et al., 2021] established a rigorous evaluation procedure for model comparison, and set the LEAR and DNN models as the two most adequate models for forecasting prices.

2.2.2 Expert models

In order to enforce the numerous physical constraints inherent to electrical network, as well as for limiting unfair behaviors from oligopoly companies, electricity markets are under heavy regulations. Many EPF models use expert-knowledge to replicate those rules and simulate prices.

2.2.2.1 Fundamental Analysis

In Fundamental analysis, functional relationships between electricity price and its main drivers (load, weather, commodity prices), which is called the bid stack, are first formulated. Then, the drivers are forecasted independently using other processes (usually statistical models). The bid stack can be constructed by sorting all available power plants by increasing

marginal cost. Plants are activated until the estimated demand is reached. Then, the price forecast is the price of the last (and therefore most expansive) plant to be turned on. A graphical explanation of the bid stack is available in Figure 2.4. Because the pricing of such plants is easier to replicate, this approach works better on hydro-dominated markets such as Scandinavia [Johnsen, 2001; Vehviläinen and Pyykkönen, 2005], where snowfall is the main variable to consider. On other markets where thermal plants are available, the prices of other commodities have to be forecasted. Other strategies estimate the relationship between the price and its drivers using a stochastic process : the Ornstein–Uhlenbeck process [Boogert and Dupont, 2008]. In combination, a transformation can be applied to the prices such as the inverse Box-Cox [Barlow, 2002] or the hockey-stick curve that better matches the empirically observed curves [Kanamura and Ōhashi, 2007]. The weaknesses of those approaches lie in the formulation of the relationship between the drivers and the prices that is hard to make exhaustive due to the number of available plants in a country. Additionally, data about certain plants may be missing, rendering the marginal cost or available capacity impossible to compute. Lastly, the forecast of the price drivers induces imprecision that are twofold. First, forecasting commodity prices such as gas price is in itself a difficult task because it also depends on numerous factors. Then, the drivers are always forecasted by minimizing a prediction error, which means obtaining the forecasts as close as possible to the true values. However, the real error that has to be minimized is the EPF error. Depending on the formulated relationship between prices and drivers, a low error on the drivers could lead to important prices differences.

2.2.2.2 Strategic Production-Cost

To improve over Fundamental Analysis, Strategic Production-Cost models consider market players as agents that aim to maximizing their gains. To do this, they construct each agent's expected behaviors taking into account their marginal costs [Batlle, 2002; Batlle and Barquín, 2005]. For this, they rely on market regulations that stipulate that suppliers can't sell energy higher than their marginal costs to prevent important actors of oligopolistic markets to manipulate the electricity price for their own profit. More sophisticated versions of this approach also simulate strategic bidding [Wood et al., 2013] that is, each agent considers other agents' strategies. The market equilibrium is expressed as a set of algebraic equations. It relies on the balancing requirements of markets that need a constant balance between generation and consumption. The market equilibrium is obtained by solving the system [Ruibal and Mazumdar, 2008] from which a price is deducted. The unavailability of generation units [García Alcalde et al., 2002; Vives, 1999] can be added to the equation system. The player's reaction to rivals can also be taken into account. Thus, the equilibrium is computed using differential equations [Baldick et al., 2004]. Those methods are still difficult to employ in applications where a good precision is required. They usually introduce a bias in the results with prices higher than observed. They fail to produce quantitative results and hardly adapt to deregulated markets.

2.2.3 Synthesis

The limitations of the EPF literature lie in the fracture between Expert models and Data-Driven approaches. Expert models follow closely market regulations, estimate bidders

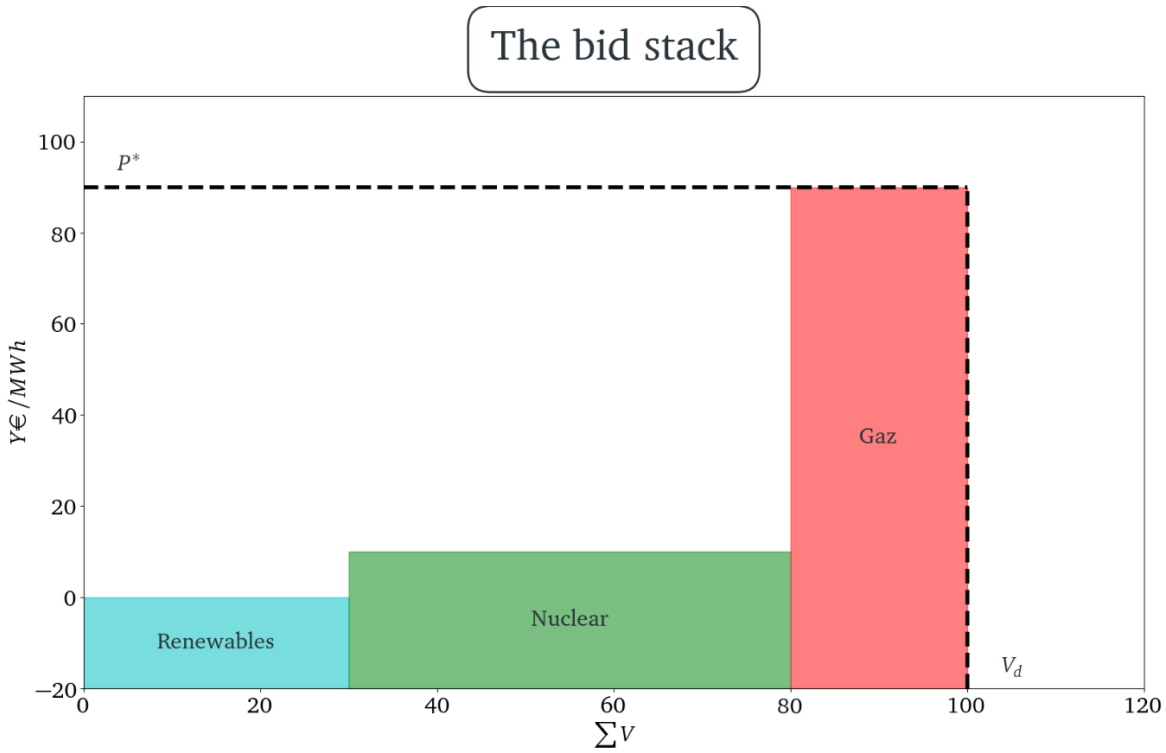


Figure 2.4 – All available plants are sorted by increasing marginal cost, until the estimated demand volume V_d is reached. Then, the price forecast is the marginal price of the most expansive plant $P^\star$.

strategies and simulate price fixing algorithms. As such, the forecasted prices are consistent with the market characteristics and their relationships with the fundamental variables are understandable. Because Expert models represent exact relationships between drivers and prices, uncertainties in the drivers drastically reverberate in the forecasts accuracy, rendering Expert models noncompetitive. Oppositely, Data-Driven methods can account for drivers unreliability by constructing its own relationships between drivers and price. It turned out that the forecasting accuracy of Data-Driven methods largely outperforms that of Expert models. However, most Data-Driven models are black-box models whose training procedures completely occult market rules and price fixation mechanisms. As a result, the constructed relationships between drivers and prices are opaques and price forecasts can lack consistency with the fundamental variables. For numerous applications of price forecasts such as day-ahead trading, consistency of the prediction with the market and understandability of the forecasts is paramount, but a model that forecasts wrong values is also undesirable.

2.3 Limitations

We use the term *correctness* to define how close a forecast is to the real price. This is easily measured by the difference between the price and the forecast. The term *consistency* describes

how coherent a forecast is with respect to the fundamental variables or market regulations. It is qualitatively measurable along several criteria that we will present throughout this section. These two terms allow us to summarize the respective limitations of Expert models and Data-Driven approaches by the trade-off between correctness and consistency. Expert models models are consistent but lack correctness. Data-Driven methods excel in correctness but miss consistency. In this section, we will focus on the attempts to improve consistency of various Data-Driven approaches and detail why they are incomplete.

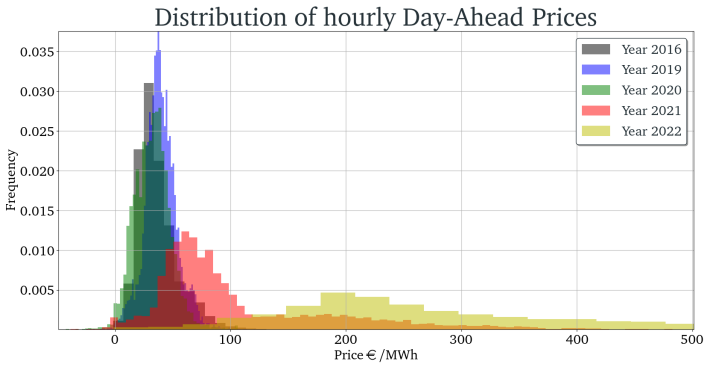
2.3.1 Understanding the predictions

Once trained, Data-Driven approaches are characterized by their opacity. The test set error provides an overview of the forecast quality, yet it is difficult to describe the interactions between the forecast and the input variables. In the EPF literature, attempts to explain forecasts are babbling. The only few studies that are covered here provide global explanations to the forecasts, that give a general idea to the end-user on how a model works, which input features are important, which market specificities are captured or how the model reacted to an unprecedented event.

For instance, the Covid-19 pandemic and its successive impacts on electricity markets have been poorly addressed. [Narajewski and Ziel, 2020b] and [Suvarna et al., 2022] focus on load forecasting, while [Arya and Chandrakala, 2021] is not comparing its results to the state-of-the art models (LEAR and DNN), and prediction errors are not properly analysed against the market characteristics using explainability tools. Consequently, the relationship between prices and current situation (lock-downs, regulations, etc...) is not put forward. The work of [Lago et al., 2021] introduces benchmarks whose test set is the year 2016. The Covid crisis and its repercussions are thus out of the scope of the benchmark, whereas experts from businesses agree that those events where game-changers. Price histogram and statistics from this period can be observed in Figure 2.5 where it clearly appears that a new regime has been in place since 2021. It is also important to note that Data-Driven models require extensive historical datasets to integrate new regimes. Furthermore, models such as the Random Forests are not able to extrapolate values not present during training. As such, drastic changes in market prices will not be properly forecasted.

2.3.1.1 Model-Specific methods

Models natively providing some kind of explanations have been used to measure the impact of input features on the forecasted price. The Auto-Regressive models' weights are analyzed for EPF in [Ziel, 2016]. They consider the weights $\beta_1, \beta_2, \beta_3, \beta_4$ of the *Expert* model trained for forecasting the German prices. They find that the last price of the previous day $Y^{(d-1,24)}$ is the most important input variable for all hours of the next day, $\beta_{2,1} > > \beta_{i,j}\ \forall i, j$. They also note that the price of the previous day $Y^{(d-1,h)}$ for the same hour to predict is important: $\beta_{1,1}$ has a high value. Moreover, they show that weekly lagged values $Y^{(d-7,h)}$ are not considered important by the model because $\beta_{1,3}$ is low. Results of [Maciejowska et al., 2022] confirm this finding using a more sophisticated LEAR model. Their results are displayed in Figure 2.6.



Year	Average (€MWh	STD (€/MWh)
2016	36.75	24.44
2019	39.44	14.02
2020	32.2	16.11
2021	109.16	84.3
2022	275.84	145.82

Figure 2.5 – (LEFT) Sample Hourly Day-Ahead prices from the French market for different years. The year 2016 that is used for testing in the literature [Lago et al., 2021] is plotted in black and does not represent the current situation, especially during time of crisis (year 2021 in red). (RIGHT) Average price and Standard Deviation for several years in the French Market.

The drawback of using LEAR weights and RF feature importance is that the obtained values are *model specific*, making them either irrelevant due to model performance or impossible to compare with other methods. The two considered models are, according to [Lago et al., 2018a], among the worst performing ones. [Van der Heijden et al., 2021] state that the ARX results can be incomplete because they can only capture linear relationships between data. Oppositely, RF can only grasp non-linearity between variables. Feature selection from these models could be misleading: perhaps the feature is not considered important by the model because it failed to capture its dynamics.

2.3.1.2 Feature selection

The abundance of price drivers available to forecasters and the incapacity for several models to be efficiently trained with an exceeding number of variables ($n_f > n_d$) led the scientific community to focus on feature selection from the feature set χ. Selection criteria are based on a feature importance value that helps in understanding the model.

For instance, [Lago et al., 2018b] considers features as hyper-parameters that are selected using a Tree Parzen Estimator (TPE) algorithm [Bergstra et al., 2011]. The TPE optimizes a black-box function M that maps a hyper-parameter set θ_i to an evaluation loss l_i. New hyper-parameters configurations θ_j are explored by taking the local minima of $M : \theta_j = \arg\min_\theta M(\theta)$. In addition to this hyper-parameter sampling process, authors use the functional ANOVA methodology [Hutter et al., 2014] that decomposes the model performance variations into the sum of effects of all possible feature subsets $x \subseteq \chi$. The importance of a subset x can be expressed as: $I_x = \frac{V_x}{V}$ with V_x the variance induced by a subset x and V the total variance. [Lago et al., 2018b] computes the importance I_x for all individual features $x \in \chi$ and pairwise interactions $x \in \chi^2$. Then, features whose individual or pairwise importance exceeds a threshold $\epsilon = 0.5\%$ are selected. They apply their methodology for predicting the Belgian prices based on the generation and consumption forecasts

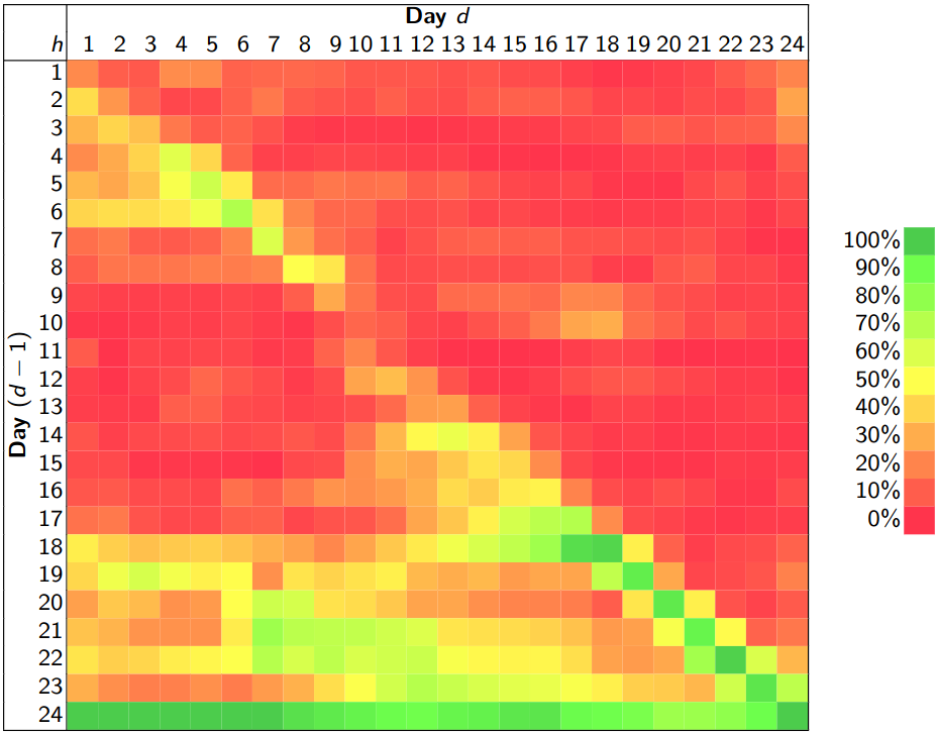


Figure 2.6 – The LEAR weights of $Y^{(d-1)}$ (y-coordinates) for predicting $Y^{(d)}$ (x-coordinates) on the German market (Figure from [Maciejowska et al., 2022]).

and price history of France and Belgium. In a first experiment, they find that the Belgian generation forecast degrades performances with a high importance value: $I_x = 75\%$. In a second experiment, they discard the Belgian generation forecasts and repeat the process. They find that French data accounts for more than 50% of the variance while significantly improving the metrics.

Also, [Van der Heijden et al., 2021] uses an ARX and a Random Forest to select candidate features to include in the training set of LEAR and DNN models. Particularly, they focus on selecting the best features from European countries to fit an EPF model on the Dutch market, using a two-stage method. In a first step, they consider the consumption, consumption forecast and past day-ahead prices from all European zones to fit an ARX and a RF model for predicting the Dutch prices. Based on the joint analysis of feature weights in the ARX and feature importance in the RF, they select all countries whose weights or importance in forecasting Dutch prices are above 5%. They also select neighbors of the Netherlands. In a second step, they use their reduced dataset that contains 7 foreign countries (France, Germany, Belgium, Great-Britain, Norway, Denmark and Italy) to fit more sophisticated LEAR and DNN models [Lago et al., 2021]. They include each foreign country's features one by one and measure the performance increase on the validation dataset. Countries are discarded if they deteriorate performance. They end up keeping only 3 countries: France, Great-Britain and Denmark. [Van der Heijden et al., 2021] show that their selected features lead to significant performance increase on a separated test dataset. The first selection step

shows the limits of *model specific* methods. Nothing states that a feature selected based on an ARX or RF model would also be significant for a LEAR and DNN. This is especially true for DNN models whose performances are sensible to the number of input features.

Apart from it, both methods use *model agnostic* methods. Aiming at selecting features that increase correctness, the proposed methods provide only *global explanations* of the forecasts: interactions between features and predictions are averaged over the considered period. Given a single prediction, it's not possible to tell which feature acted on the price and how much.

2.3.1.3 The need for model-agnostic individual explanations

The two major flaws of the four considered studies is that 1) they use *model specific* feature importance measures 2) The *Representativeness* [2] of their methods covers the entire dataset: they produced *global explanations*. The first is problematic because the used methods are tied to underperforming models that make the obtained results limited or difficult to compare. The second makes it impossible to explain a single forecast. In EPF, the price is forecasted by many variables that are estimations such as consumption or renewables, and knowing which characteristics were the most important for a prediction helps practitioners using the model. It's possible for instance to include last-minute information (e.g. a plant unavailability, a strike, etc.) directly to the forecast. Also, some usage (e.g. trading) of a price forecast requires significant trust of the model's results, that can be provided by individual forecast explanations. Model-agnostic individual explanation tools such as SHAP [Lundberg et al., 2020] would be suitable for EPF as it allows comparing contribution between different models. Individual explanations are a crucial tool and their absence limits the model usability by practitioners. Feature importance, model-agnostic explanations and individual forecast explanations are three criteria that increase the consistency of a model.

2.3.2 Considering the European Network

Markets under study span across almost all European countries. The most studied countries are Germany, Spain and the United Kingdom. Many studies also consider the Scandinavian system price [3]. Surprisingly, the French market is understudied whereas it is the second-most important market in Europe. Moreover, the French consumption or generation forecasts are crucial variables to explain the Spanish, German, or Belgian prices. Table 2.1 summarizes the most important publications for each European market.

2.3.2.1 Considering several markets

Most of the literature focuses on forecasting prices of a single market. This questions the flexibility of the trained model as there are no guarantees that performances will be good on a market with other characteristics. Many factors have to be reconsidered while crossing borders such as the energy mix of a country, its demand sensibility to temperature changes,

[2] *Representativeness* is defined in [Molnar, 2020] as the span of an explanation: global (on the entire dataset) or individual (on a single forecast).

[3] The system price is computed by isolating the Nordic regions (Norway, Sweden, Finland, Denmark and the Baltic) and considering unlimited exchange capacity between those zones.

Market	Publications
Germany	[Marcjasz et al., 2018] [Ziel and Steinert, 2016] [Lago et al., 2021]
Spain	[Conejo et al., 2005] [Díaz et al., 2019]
Belgium	[Lago et al., 2018b] [Lago et al., 2018a] [Lago et al., 2021]
France	[Lago et al., 2018b] [Lago et al., 2021]
Scandinavia	[Lago et al., 2021] [Olivares et al., 2023] [Kristiansen, 2012]
United Kingdom	[Gonzalez et al., 2011] [Karakatsani and Bunn, 2008]
The Netherlands	[Van der Heijden et al., 2021]

Table 2.1 – Most frequently considered markets in Europe, and their most relevant contributions.

the power plants availability, the bidding strategies of suppliers, local regulations, etc. Also, studies that assess their models on several markets consider them separately [Lago et al., 2021] i.e. they fit independent regressors for each countries. They evaluate 4 different LEARs and DNNs models on different European areas and find that all DNNs significantly outperform all LEARs on German and Belgian datasets, while only 2 LEARs in Scandinavia and 1 LEAR in France. If tested against more powerful models (i.e. SVR), DNNs might be outclassed in specific markets.

The attempt of [Lago et al., 2018b] to forecast the prices of 2 neighboring countries (France and Belgium) using a single model concluded that the multi-forecasting model performed significantly better than the individual models on 7 out of 24 hours. This study was only limited to 2 countries, however, on the European market, the EUPHEMIA algorithm computes prices of the 39 zones by aggregating orders from all countries. Figure 2.7 displays the different zones of the European market. Technically, all prices thus depend on the Order Book from all other countries. Modeling pan-European electricity prices is a goal that has never been achieved in the EPF literature.

2.3.2.2 Energy Exchanges

Order Books from neighboring zones z and z' interact with each other using cross-zonal flows $F_{z,z'}$. Connection lines between two zones have fixed Available Transfer Capacities (ATC) $\Gamma_{z,z'}$ that can be filled to balance energy across both countries : $\Gamma_{z,z'}$ limits energy exchanges between countries. This means that the importance of neighbors is dependant on cross-border flows as well as on its fundamental forecasts (renewable, consumption). The latter is considered in [Van der Heijden et al., 2021] that use the features of the Netherland's neighbors as input variables of their Dutch prices forecasting model. To choose which countries to include, they analyze the ARX weights, RF feature importance and ATC. The flaw in their methodology is that ATCs are only considered in the first selection step and are not included in the final model training, whereas they are necessary to quantify the energy flows between two zones. To the best of our knowledge, the consideration of cross-zonal flows in a ML EPF model has never been achieved in the literature.

Lastly, in their study, [Van der Heijden et al., 2021] find that the Italian data (IT-NORD) has significantly high ARX weights and RF feature importance, even though Italy and the Netherlands share no border. To exchange energy between those 2 countries, one has to cross

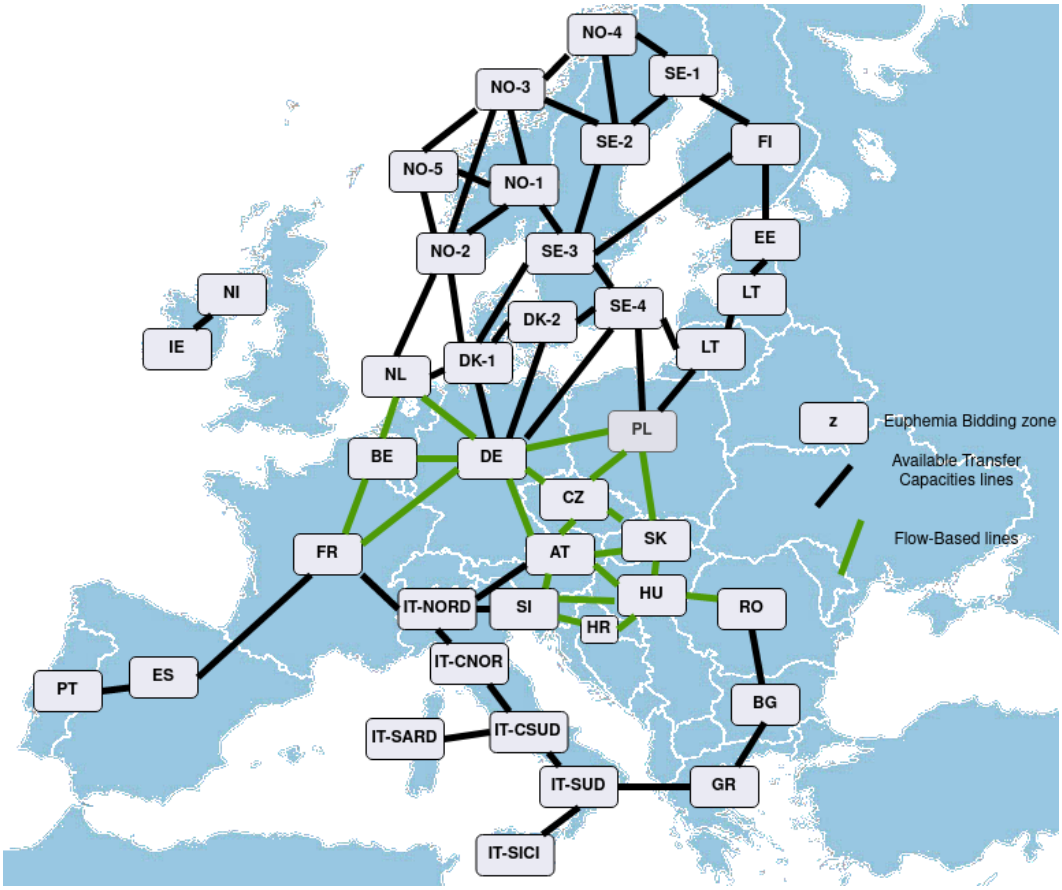


Figure 2.7 – The different bidding zones whose price is computed by EUPHEMIA. Countries can be split into several zones that have unique prices (Italy, Scandinavia). Country codes are displayed in place of the zone names. A conversion table can be found in the Appendix B.1. Connection lines between zones are represented by black segments if constrained by ATC, and by green ones if constrained by Flow-Based.

several other zones and the problem thus complexifies. The shortest path is Belgium-France, Germany-Switzerland or Germany-Austria, but perhaps the cheapest option is to send energy through more countries : the considered connection lines could already be overflown, or the cost for sending energy through a specific line too expensive. The problem to be solved is translated as finding the path with the lowest cost $t^\star$ among all possible paths $\mathbb{T}$ from the Netherlands to Northern Italy, while ensuring that all flows from crossed countries are not exceeded $F_{z,z',h} \leqslant \Gamma_{z,z',h} \; \forall \; (z,z') \in t$ for all hours h. The topological structure of the European network has to be considered. Yet, the models that they later trained to forecast the Dutch prices (LEAR, DNN) are fed with tabular data that do not account for topology. Only a Graph structure can represent the European Network. Despite this, Graph Neural Networks (GNN) have never been used in the EPF field.

Three aspects should be considered to enrich the EPF literature. First, Multi-zone forecasting models can improve both correctness and consistency. Then, cross-zonal flows

should be modeled using ATCs. Lastly, the Graph data structure is the only way to model the topology of the network.

2.3.3 Considering the Price Fixation Algorithm

Expert-Based Approaches aimed at replicating the market rules and price fixation mechanism, then simulating scenarios under those rules to obtain a price. Due to their poor performance, they have been replaced by Data-Driven approaches that completely occult the price fixation mechanism.

2.3.3.1 Aggregated Curve

The only approach that integrates elements from the PFA in a price forecasting model is based on the estimation of AGGREGATED CURVES (AC) [Ziel and Steinert, 2016]. ACs graphically represent the Order Books for a specific hour and zone (we use the notation $AC^{(d,h)}$), with all COMPLEX and BLOCK orders transformed into STEP or LINEAR orders. AC can only be drawn after the price computation to decompose complex and block orders and are published shortly after price computation by the exchange. Supply bids are sorted by increasing selling cost and the volumes are accumulated. Demand bids are sorted by decreasing buying price. We formulate the AC as a function that computes the total bid volume as a function of the selling price P:

$$AC^{(d,h)} = \left(S^{(d,h)}, D^{(d,h)}\right)$$
$$S^{(d,h)}(P) = \sum_{p=p_{min}}^{P} v_S^{(d,h)}(p)$$
$$D^{(d,h)}(P) = \sum_{p=P}^{p_{max}} v_D^{(d,h)}(p)$$

with $S^{(d,h)}$ and $D^{(d,h)}$ respectively the supply and demand curves, p_{min} and p_{max} the minimal and maximal prices for a bid fixed by the exchange and $v_S^{(d,h)}(p)$ and $v_D^{(d,h)}(p)$ the bidded volumes for a given price p for supply and demand. The optimal price $P^\star$ can then be found exactly at the intersection of the supply and demand curves:

$$P^\star = \{P | S^{(d,h)}(P) = D^{(d,h)}(P)\}$$

We formalize this notation by introducing the *Intersection* function I that maps an element from the AC domain $\mathbb{A}$ to the day-ahead price:

$$I : \mathbb{A} \mapsto \mathbb{R}$$
$$AC^{(d,h)} \to Y^{(d,h)}$$

Analysing the AC reveals the real sensibility of the price to shifts in supply or demand. Similarities between the the bid stack and the supply side can easily be drawn in Figure 2.8. On the one hand, volume shifts can be absorbed by generation excess, for instance in Figure 2.9 where lowering the demand at highest price by 30MWh does not modify the price since the same set of plants has to be turned on. On the other hand, turning on a more expensive plant can drastically increase the price with only a small extra produced volume,

as it is the case in Figure 2.10 where a small decrease of 5MWh in the generation at lowest price significantly increases the price. Note that on all three figures, we displayed the price as a function of the bid volume (S^{-1} and D^{-1}) to follow the conventional representation of AC.

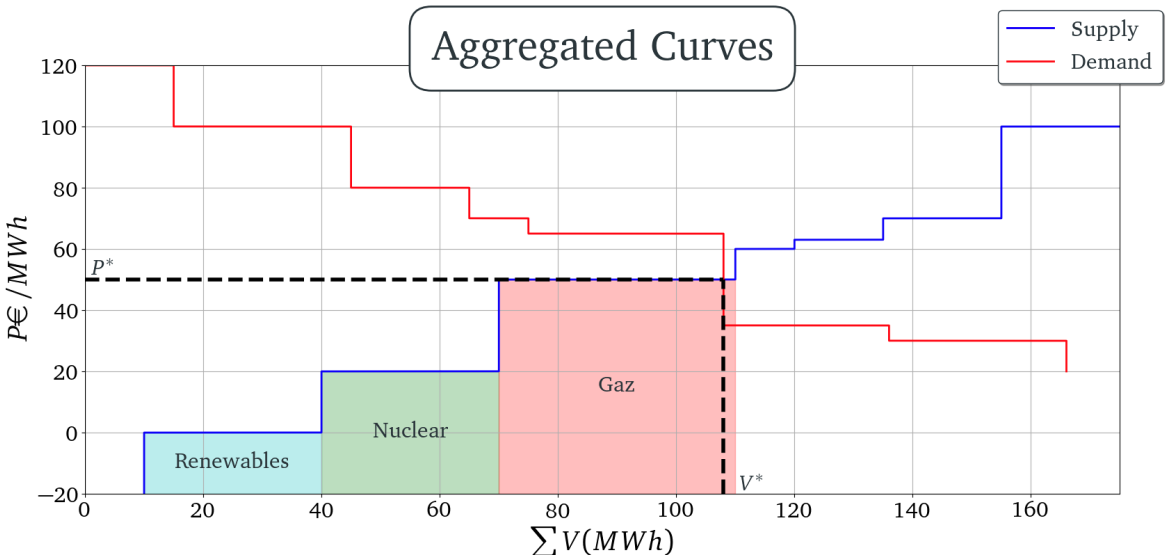


Figure 2.8 – Arbitrary Aggregated Curves with the supply in blue and demand in red. Similarities with the Bid Stack can be drawn on the supply side where each step corresponds to a plant with a fixed marginal cost and generation volume.

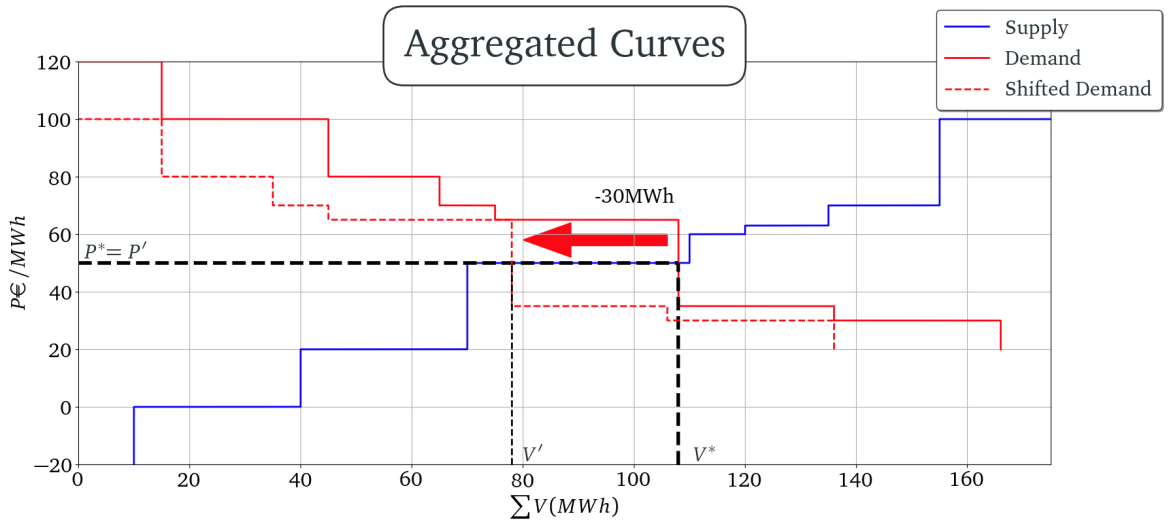


Figure 2.9 – Arbitrary Aggregated Curves with the supply in blue and demand in red. A medium shift of 30MWh in demand (red-dashed line) does no changes the optimal price $P^\star$

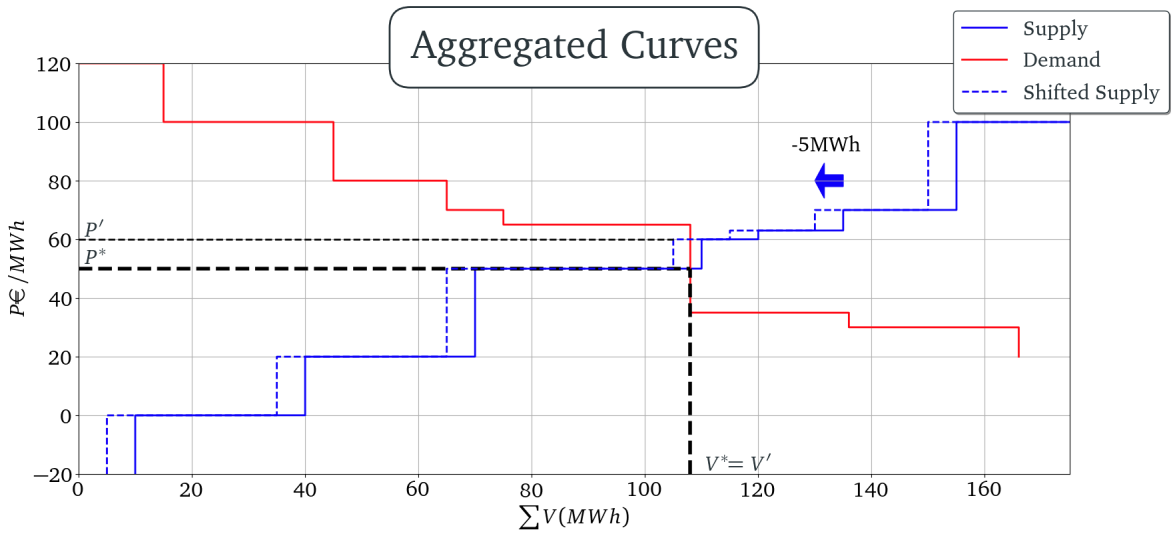


Figure 2.10 – Arbitrary Aggregated Curves with the supply in blue and demand in red. A small shift of 5MWh in generation (blue-dashed line) drastically increases the optimal price $P^{\star}$.

2.3.3.2 The X-Model

[Ziel and Steinert, 2016] propose a method to estimate AC for the hour to predict and to compute the price as the intersection of the two curves, forming the so-called X-Model. This approach is unique because it aims at integrating elements from the PFA directly into the prediction model. Considering EUPHEMIA while forecasting prices is a major issue in EPF and the X-model is the only approach that addresses it.

Estimating $\widehat{AC}^{(d,h)}$ is a difficult task as the number of points in each curve is high (several hundreds) and varies every hour. Thus, prior to training, the authors first summarize the curves by extracting the sum of bid volumes for $n_p = 15$ predefined prices $\mathbb{P}_S$ and $\mathbb{P}_D$, on the supply and demand side. The AC are first averaged across the dataset:

$$\overline{AC} = \left(\bar{S}, \bar{D}\right)$$
$$\overline{S(P)} = \frac{1}{n_d * n_h} S^{(d,h)}(P)$$
$$\overline{D(P)} = \frac{1}{n_d * n_h} D^{(d,h)}(P)$$

Then, $\mathbb{P}$ are computed by projecting equally spaced volumes on the curves:

$$\mathbb{P}_{Si} = \bar{S}^{-1}(i * V_{step})$$
$$\mathbb{P}_{Di} = \bar{D}^{-1}(i * V_{step})$$

with $i = 1 : n_p$ and V_{step} a predefined volume $V_{step} = \frac{V_{max}}{n_p}$ and $V_{max} = \max_P(\bar{S}(P), \bar{D}(P))$. $\mathbb{P}$ are

then projected back on each available curves and volume values are extracted:

$$S_{vi}^{(d,h)} = S^{(d,h)}\left(\mathbb{P}_{Si}\right)$$
$$D_{vi}^{(d,h)} = D^{(d,h)}\left(\mathbb{P}_{Di}\right)$$

A graphical explanation of this process is displayed in Figure 2.11. The 30 extracted values constitute a reasonably-sized targets that is forecasted using an Auto-Regressive process m and exogenous features X.

$$\left(\widehat{S^{(d,h)}}_v, \widehat{D^{(d,h)}}_v\right) = m(X)$$

with m trained using historical S_v and D_v. Then, they employ a reconstruction algorithm that simulates orders between forecasted points. The idea is to draw more points from $AC^{(d,h)}$ to get a finer price estimate. The general formula for the reconstructed curves is given by:

$$\widehat{AC^{(d,h)}} = \left(\widehat{S^{(d,h)}}, \widehat{D^{(d,h)}}\right)$$
$$\widehat{S^{(d,h)}}\left(P\right) = \frac{R_s(P)\bar{S}(P)}{\sum_{p\in[\mathbb{P}_{Si},\mathbb{P}_{Si+1}[} R_S(p)\bar{S}(p)}\widehat{S^{(d,h)}}_{vi} \qquad \forall\ P \in [\mathbb{P}_{Si}, \mathbb{P}_{Si+1}[$$
$$\widehat{D^{(d,h)}}\left(P\right) = \frac{R_D(P)\bar{D}(P)}{\sum_{p\in[\mathbb{P}_{Di+1},\mathbb{P}_{Di}[} R_D(p)\bar{D}(p)}\widehat{D^{(d,h)}}_{vi} \qquad \forall\ P \in [\mathbb{P}_{Di+1}, \mathbb{P}_{Di}[$$

with $R_S(P)$ being a Boolean variable that indicates if there is a supply order at price P between $\mathbb{P}_{Si}$ and $\mathbb{P}_{Si+1}$. Authors use the following formula:

$$R_S(P) = \begin{cases} 1 \text{ if } |\{v_S^{(d,h)}(P) > 0 | d = 1 : n_d, h = 1 : n_h\}| \geqslant 2 * n_d * n_h \\ 0 \text{ otherwise} \end{cases}$$

that is, if price P is used for bidding at least twice a day in the historical dataset. Similar reasoning is applied to $R_D(P)$. We call this step the *Reconstruction* function ρ that maps extracted points $\left(\widehat{S}_v^{(d,h)}, \widehat{D}_v^{(d,h)}\right)$ to an AC:

$$\rho : \mathbb{R}^{2\times n_p} \mapsto \mathbb{A}$$
$$\left(\widehat{S}_v^{(d,h)}, \widehat{D}_v^{(d,h)}\right) \to \widehat{AC}^{(d,h)}$$

The method proposed by [Ziel and Steinert, 2016] is summarized below:

TRAINING :

1. Compute $\overline{AC} = \left(\bar{S}, \bar{D}\right)$
2. Compute $\mathbb{P}_S$ and $\mathbb{P}_D$ by projecting V_i on $\bar{S}$ and $\bar{D}$
3. Compute $S_{vi}^{(d,h)}$ and $D_{vi}^{(d,h)}$ on all (d,h) of the training dataset by projecting $\mathbb{P}_S$ on $S^{(d,h)}$ and $\mathbb{P}_D$ on $D^{(d,h)}$.
4. Fit m to estimate $\widehat{S^{(d,h)}}_{vi}$ and $\widehat{D^{(d,h)}}_{vi}$ based on X

PREDICTION :

1. Forecast $\left(\widehat{S^{(d,h)}}_{vi}, \widehat{D^{(d,h)}}_{vi}\right) \leftarrow m(X)$

2. Reconstruct $\widehat{AC^{(d,h)}} \leftarrow \rho\left(\widehat{S^{(d,h)}}_{vi}, \widehat{D^{(d,h)}}_{vi}\right)$

3. Intersect $\widehat{Y^{(d,h)}} \leftarrow I(\widehat{AC}^{(d,h)})$

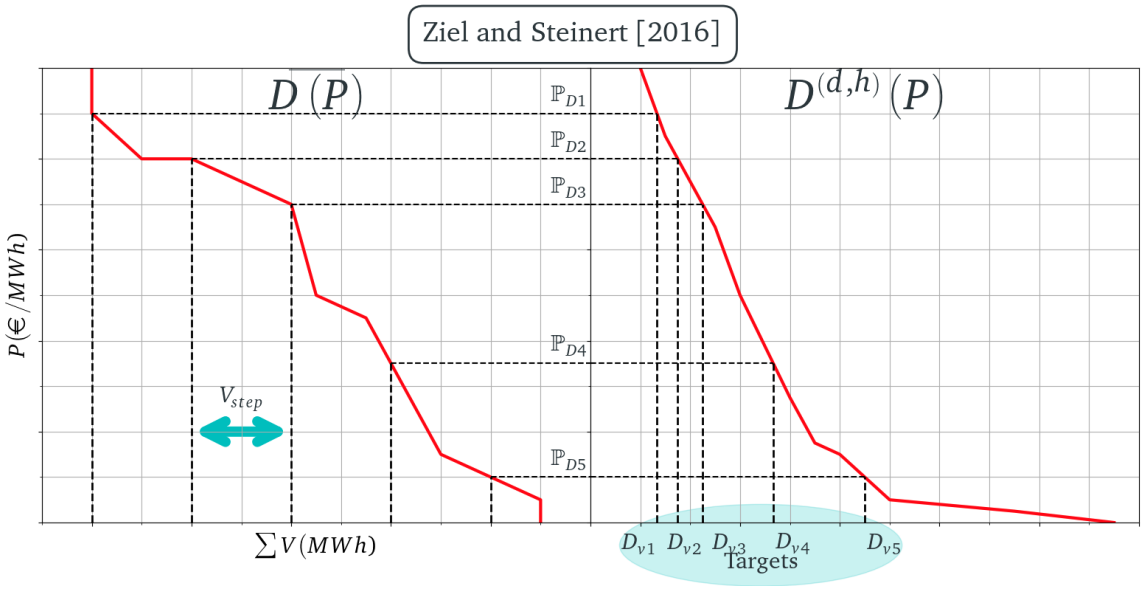


Figure 2.11 – [Ziel and Steinert, 2016]'s method for summarizing AC. (LEFT) the computation of $\mathbb{P}_D$ by projecting equally spaced volumes on the averaged demand curve $\bar{D}$. (RIGHT) Extraction of 5 points from an arbitrary demand curve $D^{(d,h)}$ by projecting the prices back to the volumes. The obtained D_{vi} are the targets to estimate.

[Kulakov, 2019] later extends this reasoning by combining the supply and demand curve. They extract the inelastic demand $D^{(d,h)}(p_{max})$ and form a combined curve $SD^{(d,h)}(P)$ by adding supply and demand volumes.

$$SD^{(d,h)}(P) = \sum_{p=p_{min}}^{P} v_s^{(d,h)}(P) + v_d^{(d,h)}(P)$$

The day-ahead price can then be found at the intersection of the curve $SD^{(d,h)}$ and the line $D^{(d,h)}(p_{max})$:

$$Y^{(d,h)} = \{P | SD^{(d,h)}(P) = D^{(d,h)}(p_{max})\}$$

A graphical summary of their approach is displayed in Figure 2.12. As a result, there is only one curve to estimate that they represent using 19 extracted points $SD_{vi}^{(d,h)}$ and the inelastic demand $D^{(d,h)}(p_{max})$. Estimation of $\widehat{AC^{(d,h)}}$ is performed using the AR model and reconstructed with [Ziel and Steinert, 2016]'s method. They report an error drop of 1.1€/MWh compared to the original X-model, with an execution time 3 times lower.

[Schnürch and Wagner, 2019] apply [Kulakov, 2019]'s transformation to the AC. The difference is that rather than estimating $\widehat{AC^{(d,h)}}$, they consider the previous day's extracted

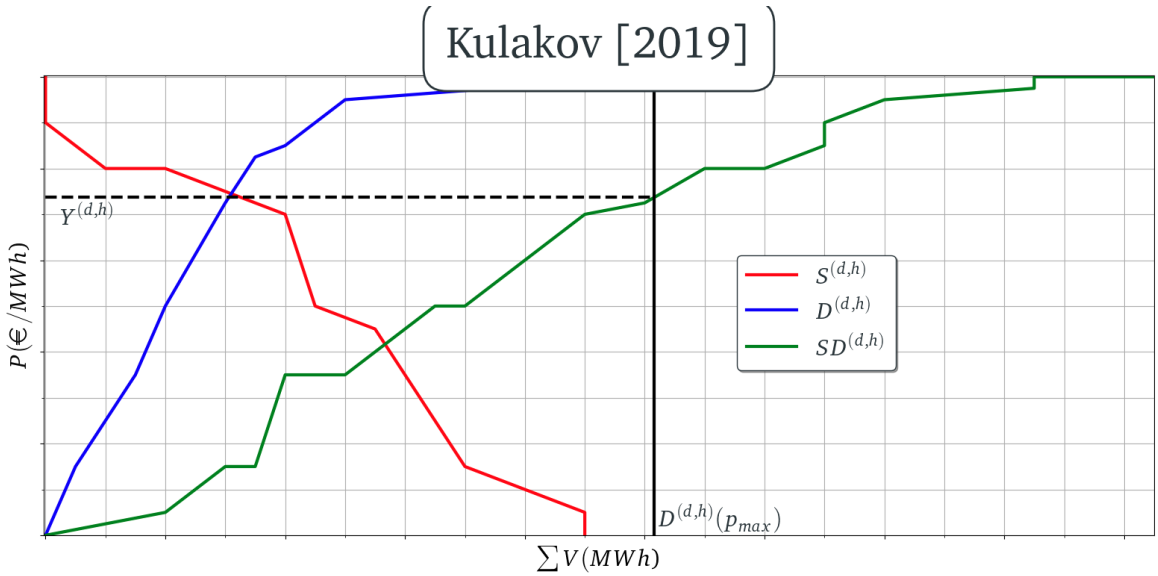


Figure 2.12 – [Kulakov, 2019]'s extension of the X-model. The supply and demand curves are added and the price can be found by intersecting this curve with the inelastic demand.

points $SD_{vi}^{(d-1,h)}$ as input features and directly forecast prices. The prices are forecasted using an AR, Random Forest or DNN model. Thus, they leave the curve reconstruction and intersection part to a black-box ML model. They somehow also choose to map input data not to the expected price $Y^{(d,h)}$ but to the closest extracted price $\mathbb{P}^{(d,h)}$. The error $|Y^{(d,h)} - \mathbb{P}^{(d,h)}|$ is 2€07/MWh on their dataset meaning that even a perfectly forecasted $\widehat{\mathbb{P}^{(d,h)}}$ model will have a significant error. They also perform feature selection using Random Forest feature importance to fit the DNN using all $SD_{vi}^{(d-1,h)}$, 10 of them, 20 or 0. They find that the best results are achieved by the DNN using all $SD_{vi}^{(d-1,h)}$, supporting their method.

2.3.3.3 Limitations of the X-Model

[Ziel, 2016]'s method is the only attempt to mix the Data-Driven approaches and Expert Models in a single model. However, it is incomplete in many ways but illustrates well the difficulty of producing forecasts that are both correct and consistent with the market.

- First, the averaging on all historical dataset $\overline{AC}$ will distribute the $\mathbb{P}_i$ around a price area that is not representative of the distribution of $Y^{(d,h)}$ that are subject to high seasonality. Projected back to specific auctions, $\mathbb{P}_i$ are not guaranteed to represent important points of $AC^{(d,h)}$.

- Also, the AR model m that is used in [Ziel and Steinert, 2016] and [Kulakov, 2019] is outdated. Modern ML models could be better suited to forecast the extracted points. Also, the obtained metrics are not significantly better than evaluated baselines. The 3 considered studies fail to beat the EXAA benchmark.

- Thereafter, in this method, the forecasting model is trained to minimize the error on SD_{vi}. However, nothing states this will lead to a good price prediction error. The price forecasting error should be taken into consideration while training m. Figure 2.13 illustrates this problem.

- Lastly, the curve reconstruction ρ, based on the occurrences of bids at a specific price in the historical dataset is not representative of the reality because it assumes that bid prices are constant over time. Except for specific plants with a relatively fixed marginal cost (nuclear), plant owners will modify their selling price according to their costs that is dependant on other commodities prices. Similarly, buyers usually adapt their strategies with the current price levels. This method is incompatible with drastic changes that occur in the market (see Figure 2.5). As such, it introduces bias since it's impossible to perfectly reconstruct the AC based on SD_{vi} : $\rho\left(SD_{vi}^{(d,h)}\right) \neq AC^{(d,h)}$ and thus $I\left(\rho\left(SD_{vi}^{(d,h)}\right)\right) \neq Y^{(d,h)}$.

Considering the end task (price forecasting) during curve estimation as well as the kind of selected models affect the correctness of the results. It involves adapting the proposed methodology to lower the price forecasting error. This will not affect consistency because nothing new will be modeled or explained. Oppositely, selecting more relevant points to summarize $AC^{(d,h)}$ or improving reconstruction by repricing supply bids according to commodity prices will not necessarily improve the correctness of the results. However, it increases the consistency of the model since it gets closer to the EUPHEMIA algorithm.

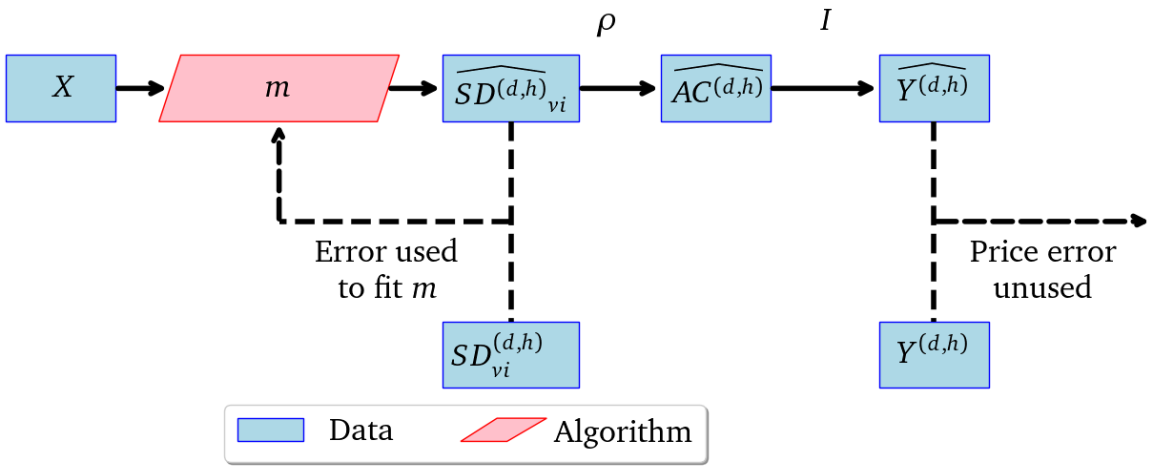


Figure 2.13 – In the Xmodel, the training of the $\widehat{SD}$ forecasting model m only considers the error on $\widehat{SD}$, while the error on the price forecast $\widehat{Y^{(d,h)}}$ is disregarded.

2.3.4 Synthesis

Figure 2.14 summarizes the cited works. It displays the models on a 2-dimensional. The x-coordinate represents the consistency, measured by how many criteria are met. The y-

coordinate represents the correctness, estimated based on the average absolute error. Improvements to these models, either from the literature or discussed in this chapter, are displayed with arrows.

- [Ziel, 2016] and [Maciejowska et al., 2022] studied the possibility to understand the forecasts of their models by analyzing the weights of an AR-X model and feature importance of a RF model. Although limited to models less efficient than DNN, their work reveals the benefits of understanding a model's forecast for feature selection and user confidence, leading to more consistent results. A model-agnostic feature-selection method could increase consistency of any model.
- [Lago et al., 2018b] and [Van der Heijden et al., 2021] have shown that including data from neighboring countries, and performing multi-country forecasting using a single model, significantly improved performances of a LEAR and DNN models while getting closer to the real price fixation mechanism.
- [Ziel and Steinert, 2016], [Kulakov, 2019] and [Schnürch and Wagner, 2019] study a transformation applied to Aggregated Curves that allow them to incorporate information about the bidding structure of the auction in a price forecasting model. Even if the proposed method suffers from various flaws, their proposed X-Model is able to grasp part of the price-fixing mechanism while yielding a reasonable accuracy.

In the literature, the pursuit of correctness is put forward with the abundant use of Data-Driven approaches. As such, price fixation mechanisms, which are better captured by Expert models, are no longer considered. In particular, the EUPHEMIA algorithm, a Mixed-Integer Optimization Problem (MIQP), has never been considered in combination with a ML model in the field of EPF whereas it is the most consistent way of predicting prices. Specifically, we will consider a ML model m that forecasts Order Book, the input of EUPHEMIA. The challenge of such a model is to train the *OB* forecasting model to minimize the price forecasting error, as displayed in Figure 2.15. The next section of this chapter is thus dedicated to the study of the *Constrained Optimization Learning* field, in the objective of integrating the EUPHEMIA logic into a ML model.

2.4 Constrained Optimization Learning

In this part, we focus on methods that will allow us to integrate the EUPHEMIA problem into an EPF model. Many other real-world problems are constrained by technical requirements or physical laws. As such, Constrained Optimization Problems (COP) are the most adequate formulation for many well-known tasks (traveling salesman, knaspack, shortest path, etc.). However, sometimes, parameters of the COP are unknown and estimated using a ML model. In this section, we describe the approaches that learn to forecast the parameters of a CO problem using ML, called *Constrained Optimization Learning* (COL).

2.4.1 Notations

We use the following notation for optimization problems, where the cost function J depends on n optimization variables θ and m fixed coefficients c. The p equality constraints G_i and

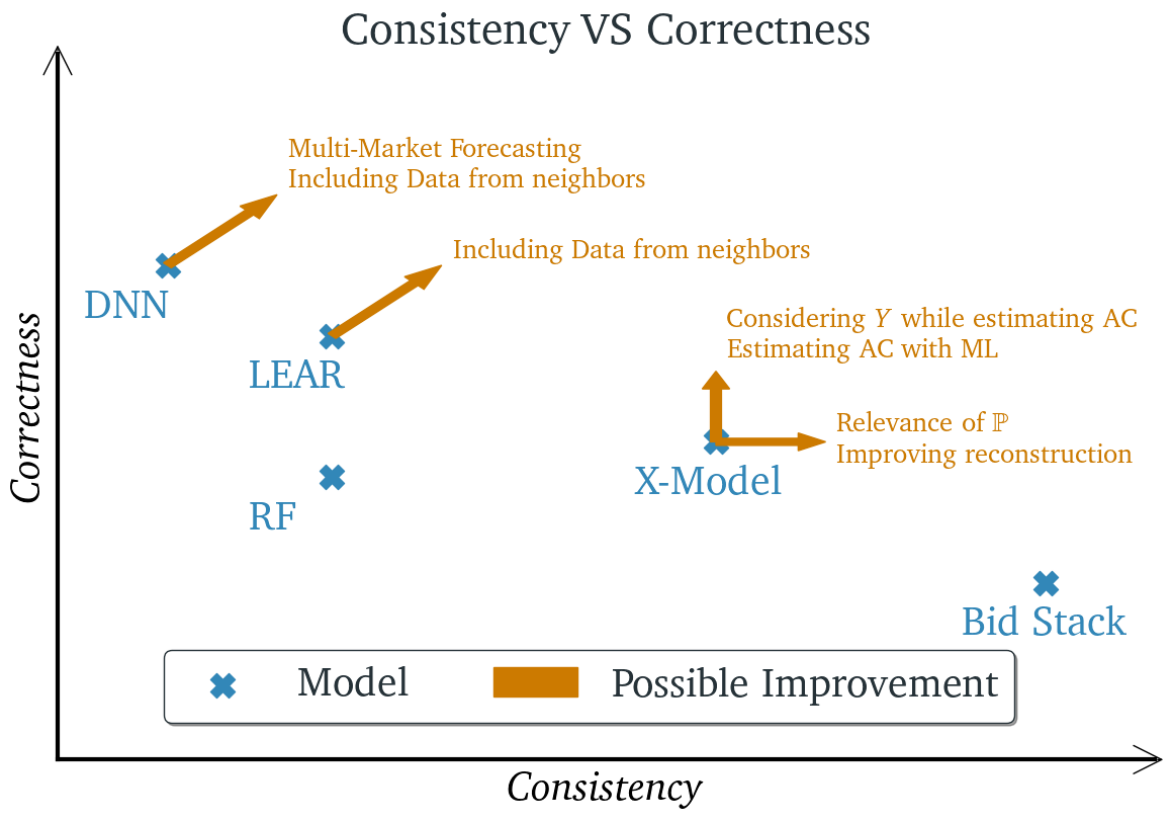


Figure 2.14 – A summary of the current EPF models evaluation against 2 criterion : consistency and correctness. Consistency is estimated based on the various criteria listed in previous sections. Correctness values are based on the values of the price forecasting metrics. Arrows indicate a possible improvement against one or two criteria. Green color is associated with improvement found in the literature.

the q inequality constraints H_i also depend on θ and c.

$$\begin{aligned} \theta^\star = \underset{\theta}{argmax} \;\; & J(\theta, c) \\ \text{u.c} \;\; & G_i(\theta, c) = 0 \;\forall\; i \in 1:p \\ & H_j(\theta, c) \leqslant 0 \;\forall\; j \in 1:q \end{aligned}$$

Under this notation, the Lagrangian of the problem is written:

$$\mathcal{L}(\theta, \lambda, \mu) = J(\theta, c) - \sum_{i=1}^{p} \lambda_i G_i(\theta, c) - \sum_{j=1}^{q} \mu_j H_j(\theta, c)$$

Where λ are the p dual variables associated to the equality constraints and μ are the q dual variables associated to the inequality constraints. The Karush–Kuhn–Tucker (KKT) conditions state that for a solution θ to be optimal θ^*, the following must be observed:

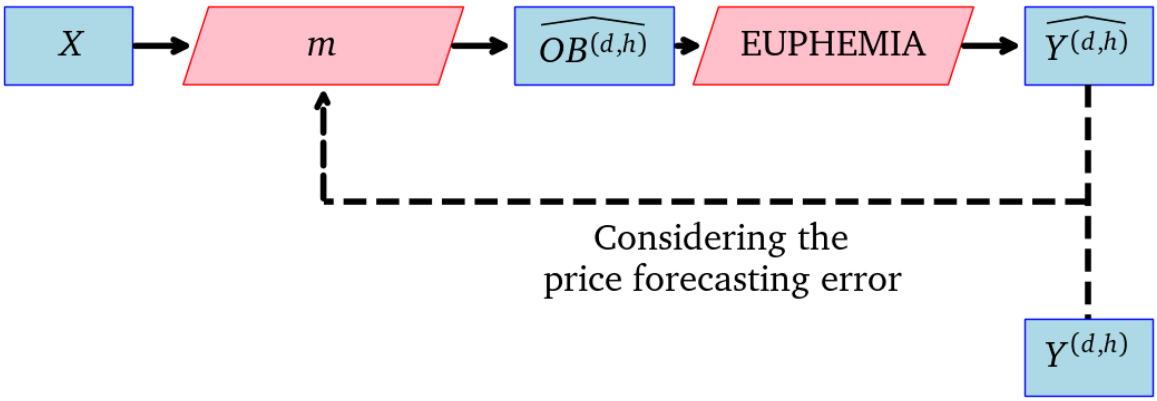


Figure 2.15 – For integrating the EUPHEMIA algorithm in a EPF model, an Order Book forecasting model can be trained to minimize the price forecasting error.

STATIONARITY

$$\frac{\partial \mathcal{L}(\theta, \lambda, \mu)}{\partial \theta} = 0$$

PRIMAL FEASIBILITY

$$G_i(\theta, c) = 0 \forall\ i\ \in 1 : p$$
$$H_j(\theta, c) \leqslant 0 \forall\ j\ \in 1 : q$$

DUAL FEASIBILITY

$$\mu_j \geqslant 0\ \ \forall\ j\ \in 1 : q$$

COMPLEMENTARY SLACKNESS

$$\sum_{j=1}^{q} \mu_j H_j(\theta) = 0$$

DUAL FUNCTION

$$\mathcal{D}(\lambda, \mu) = \min_{\theta}\ \mathcal{L}(\theta, \lambda, \mu)$$

DUAL PROBLEM

$$\lambda^\star, \mu^\star = \min_{\lambda, \mu}\ \mathcal{D}(\lambda, \mu)$$

Lastly, for clarification, we note the solution to the COP $\theta^\star$ given a value of the coefficients c as $\theta^\star(c)$.

2.4.2 Predict-then-Optimize

Many solvers exist to solve COPs for various cases of J, G and H. However, most of them assume that the parameters c are known. In real-world situations, this is not always true. The most straightforward method for tackling unknown c consists in a two-stage approach. First, a ML model is trained to estimate c using exogenous variables X : $\widehat{c} = m(X)$. Typically, a loss function that measures the distance between the real and forecasted coefficient $L(c, \widehat{c})$ is minimized. Coefficients can be estimated by a Neural Network based on exogenous data X. In such case, the gradient of the loss with respect to the input variables is computed using the chain rule for all layers of the network.

$$\begin{aligned}\frac{\partial L(c,\widehat{c})}{\partial X} &= \frac{\partial L(c,\widehat{c})}{\partial c} \underbrace{\frac{\partial c}{\partial X}}_{=0} + \frac{\partial L(c,\widehat{c})}{\partial \widehat{c}} \frac{\partial \widehat{c}}{\partial X} \\ &= \frac{\partial L(c,\widehat{c})}{\partial \widehat{c}} \frac{\partial m(X)}{\partial X}\end{aligned}$$

Once training is finished, the optimization problem can be solved using the forecasted coefficient, and the solution is $\theta^\star(\widehat{c})$. This approach is called *Predict-then-Optimize* and has been used to solve various problems. In our case, it is the approach of the XModel described in [Ziel and Steinert, 2016]. The drawback of this approach is that it does not consider the interplay between an error in the estimation of c and its impact on the OP solution. The final quantity to minimize is the difference in the solutions to the OP $L(\theta^\star(c), \theta^\star(\widehat{c}))$, whose gradient with respect to the input data $\frac{\partial L(\theta^\star(c),\theta^\star(\widehat{c}))}{\partial X}$ is not incorporated in the gradient descent. In most cases where the COP is expressed as a minimization problem, $L(\theta^\star(c), \theta^\star(\widehat{c}))$ is also called *regret* and simply consists of the difference between the sub-optimal cost induced by a decision based on $\widehat{c}$ and the optimal cost with c : $J(\theta^\star(\widehat{c}), c) - J(\theta^\star(c), c)$.

2.4.3 Decision Focused Learning

The body of methods that aims to consider the OP cost (i.e. the decision taken based on c) while training a ML model is called *Decision-Focused Learning* [4]. The main difficulty in integrating the *regret* in the learning process of a NN is the determination of its gradient with respect to the input features. The gradient can be expressed as:

$$\begin{aligned}\frac{\partial L(\theta^\star(c),\theta^\star(\widehat{c}))}{\partial X} &= \frac{\partial L(\theta^\star(c),\theta^\star(\widehat{c}))}{\partial \theta^\star(c)} \frac{\partial \theta^\star(c)}{\partial c} \underbrace{\frac{\partial c}{\partial X}}_{=0} + \frac{\partial L(\theta^\star(c),\theta^\star(\widehat{c}))}{\partial \theta^\star(\widehat{c})} \frac{\partial \theta^\star(\widehat{c})}{\partial \widehat{c}} \frac{\partial \widehat{c}}{\partial X} \\ &= \frac{\partial L(\theta^\star(c),\theta^\star(\widehat{c}))}{\partial \theta^\star(\widehat{c})} \frac{\partial \theta^\star(\widehat{c})}{\partial \widehat{c}} \frac{\partial m(X)}{\partial X}\end{aligned}$$

The first term $\frac{\partial L(\theta^\star(c),\theta^\star(\widehat{c}))}{\partial \theta^\star(\widehat{c})}$ can easily be computed knowing the loss function L. The last term $\frac{\partial m(X)}{\partial X}$ is the standard gradient for Neural Networks. Only the second term $\frac{\partial \theta^\star(\widehat{c})}{\partial \widehat{c}}$ is non-trivial to express because it requires differentiating argmin. A graphical summary is displayed in Figure 2.16.

[4] In [Teso et al., 2022], a Dagstuhl seminar report, the group of experts recommend to use decision-focused learning instead of predict-and-optimize [El Balghiti et al., 2019; Mandi et al., 2022] or predict+optimize[Demirovic et al., 2020] because of some confusion around the terminology.

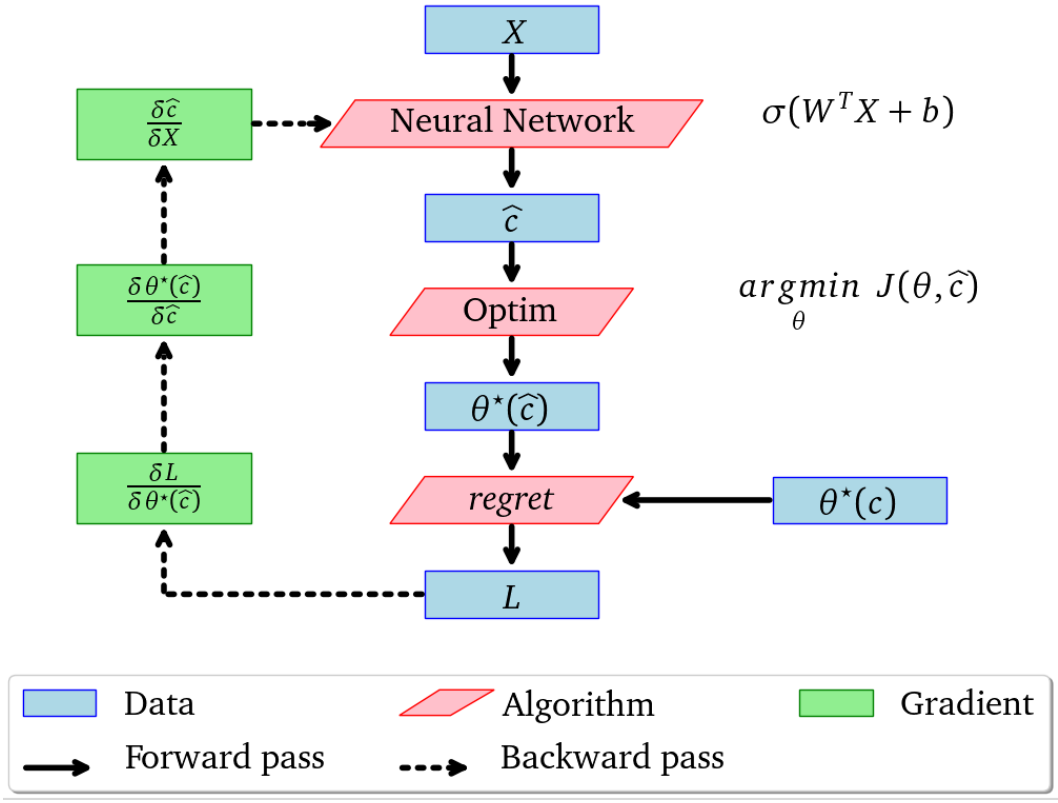


Figure 2.16 – In *Decision-Focused Learning*, the *regret* is taken into account during the training of the Neural Network that forecasts $\widehat{c}$. The chain rule is applied as usual and the gradient of the solution to the optimization problem with respect to its input $\frac{\partial \theta^*(\widehat{c})}{\partial \widehat{c}}$ has to be estimated.

2.4.3.1 Differentiable QP

In the literature, most studies are centered [Amos and Kolter, 2017]'s work. They consider a Quadratic Optimization problem (QP) with linear constraints:

$$\begin{aligned} \theta^\star = \underset{\theta}{argmin}\ & \frac{1}{2}\theta^T Q\theta + q^T\theta \\ \text{u.c}\ & A\theta - a = 0 \\ & B\theta - b \leqslant 0 \end{aligned}$$

With parameters $Q \in \mathbb{R}^{n\times n}$, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$, $a \in \mathbb{R}^n$ and $B \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ previously estimated by a Neural Network. They first express the Lagrangian and express the Stationarity

and PRIMAL FEASABILITY KKT condition:

$$\begin{aligned}\mathcal{L}(\theta,\lambda,\mu) &= \frac{1}{2}\theta^T Q\theta + q^T\theta + \lambda^T(A\theta - a) + \mu^T(B\theta - b)\\ \frac{\partial\mathcal{L}(\theta,\lambda,\mu)}{\partial\theta} &= Q\theta + q + A^T\lambda + B^T\mu = 0\\ A\theta - a &= 0\\ D(\mu)(B\theta - b) &= 0\end{aligned}$$

with $D(\mu)$ the diagonal matrix formed with vector μ. Then, they write the derivative of the KKT in a matrix form:

$$\begin{bmatrix} Q & B^T & A^T \\ A & 0 & 0 \\ D(\mu)B & D(B\theta - b) & 0 \end{bmatrix} \begin{bmatrix} d\theta \\ d\lambda \\ d\mu \end{bmatrix} = - \begin{bmatrix} \theta dQ + dq + \lambda dA^T + \mu dB^T \\ \theta dA - da \\ D(\mu)\theta dB - D(\mu)db \end{bmatrix}$$

They deduct the expression of the gradients with respect to $L(\theta^\star(c), \theta^\star(\widehat{c}))$ of the different parameters of the QP by multiplying the inverse of the left-hand side matrix with the previous backward pass vector $\frac{\partial L(\theta^\star(c),\theta^\star(\widehat{c}))}{\partial\theta(\widehat{c})}$. Their approach showed a significant improvement over *Predict-then-Optimize* on small experiments (sudoku, MNIST). In another paper, [Donti et al., 2017] consider a battery optimization algorithm that decides when to charge or discharge based on estimated electricity prices. Using a stochastic method similar to [Amos and Kolter, 2017], they also demonstrate a small improvement over *Predict-then-Optimize*.

[Wilder et al., 2019] extend this method to Linear Programming (LP) problems by introducing a quadratic regularization term $\gamma||\theta||_2^2$ in the cost function to make the optimal solution twice differentiable with respect to $\widehat{c}$, converting the LP into a QP. They then apply the method of [Amos and Kolter, 2017] to compute $\frac{\partial\theta\widehat{c}}{\partial\widehat{c}}$. During testing, the γ regularization is turned down to 0 to output exact solutions to the problems. Their work assumes that the constraints of the problem are not dependant on $\widehat{c}$. They show significant improvements over the two-stage *Predict-then-Optimize* methods.

Similarly, [Mandi and Guns, 2020] proposes a Log-Barrier regularization term, expressed as $\gamma\left(\sum_{i=1}^{N} ln(\theta_i)\right)$. This term ensures double differentiation, and is consistent with the regularization term used in primal-dual interior point solvers. Lastly, instead of differentiating the KKT matrix, they use the Homogeneous Self Dual formulation(HSD) to compute $\frac{\partial L(\theta^\star(c),\theta^\star(\widehat{c}))}{\partial\widehat{c}}$. They fond that their approach yields better results than a *Predict-then-Optimize* approach and [Wilder et al., 2019]'s method on the knapsack and shortest path problems. They also find that [Elmachtoub and Grigas, 2022]'s differentiable QP problem is more performing on the simpler tested problem (knapsack) whereas their own solution is preferable for more complicated problems (shortest path).

In a similar manner, [Ferber et al., 2020] also tackles a MILP problem starting with its LP relaxation. If the solution is also feasible for the MILP, the algorithm stops. If not, a cut removes the fractional solution and this translate as the introduction of another constraint set $S_k(\theta) \leqslant 0 \ \forall\ k \in [1, K]$. Once an integral solution is found, they compute the differential of the gradient of the solution with respect to the estimated parameters using [Amos and Kolter, 2017]'s methodology, and adding the squared regularization term of [Wilder et al.,

2019]. They compare their method to the *Predict-then-Optimize* approach and with the direct relaxation of their MILP problem without cutting planes. This last is equivalent to [Mandi and Guns, 2020] with the usual KKT matrix approach of [Amos and Kolter, 2017]. They found that their method significantly outperforms the others on combinatorial portfolio optimization, diverse bipartite matching and unit commitment problems. They also investigate the case where both the coefficient loss $L(c, \widehat{c})$ and the optimization loss $L(\theta^\star(c), \theta^\star(\widehat{c}))$ are taken into account in the gradient descent, with a parameter α that trades off between accurate coefficients and optimal decision : $L = \alpha L(c, \widehat{c}) + (1 - \alpha) L(\theta^\star(c), \theta^\star(\widehat{c}))$. However, this approach leads to worse $L(\theta^\star(c), \theta^\star(\widehat{c}))$ than with $\alpha = 0$ or the relaxed solution. Moreover, the choice of α is not discussed.

The KKT differentiation method, introduced by [Amos and Kolter, 2017], involves the factorization of the KKT matrix of size $(n * p * q) \times (n * p * q)$ that cannot scale to large problems. Authors recommend limiting it to cases with less than 1000 hidden dimensions. Moreover, regularization terms must be applied to linear cost functions for this method, adding a hyper-parameter to tune.

2.4.3.2 Smart Predict-then-Optimize

[Elmachtoub and Grigas, 2022] employs a different method. They focus on defining a loss function E_{ML} for the ML model that takes into account the COP loss $L(\theta^\star(c), \theta^\star(\widehat{c}))$. For this, they consider a simple LP minimization problem where $J(\theta, c) = c^T \theta$. The *regret* can then be expressed as $L(\theta^\star(c), \theta^\star(\widehat{c})) = c^T \left[\theta(\widehat{c}) - \theta(c)\right]$. They show that the surrogate loss function

$$\begin{aligned} L_s &= S(c - 2\widehat{c}) + \left[2\widehat{c} - c\right]^T \theta(c) \\ S(c) &= \max_{\theta} c^T \theta \end{aligned}$$

gives an upper bound of the best achievable regret : $L(\theta^\star(c), \theta^\star(\widehat{c})) \leqslant L_s$. They also find that the gradient of L_s with respect to $\widehat{c}$ can be approximated by the expression $2(\theta(c) - \theta(2\widehat{c} - c))$. They show that for an optimal path or portfolio optimization problem, using a simple linear model for predicting c with their surrogate regret function is enough to outperform L1 (Absolute Loss) and L2 (Least Square) losses. Interestingly, the linear model trained with L_s also surpasses the *Predict-then-Optimize* approach with a stronger c forecasting model (Random Forest).

[Mandi et al., 2020] pursues research in this direction. They replace the linear and RF models of [Elmachtoub and Grigas, 2022] by a Neural Network and train it using the loss L_s. Particularly, they investigate on means to reduce the computational burden of solving an optimization problem at each training epoch. Based on the knapsack and energy scheduling problem, they find that using the continuous relaxation of MILP problems achieves relatively similar performances compared to un-relaxed problems, in a much faster time. Using L_s outperforms the differentiable QP method of [Amos and Kolter, 2017] and [Wilder et al., 2019].

Lastly, [Mandi et al., 2022] generalizes the DFL problem to a Learning To Rank (LTR) problem, where the goal is to rank items correctly given a query. Given a pool of feasible

solution to an COP, the ML model learns to rank them by cost order. Results are mitigated as the developed model fails to outperform [Wilder et al., 2019] or [Elmachtoub and Grigas, 2022] approaches on the studied problems. This is later extended to the full range of ML for CO problems in [Teso et al., 2022].

2.4.4 Synthesis

Constrained Optimization Learning is a research field that aims to bridge Optimization Problems and Machine Learning. The two main methods are the differentiable QP of [Amos and Kolter, 2017] and its adaptation to LP and MILP/MIQP, as well as the SPO Loss defined in [Elmachtoub and Grigas, 2022]. Both methods are applied only to small problems (less than 1000 dimensions) because they require solving the optimization problem at every training step that is computationally expensive. Moreover, some proposed methods consider that only the cost function is dependant on previous layer's outputs, ignoring the constraints. The EUPHEMIA algorithm which we aim to incorporate into a ML model has $|OB_z^{(d,h)}|$ orders for each zone z and hour h. This numbers in dozens of thousands of orders, each having one optimization variable A. Additionally, the domain constraints of A, which are either bound to or between 0 and 1, double this number. Moreover, BLOCK ORDERS, COMPLEX ORDERS, AVAILABLE TRANSFER CAPACITIES and ENERGY BALANCE can add several thousands of constraints per auctions. Clearly, the methods proposed in the *Constrained Optimization Learning* field cannot be directly adapted to the EPF field.

2.5 Conclusion

Forecasting day-ahead electricity prices is a complex task. Because electricity prices are set using a Price-Fixing Algorithm, the relationship between fundamental variables and the price is not intuitive. In Europe, the EUPHEMIA algorithm is used to compute the prices which generate the most social welfare, while ensuring that the network constraints are satisfied. EUPHEMIA uses an MIQP optimization problem that sets the prices Y_z^d for a day d of all considered zones z by selecting which order to accept and determining the cross zonal exchanges.

The EPF field is an important domain that has attracted many scientists over the past decades, and the combination of market deregulation and energetic transition policies exacerbate this phenomenon. Traditionally, Expert models approaches replicate EUPHEMIA and solve it by feeding estimated Order Books. The latter are obtained by defining bidding strategies at hand, based on variables (gas price, temperature, ...) predicted using stochastic methods. Nevertheless, those methods fail to produce accurate results in practice. The prediction of variables induces imprecision which can reverberate badly on the price computation because using estimated Order Books can drastically disrupt the supply-demand equilibrium. Oppositely, Data-Driven approaches completely occult the PFA and learn relationships between prices and drivers. Those methods have shown to outperform Expert models approaches [Weron, 2014] when comparing predictive accuracy. In particular, Machine Learning models such as the *DNN* introduced by [Lago et al., 2018a] can significantly outperform traditional Expert models approaches as well as Statistical Models.

However, there are still many challenges to tackle in this domain; the ML models suffer from several drawbacks. First, their opacity and possible non-intuitive results can make them difficult to employ by business practitioners. There is a need for understanding which variables are considered by the model and to explain a forecast. This is paramount for building trust of the business practitioners towards the model. Then, because EUPHEMIA computes the prices jointly, the EPF task has to be considered on several countries at once. Many key topological aspects of the European network, such as the energy exchanges, have not yet been taken into account while predicting prices. Lastly, Data-Driven approaches are bound to an inherent limit because they occult the PFA. This brings forward the necessity of considering EUPHEMIA in combination with a Data-Driven approach. The latter has been introduced by [Ziel and Steinert, 2016] but the employed method uses an approximate of EUPHEMIA that is too simple, in combination with outdated statistical models. We believe that using a ML model for predicting Order Books before solving EUPHEMIA is an ideal solution to the EPF task. This approach is not without difficulties. Order books prediction errors can reverberates badly on the optimization result. To minimize this effect, differentiable optimization methods [Amos and Kolter, 2017] consider the difference in optimization results in the training process. The proposed methods are difficult to apply to EUPHEMIA because of its high dimensionality and MIQP nature.

Because the EPF litterature focuses on lowering the prediction error, many key aspects of the problem are not considered. In this thesis, we tackle the EPF problem with multiple objectives. First, we aim to produce explainable results, while mainaining a good accuracy. Then, we consider the European market as a whole, and we model the energy exchanges between countries. Lastly, we aim to integrate EUPHEMIA in a price forecasting model.

Chapter 3

Extending Data-Driven methods for EPF

Data-Driven methods are the best approach in the literature. However, no study evaluates several models on several markets, and the employed datasets are outdated or limited in terms of considered variables. Also, those methods consist of training black-box models that are not satisfactory for business practitioners. In this Chapter, we consider five ML models that we evaluate on three markets using two different datasets. We show that the features added in the second dataset significantly improve the quality of forecasts, even in the current period when sudden price changes are occurring. We also develop an analysis of the contribution of the different features in model predictions using Shap values, in order to shed light on how models make their predictions and to build user confidence in models.

3.1 Introduction

In Section 1.2 of the Introduction, we presented several use-cases of Electricity Price Forecasts that motivated this thesis. Particularly, we presented the problem of trading on the Day-Ahead market, which requires forecasts as accurate as possible, for a day d on time, before 12am in $d-1$. Given that the trader has to make a decision based on the forecast, he immensely benefits from an explanation of the obtained price with respect to the input variables. Such explanations allow one to assess risks and provide confidence in the model. The business activities conducted by BCMEnergy on the French market, as well as the Islander project that takes place in the German market, motivate us to forecast prices from several countries. In the preceding Chapter, we presented the different Data-Driven approaches for predicting the Day-Ahead prices in Section 2.2.1. Particularly, we highlighted several flaws of those approaches in Section 2.3. We stated that the state-of-the-art papers compared either several models on a single market ([Lago et al., 2018a]) or 2 models on several markets ([Lago et al., 2021]), limiting the generalization of their findings. In addition, they studied an outdated test period (the year 2016) with prices in Europe that are not representative of the current situation, because it omits the changes provoked by the Covid lockdown. Moreover, we stated that Data-Driven approaches suffer from their opacity and we emphasized the need for model-agnostic individual explanations of the forecasts. Using the current methods proposed by the literature, it is impossible to fulfill all requirements induced by the nature of industrial activities based on price forecasts.

3.1.1 Contributions

Following the guidelines introduced in a recent publication [Lago et al., 2021], we apply a rigorous, transparent, and reproducible methodology for using ML models for EPF. We evaluate 5 ML models over three different areas of Europe on two separated test periods. By doing so, we challenge [Lago et al., 2018a]'s work which evaluated 24 different models on a single (Belgian) market and concluded that the *DNN* model was the most accurate one. We find that the best-performing model varies according to the studied market. Next, by incorporating new features on the training dataset (neighboring prices and gas prices), we show that predictive accuracy can significantly increase for all models and that the *DNN* model can be outclassed by a simpler *SVR* model in all studied markets, including Belgium. We also update the test period to one more representative of the current situation. We show that the ML models are capable of correctly adapting to recent electricity prices that have been greatly affected by the Covid lock-downs of 2020 or by the energy crisis of 2021. Interestingly, the most performing model of each country remains the same on the updated period.

We then investigate what explains the performance of a particular model on a given dataset by conducting a feature analysis based on Shap values [Lundberg and Lee, 2017]. Using these tools, we find that the feature responsible for the performance increase are the Swiss prices. This is even the case for the Belgian dataset whereas Belgium and Switzerland share no border. We show that the past occurrences of features contribute very little to the forecast and suggest that they could be removed. Lastly, we also analyse individual explanation i.e. we explain each forecast independently. We focus on explaining the fore-

casting errors during the successive lock-downs in France on the period 2020-2021. We find that relying on the gas price on $d-1$ could be misleading when the latter also suffers from increased volatility.

3.1.2 Chapter Structure

This Chapter is structured as follows. As a starter, we detail the specificity of our datasets in Section 3.2. Due to data availability, we differentiate the features used in the two time periods considered in our experiments. Then, we present the technical requirements on ML models, as well as their evaluation in Section 3.3. We also explain our explainability approach used to analyze black-box ML models : the Shap values. Section 3.4 reports our results. We first analyze the quality of the models using metrics, backed with statistical tests. Then, we give the results of the explanations of the predictions, by analysing the importance of each feature for different models. We also observe the temporal changes in feature importance throughout the years 2020 and 2021 and comment the effects of the successive lock-downs.

3.2 Datasets

Many multivariate time series forecasting research articles [Bagnall et al., 2017; Ruiz et al., 2021] recommend evaluating models on several datasets as the behavior of an algorithm can be very different depending on inherent characteristics of the dataset. The relative performance of several models can even vary and considering a large number of datasets makes it possible to have a more robust evaluation of the model performance. To assess the specific qualities of a model, it is therefore relevant to consider datasets from different countries. Indeed, the energy mixes are very different from one country to another and have a strong influence on the dynamics of the prices of electricity. To build predictive models of electricity prices, we extend the classically considered datasets [Lago et al., 2021], called hereafter SOTA, by adding new attributes as predictive features and considering more recent data. These datasets and their specificity are presented below.

3.2.1 Markets

We consider the three following markets: France (FR), Germany (DE), and Belgium (BE). These countries are at the same time geographically close yet have features that make them unique. In this section, we state their peculiarities by detailing their generation by source, as well as what drives consumption. Figure 3.1 summarizes the energy mix per country, using the data provided by the International Energy Agency, for the year 2021 [1]. Figure 3.2 displays the hourly Day-Ahead prices for those markets over the years 2016-2022. Lastly, Figure 3.3 displays the yearly average of exchanged energy between the three countries.

3.2.1.1 The French Market

Nuclear power plants account for roughly 71% of the average 555TWh of annual produced electricity. They are to some extent controllable and usually provide enough energy to satisfy

[1] https://www.iea.org/, 23-10-2023

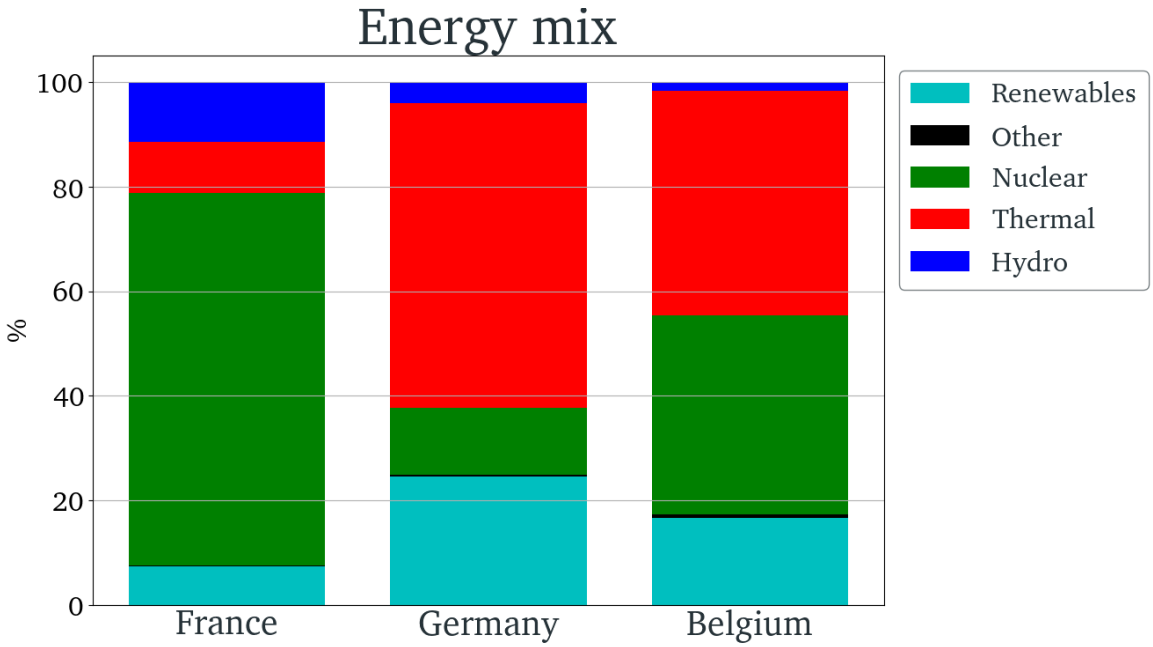


Figure 3.1 – Average Energy mix in France, Germany and Belgium from 2016 to 2019. Thermal plants consist in Gas, Oil, Coal an Waste powered plants. Renewables are Wind, Solar and Tide power plants.

the consumption and some surplus is sold in Germany, Spain or Italy. However, maintenance operations, material failure, strikes or even cooling river temperatures can significantly lower the generation capacity, making it mandatory to buy energy from neighbors or to turn on much more expensive gas, oil or coal-fired thermal stations. Lastly, the hydro plants, producing during the spring when the snow melts, increases the gap between winter and summer prices. On the other hand, the consumption is mainly heat-sensitive due to the massive use of electric heaters, leading to higher prices in the winter period. French prices are usually lower than its neighbors but extreme price spikes can occur during cold winters when nuclear power plants are under maintenance.

3.2.1.2 The German Market

The German generation is based on thermal plants for approximately 58% of its average 610TWh of annual production. The electricity prices are thus continuously dependent on gas, oil and coal, but also on the price of carbon emissions, whose allowances are sold on an open market. Furthermore, wind turbines and solar panels play a major role in the price volatility because their total generation can't be established in advance: negative prices are commonly reached. Traditionally, Germany exports energy to its neighbor (mainly France) when wind generation is high but imports nuclear energy from France when it's low. The consumption, less based on heaters, is more stable than in France.

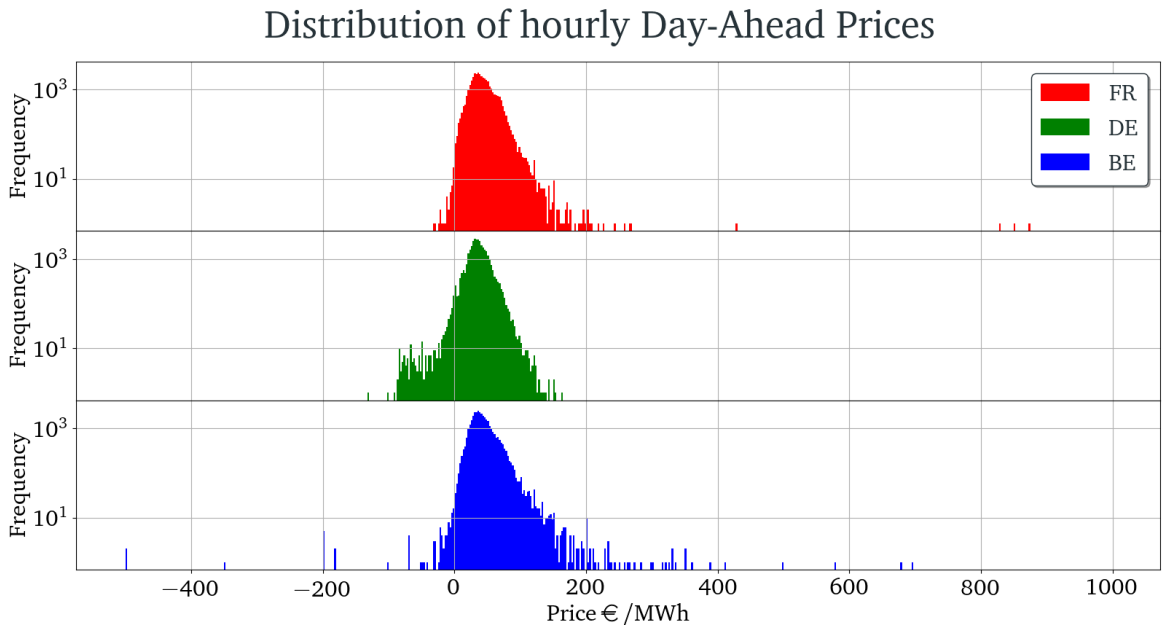


Figure 3.2 – Price histograms over the period 2016-2019 for France, Germany and Belgium. We observe price spikes in France while Germany and Belgium display negative prices.

3.2.1.3 The Belgian Market

Nuclear and thermal generation account for almost 80% of Belgium's generation. Due to the small size of this market (consumption of 80TWH yearly on average), prices can sometimes reach extreme negative prices when generation peaks. Interestingly, since the opening of the connection line with Germany, its sometimes used to send energy from France to Germany when the line between both countries is full. Prices in Belgium are also sometimes negative because of the important part of wind turbine generation.

3.2.2 State-of-the-art datasets

We first consider the multivariate time series made of daily data from [Lago et al., 2021]. Each dataset includes next day prices for the period $T1$ (from 2011 to 2016) and has two additional exogenous features given in Table 3.1. Typically, the features used are national or regional consumption and generation forecasts, published by the Transmission System Operators (TSOs) several hours before price fixation. The ENTSOE transparency platform [2] aggregates this data, as well as the Day-Ahead prices, for all European markets.

Those datasets can be reconfigured into a (X, Y) pair suitable for training machine learning models. The predictive data is represented by a two dimensional matrix $X \in \mathbb{R}^{n_d \times n_c}$ whose rows represent days and columns are n_c predictive time-dependent values. The values to be predicted correspond to another matrix $Y \in \mathbb{R}^{n_d \times 24}$, where rows also stand for the days and columns are the 24 day-ahead prices to be predicted: $Y_d = \left(Y^1_{d+1}, \ldots, Y^{24}_{d+1}\right)$. Typically, a row of Y contains 24 prices, corresponding to the 24 hour prices to be predicted the next

[2] https://transparency.entsoe.eu/

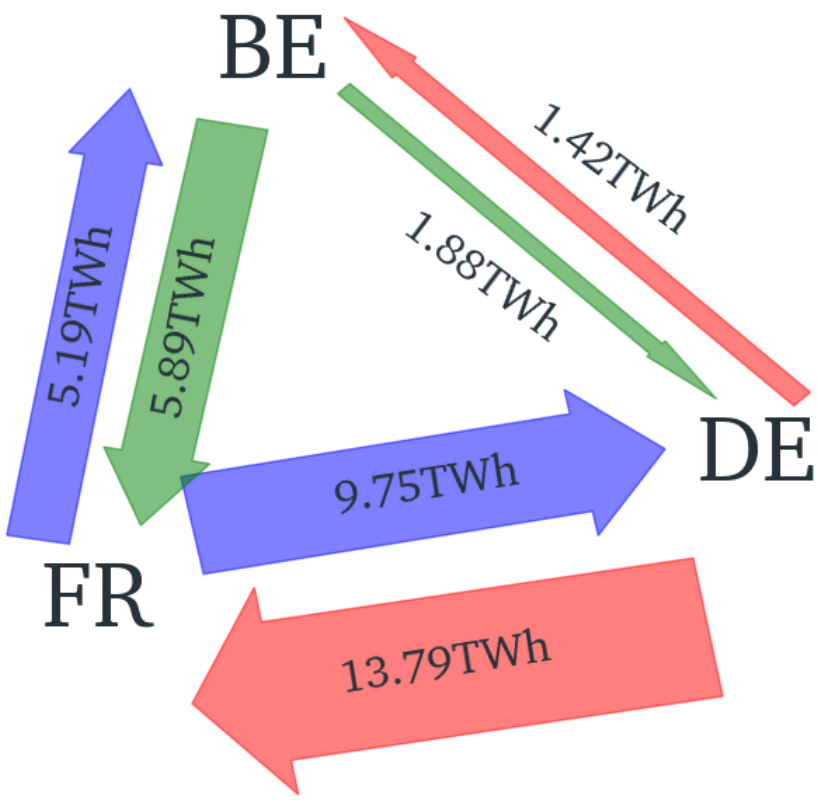


Figure 3.3 – Average yearly exchanges between France, Germany and Belgium. As there were no connections between Belgium and Germany prior to 2020 and because we don't integrate energy exchanges in our datasets, this plot shows the years 2020-2021.

day. To model the time series aspect of the features, X includes the prices of the current day, those of the day before, two days before and the previous week (1, 2, 3 and 7 days lag). Exogenous features are included for the current day, the day before and the previous week. Those values where chosen because of the similarity between two consecutive days, or between the same day of two consecutive weeks.

Besides these 240 characteristics, the day of the week is also encoded as an integer and added to the matrix X. Indeed, electricity prices are non-stationary time series and exhibit seasonal trends captured by this additional feature. All features (prices and exogenous) are provided with hourly granularity. Consequently, the predictive matrix X is defined as:

$$X_d = (Y_{d-1}, Y_{d-2}, Y_{d-3}, Y_{d-7}, E1_d, E1_{d-1}, E1_{d-7}, \\ E2_d, E2_{d-1}, E2_{d-7}, \text{DayOfWeek}) \text{ with } n_c = 241.$$

In order to forecast 24-hour prices for the next day, the datasets are reshaped so that for one day d, Y_d contains all 24 prices for the next day: $Y_d = \left(Y^1_{d+1}, \ldots, Y^{24}_{d+1}\right)$.

Dataset	Exogenous input 1	Exogenous input 2
FR	Consumption forecast	Production forecast
BE	French consumption forecast	French production forecast
DE	Amprion consumption forecast	Amprion, TenneT, 50 Hertz renewable forecasts

Table 3.1 – Exogenous inputs of EPFTOOLBOX dataset. Each dataset is composed of the Day-Ahead prices for the specified country and 2 exogenous features.

3.2.3 Enriched datasets

For the enriched datasets considered in this study, we incorporate new features to the SOTA, datasets. We also modify the training and testing period to more recent data : $T2$ spans from 2016 to 2021. From these data we build four datasets, three (FR, DE, BE) comprising the data of each country taken individually, and a fourth (Multi-Output) merging together the data of the three countries. With this dataset, we seek to forecast the prices of the three countries at once. Due to the pricing algorithm, all European prices are set at the same time and we want to model this phenomenon.

Features	FR		DE		BE	
	T_1	T_2	T_1	T_2	T_1	T_2
French Prices	Target	Target	✓	✓	✓	✓
German Prices	✓	✓	Target	Target	✓	✓
Belgian Prices	✓	✓	✓	✓	Target	Target
Dutch Prices	✓	✓	✓	✓	✓	✓
Spanish Prices	✓	✓				
Swiss Prices	✓	✓	✓	✓		
French Consumption Forecast	✓	✓	✓	✓	✓	✓
German Consumption Forecast	✓	✓	✓	✓	✓	✓
Belgian Consumption Forecast		✓		✓		✓
French Production Forecast	✓	✓	✓	✓	✓	✓
German Renewable Energy Forecast	✓	✓	✓	✓	✓	✓
Belgian Renewable Energy Forecast		✓		✓		✓
French Gas Prices		✓		✓		✓
Date Dummies	✓	✓	✓	✓	✓	✓

Table 3.2 – Composition of the datasets for each country and the two time periods.

3.2.3.1 Data from neighboring countries

The electricity day-ahead price is fixed by EUPHEMIA through the coupling of different markets where energy transactions can involve sellers and buyers from different countries, only limited by the constraints of the electricity network. All bilateral interconnections make it possible to transport less expensive production assets from one country to another with an important demand. Thus, the price within a country is highly dependent on exogenous factors in surrounding countries. This is why we have included production and consumption forecasts from neighboring countries in our datasets. Similarly, we used Dutch, Spanish and Swiss prices. Swiss prices are attractive as they are available every day at 11.15 am and can

be used in a forecasting model before the European market closes at noon. Consumption and generation forecasts from the day to predict, the day before and one week before are included. Prices of the day before and one week before are included. Swiss prices of the day to predict are also included.

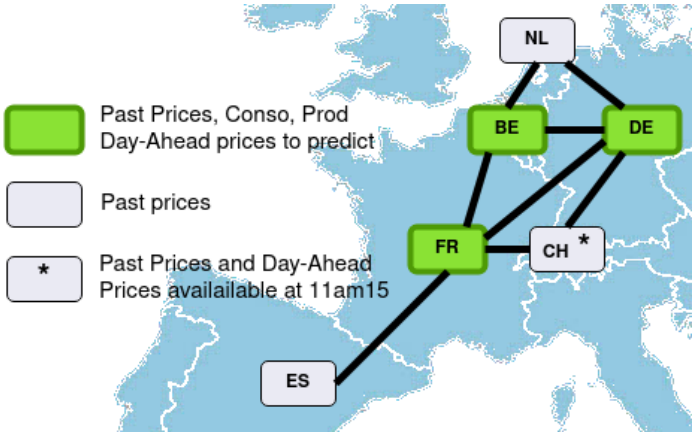


Figure 3.4 – Markets corresponding to the enriched datasets. In green are the markets whose Day-Ahead prices are forecasted and their Past Prices, consumption and generation forecasts are in the training dataset. We only include Past Prices (or current if available) of the markets in grey.

3.2.3.2 Date indicators

Another aspect that can strongly influence the prediction is the date, especially the days of the week that involve differentiated human activity and therefore impact energy consumption and production. But, as shown by [Cartea and Figueroa, 2005], the seasonality of the electricity market is not only dependent on the day of the week. We therefore propose to incorporate various date dummy variables into our enriched dataset. We decided to include weekday, week number, day of month and month number as predictive functions. To better integrate these cyclic data into our ML models, we apply a circular encoding transformation $\mathcal{F}$ of a cyclic feature that encodes the original feature of the domain value C (with cardinality α) into two numeric values:

$$\begin{aligned} \mathcal{F} : C &\mapsto \mathbb{R}^2 \\ x &\mapsto (\sin(\frac{2\pi x}{\alpha}), \cos(\frac{2\pi x}{\alpha})) \end{aligned}$$

3.2.3.3 Gas Prices

Finally, we also integrate gas prices. Indeed, to maximize social welfare, the EUPHEMIA algorithm favors the power plant with the lowest marginal cost. Accordingly, there is an order of merit for the technology of production plants. Gas-fired power plants are one of the cheapest ways of generating electricity among other coal or oil-fired thermal power plants. However, its marginal cost is a function of gas prices. Therefore, depending on the country's energy mix, gas prices are an important feature of electricity prices. We therefore decided to

include the EGSI gas index[3] in our dataset. As this index is available every day at 6pm, after the market closure, it has to be included for predicting prices 2 days after.

3.2.3.4 Datasets dimensions

As previously mentioned, to model the time series aspect of these features, X contains the country's prices for the previous day, those of two days before, three days before and the previous week (1, 2, 3 and 7 days lag). Other features (see table 3.2) are included for the day before, the previous week, and if possible the current day. Indeed, production and consumption forecasts as well as Swiss prices are available for the day to be forecast before noon. Then, with the exception of gas prices and date dummies, all features are included for the 24 hours of the day. The datasets therefore have $n_c = 24 \times n_f^l \times n_f + 8 + 1$ columns, with n_f the number of features as described in Table 3.2 and n_f^l the number of shift days for a given feature f.

3.2.4 Train/Test splits

In [Lago et al., 2021], the authors provide open-access benchmark datasets for the six-year period between 2011 and 2016. A good practice in the field of machine learning is to evaluate models over the same time period to allow comparison of results. We therefore start our analysis by evaluating our models on the same data (see dataset description T_1 in Table 3.2). It is also important to extend our study to the current period, the peculiarities of which serve as an evaluation of the robustness and adaptability of the models in a context of high variability. Therefore, we consider a second dataset with 4 years from 2016 to 2019 for training and two test years from 2020 to 2021 (see dataset description T_2 in Table 3.3). Recent electricity prices present a more difficult challenge for prediction because the lock down related to Covid-19 has caused massive changes in the European market, and in its aftermath, caused small energy crisis in 2021. In addition, since 2015, the ENTSOE transparency platform[4] has brought together and published data from almost all European TSOs in free access. This results in much more available data. The features of our datasets therefore vary depending on the period considered as described in Table 3.2.

Period	Train start	Validation start	Test start	Test end
$\mathbf{T_1}$	2011-01-16	2014-01-07	2015-01-04	2016-12-31
$\mathbf{T_2}$	2016-01-01	2019-01-01	2020-01-01	2021-12-31

Table 3.3 – Time period of data used for training (learning of model parameters), validation (determining hyperparameter values), and testing (evaluating models) for EPFTOOLBOX datasets. As we use the first seven days of the dataset as the input features, the train dataset starts seven days after the first data sample.

[3] https://www.boursorama.com/cours/1rPGTT/
[4] `https://transparency.entsoe.eu`

3.3 Machine Learning for EPF

As we believe that the capabilities of ML models have not yet been fully unraveled in the field of EPF, we focus on these approaches. In particular we consider four different models exposed below. We also present the metrics and tests used to compare them and the way we preprocess data. Additionally, we explain how we fix hyper-parameter values and present the recalibration strategy used to adapt models to recent changes in the data. Finally, we describe the SHAP method that we employ in our analysis to assess the importance of features in the prediction process.

3.3.1 Machine Learning Models

We present the four models that we will use with our own parameters, as well as the two that we borrow from the EPFTOOLBOX.

Support Vector Regressor

Support Vector Machines (*SVR*) [Kampouraki et al., 2008; Orsenigo and Vercellis, 2010] are a category of models with a good mathematical background based on an optimization problem. With the use of kernels [Che and Wang, 2014; Rakotomamonjy et al., 2008], they can be applied on complex data structures and model non-linearity. Originally designed to solve univariate forecasting problems, we adapt them to the multivariate case in two ways: 1) The *ChainSVR* method that uses the first forecast to predict the second one, the second forecast to predict the third one, and so on; 2) The *MultiSVR* that uses one model per time series in Y, so 24 in total. We use the method SVR as implemented in `scikit-learn` [Pedregosa et al., 2011].

Random Forest Regressor

Random Forest Regressor models (*RFR*) are widely used ML models both in the field of EPF [Díaz et al., 2019; Lago et al., 2018a; Mei et al., 2014] and in forecasting tasks in general [Bagnall et al., 2017; Ruiz et al., 2021]. They consist of a combination of several Decision Tree Regressor (DTR) that are trained using different subsets of the data. The Bagging [Breiman, 1996] method used in this paper outputs the average of their predictions. We use `scikit-learn`'s implementation [Pedregosa et al., 2011] of RFR.

Deep Neural Networks

The model capabilities and tremendous range of application made Deep Neural Networks (*DNN*) the center of interest of numerous researchers in EPF [Lago et al., 2018a,b, 2021; Mosbah and El-Hawary, 2016] but also in forecasting tasks in general [Abiodun et al., 2018; Chen and Zhang, 2021; Zhang and Qi, 2005]. The *DNNs* we use have $\ell + 2$ layers stacked sequentially. The number of neurons of the first and the last layer are respectively n_c and 24, the second dimensions of X and Y respectively, the other layers having $(n_1, \cdots, n_\ell)$ neurons. These hyper-parameters $(\ell, n_1, \cdots, n_\ell)$ are set with a grid search. The model is trained using a gradient descent algorithm of the forecast errors back to the network weights.

Convolutional Neural Network

Convolutional Neural Networks (*CNN*) are a variant of Deep Neural Networks which became popular for their image processing capabilities [Badrinarayanan et al., 2015; Krizhevsky et al., 2012; LeCun et al., 1995; Szegedy et al., 2017]. They are now also used for multivariate time series regression tasks [Borovykh et al., 2017; Li et al., 2018; Zheng et al., 2014, 2016] and in particular, for EPF [Cheng et al., 2020; Khan et al., 2020; Lago et al., 2018a]. The eponymous convolutional layers combined with pooling layers are the particularity of *CNNs*. By applying numerous filters on the data, convolutional layers extract complex patterns that are then generalized by a pooling operation to provide complex feature representations of the input. We use the keras[5] implementation with tensorflow [Abadi et al., 2016] back-end to implement our Neural Networks models (*DNNs* and *CNNs*).

EPFTOOLBOX

The LASSO-Estimated Auto Regressive model (*LEAR*) model acts as the State-of-the art Statistical model widely used in the EPF literature for comparison purposes [Lago et al., 2021]. Its formal definition is available in the precedent Chapter2.2.1.2. In addition to the *DNN*, this model is provided with its own hyper-parameters in the EPFTOOLBOX. As we aim to compare our models with those, we also re-run their configurations for four *LEAR* models and four *DNN* models. The *LEAR* models are denoted $LEAR_{52}$, $LEAR_{84}$, $LEAR_{1092}$ and $LEAR_{1456}$, in reference to their respective calibration window size. The *DNN* models are denoted DNN_1, DNN_2, DNN_3, DNN_4.

3.3.2 Evaluation metrics and test

We recall that $Y^{(d,h)}$ is the price for a day d and an hour h of a given zone, and by $\overline{Y}^{(d)} = \frac{1}{24}\sum_{h=1}^{24} Y^{(d,h)}$ the average price of that day. Let $\widehat{Y}^{(d,h)}$ be the values predicted by a model. The comparison of these values is used to evaluate and test the quality of a model, but also to learn it, through the loss function used to adjust the parameters of the model.

3.3.2.1 Metrics

The most commonly used metric to evaluate the quality of a model in the field of EPF is the Mean Absolute Error (*MAE*):

$$MAE(Y, \widehat{Y}) = \frac{1}{n_d}\sum_{d=1}^{n_d}\sum_{h=1}^{24} |Y^{(d,h)} - \widehat{Y}^{(d,h)}|$$

It allows business owners to quickly estimate how they could use a forecasting model to generate profit. However, since electricity prices can range from -500 to 3000 €/MWh in the European markets, it is useful to use a relative error measure. While the Mean Absolute Percentage Error (*MAPE*) is usually used for this purpose, we prefer employ the Symmetric Mean Absolute Percentage Error (*SMAPE*). Indeed, prices close to 0 that are incorrectly

[5] https://keras.io

predicted lead to an unnecessary high *MAPE*, which is not the case with *SMAPE* values:

$$MAPE(Y, \widehat{Y}) = \frac{100}{n_d} \sum_{d=1}^{n_d} \sum_{h=1}^{24} \frac{|Y^{(d,h)} - \widehat{Y}^{(d,h)}|}{|Y^{(d,h)}|}$$

$$SMAPE(Y, \widehat{Y}) = \frac{100}{n_d} \sum_{d=1}^{n_d} \sum_{h=1}^{24} \frac{|Y^{(d,h)} - \widehat{Y}^{(d,h)}|}{\frac{1}{2}\left(|Y^{(d,h)}| + |\widehat{Y}^{(d,h)}|\right)}$$

We also consider a new metric called the Daily Average Error (*DAE*). It consists in computing the MAE between the average predicted price for a day and the real average price. This metric is very useful for trading-related activities, when one speculates on the average price for a given day.

$$DAE(Y, \widehat{Y}) = \frac{1}{n_d} \sum_{d=1}^{n_d} \left| \frac{1}{24} \sum_{h=1}^{24} Y^{(d,h)} - \frac{1}{24} \sum_{h=1}^{24} \widehat{Y}^{(d,h)} \right|$$

Next, to enable cross-dataset comparison, we use the Relative Mean Absolute Error. The idea is to compare the *MAE* of a model with the *MAE* of a naive forecaster. As naive forecaster, we use the following strategy:

$$\widehat{Y}^{(d,h)}_{naive} = \begin{cases} Y^{(d-1,h)} \text{ if } d \text{ is a week day} \\ Y^{(d-7,h)} \text{ otherwise} \end{cases}$$

$$RMAE(Y, \widehat{Y}, \widehat{Y}_{naive}) = \frac{MAE(Y, \widehat{Y})}{MAE(Y, \widehat{Y}_{naive})}$$

Lastly, we use the LogCosH loss function for training Neural Networks models (*DNNs* and *CNNs*). It combines the benefits of both *MAE* and Mean Squared Error by being approximately equivalent to $\frac{(Y-\widehat{Y})}{2}$ when $Y - \widehat{Y}$ is small, and to $|Y - \widehat{Y}| - \log(2)$ when differences are large. Due to the presence of spikes in electricity prices, it is useful not to put too much weight on outliers:

$$LogCosH(Y, \widehat{Y}) = \log(\frac{e^{Y-\widehat{Y}} + e^{\widehat{Y}-Y}}{2})$$

3.3.2.2 Diebold & Mariano Test

We use the Diebold & Mariano [Diebold, 2015; Diebold and Mariano, 2002] test to perform more robust model comparison. Instead of averaging a loss g across the entire dataset, it computes the loss difference d_g between two model predictions Y_1 and Y_2. A one sided z-test is then performed to assess if the second model forecasts are significantly better than the first ones :

$$d_g(Y, \widehat{Y}_1, \widehat{Y}_2) = g(Y, \widehat{Y}_1) - g(Y, \widehat{Y}_2)$$

$$H_0 : \mathbb{E}\left[d_g(Y, \widehat{Y}_1, \widehat{Y}_2)\right] > 0$$

$$H_1 : \mathbb{E}\left[d_g(Y, \widehat{Y}_1, \widehat{Y}_2)\right] \leqslant 0$$

If the obtained p-value is lower than a fixed threshold of 0.05, then H_0 is rejected and we can conclude that the first model is better than the second one. We use the absolute loss $g(Y, \widehat{Y}) = MAE(Y, \widehat{Y})$ in our experiments as it better reflects business applications.

3.3.3 Data preprocessing

Data scaling is critical during ML model training. Most algorithms require that both the input (X) and output (Y) data are pre-processed. To this aim, we design simple data pipelines to process the features and target variables of our datasets. We distinguish the scaler used to process the input data X from the one used for processing the predicted values Y. We consider these two functions as hyper-parameters with four different possibilities for each of them: (1) the standard scaler that standardizes data so it has a 0 mean and 1 variance, (2) the median scaler, a outlier-robust version of standard scaler using the median and median average deviation, and (3) their combination with the *arcsinh* function [Uniejewski et al., 2017] or not:

$$SS(X) = \frac{X - \mu_x}{\sigma_x^2} \tag{3.1}$$

$$MS(X) = \frac{X - median_x}{MAD_x^2} \tag{3.2}$$

$$arcsinh(X, f) = \log\left(f(X) + \sqrt{f(X)^2 + 1}\right), \quad \text{with } f \text{ either SS or MS.} \tag{3.3}$$

3.3.4 Hyper-parameters Search

Despite their high modeling power, ML models suffer from a critical issue, which is hyper-parameter optimization. Hyper-parameters must be configured before training the model on the data. They need to be tuned for optimal results. This is done by testing numerous combinations of hyper-parameters and selecting the optimal one. As this part is very time consuming, we use a Randomized Grid Search [Bergstra and Bengio, 2012] that samples 4000 hyper-parameter combinations for each models in a pre-defined search space. Details of the search spaces as well as the best configuration found are available in the Appendix C.

3.3.5 Recalibration

Another drawback of ML models is their implicit assumption that the future will be similar to the past. However, as seen in Figure 2.5, electricity prices can be very volatile and sudden unpredictable changes can drastically modify the prices, such as the Covid lockdown [Narajewski and Ziel, 2020b] or the European energy gas crisis of fall 2021 [Sheppard, 2021]. Those changes are critical, for example, [Lago et al., 2018b]'s model gets confusing results while forecasting Belgian prices due to a sudden change in the generation patterns. To address such problems, [Mei et al., 2014] uses an online Random Forest method to keep the forecasting model up to date, [Demir et al., 2021] generates more current data using autoencoders and [Lago et al., 2021] uses model recalibration. Recalibration consists in retraining the model with most recent data, that is to say using $X^{(1)}, \cdots, X^{(d-1)}$ and $Y^{(1)}, \cdots, Y^{(d-1)}$ to train the

model before forecasting a new sample $X^{(d)}$, $(X^{(d)}, Y^{(d)}), \cdots, (X^{(d-1)}, Y^{(d-1)})$ being in the test set. However, this method incurs computational costs as the models have to be re-trained from scratch for each new sample to predict. Each evaluation step requires as many model trainings as there are samples in the test set. The search of optimal hyper-parameters, that is based on the evaluation of numerous combinations, becomes too costly. We decided to evaluate the performance of a combination on the basic forecasts, without recalibration.

3.3.6 Shapley and SHAP Values

With the help of a domain-knowledge expert, it's possible to explain a Day-Ahead price with respect to the market variables. However, EPF models compute their own relationships between features and prediction, making the latter hard to understand for opaque models. In this objective, Shapley values [Shapley, 1953] were defined within the framework of game theory in order to fairly distribute a gain among several players in a cooperative game. Players are not only paid for what they are able to gain when they are alone, but also for their contribution to the group when interacting with others. The gain to be distributed is here the deviation between a forecast and the average forecasted value $\widehat{Y}^{(d,h)} - \mathbb{E}(\widehat{Y})$. To detail the Shapley value computation, we introduce the following notations : let $\mathbb{F} = \{1, \cdots n_c\}$ be the set of all features in the input dataset X and $S \subseteq \mathbb{F}$ be a coalition of features, and ϕ_i be the Shapley value associated with the feature i. We note the model m as a function that maps a data instance $x = X^{(d,h)}$ to a prediction $\widehat{Y}^{(d,h)}$, and $\widetilde{m}_{X^{(d,h)}}(S)$ the marginalised prediction function over the coalition S. The latter is the expected prediction when using the values of features present in the coalition S and default values for features not in S :

$$\widetilde{m}_x(S) = \int m(x) d\mathbb{P}_{x \notin S} - \mathbb{E}[m(X)]$$

Then, it's possible to compute for each coalition S in which feature i does not appear, the difference in gain:

$$\widetilde{m}_x(S \cup \{i\}) - \widetilde{m}_x(S)$$

This gives a value of the impact of the collaboration of feature i with the set S. The sign of this differences indicates if the feature contributes toward increasing or decreasing the forecast in general. The last step is to sum and weight this value for all possible coalitions:

$$\phi_i(x) = \sum_{S \subseteq \mathbb{F} \setminus \{i\}} \frac{|S|!(n_c - |S| - 1)!}{n_c!} \left[\widetilde{m}_x(S \cup \{i\}) - \widetilde{m}_x(S)\right]$$

The calculation of a Shapley coefficient poses two difficulties: estimating the conditional expectations and dealing with the combinatorial explosion of the number of coalitions to go through, when the number n_c of features increases. The number of coalitions to be covered is exponential, in 2^{n_c}. [Lundberg and Lee, 2017] introduces the concept of Shapley kernel to approximate Shapley values and makes it possible the use of this approach on real-world dataset such as EPF ones. The method SHAP (SHapley Additive exPlanations) uses the Shapley values to compute an additive explanatory model g_x that is a linear combination of Shapley values for each possible coalitions:

$$g_x(z) = \mathbb{E}(m(x)) + \sum_{i=1}^{n_c} \phi_i z_i$$

with $\mathbb{E}(m(x))$ the average output of the model while being fed x, ϕ_i the explained effect of feature i and $z \in \{0,1\}^{n_c}$ a binary encoding of a coalition. This explanatory model is trained to be roughly equal to m in the vicinity of x. After wane to training n_d linear regression models. We use python's SHAP[6] package to compute the SHAP values of our models, using a total of 2500 coalitions per forecasts. We analyze the results $\Phi^{(d,h)} \in \mathbb{R}^{n_f}$ for each (d,h) of the test set in Section3.4.2.

3.4 Evaluation of the models on the different datasets

The objective of this section is to evaluate the different models of machine learning. First, we measure the impact of considering the additional features on the accuracy of predictions. We also evaluate the interest of simultaneously predicting the price of electricity in several countries. Then, we propose to study the models from an XAI point of view, to identify on which variables the predictions are based upon. Results are available in our repository[7].

3.4.1 How well do the models perform?

3.4.1.1 Period T1

We present the performance measures of the different models in Table 3.4. We compare the models to each other and evaluate the impact of adding features on the predictions. To do a fair comparison with [Lago et al., 2021], we consider the T_1 time period that was used in this paper. For a better interpretability of the multi-market models, we compute the metrics market by market.

First, we can observe that using additional features to predict prices always increases performance up to 15% gain. We support this finding by highlighting the p-values of Diebold & Mariano tests between models trained on SOTA datasets and their counterparts trained on enriched datasets in Table 3.5 (column A). We can observe that this difference is statistically significant for the vast majority of countries and models (values in bold lower than 0.05).

In the single-country framework, the Belgian market is the most difficult to tackle. The *RMAE* in Table 3.4 indicates that the best model for that country only achieves a fraction of 0.7 of the error of a naive forecaster. The other datasets have an *RMAE* lower than 0.6, or even 0.45 on German dataset. We believe this is due to the fact that Belgian consumption and production forecasts are not present in the SOTA datasets for period T_1. This is further discussed in Section 3.4.2.

Another conclusion from these experiments is that Random Forest models are not competitive enough for EPF. Their metric values are always significantly worse than those of the other models for all markets. Adding features increases performances, but this is not necessarily enough to outclass other models based on SOTA datasets. It also appears that *CNN* models are not state-of-the-art forecasting models for EPF. Even though they obtain relevant metric values on the enriched datasets, they don't significantly outperform the *DNN* or *SVR* models on any markets. We believe that the data provided to *CNN* models is not suitable for

[6] `https://shap.readthedocs.io/en/latest/index.html`

[7] see repository `https://www.dropbox.com/sh/2n7qje9dmhixh35/AADffdnjmJXRQEdvxbcBECgma?dl=0`

SOTA datasets														
Markets	Metrics	LEAR				DNN				CNN	DNN	RF	SVR	
		56	84	1092	1456	1	2	3	4				Chain	Multi
FR	smape	13.32	13.41	13.57	14.59	12.00	11.65	11.75	11.51	12.05	11.57	13.42	**11.23**	11.26
	mae	4.63	4.58	4.35	4.48	4.34	4.15	4.17	4.12	4.27	4.15	4.73	**4.03**	4.06
	dae	3.37	3.34	3.25	3.38	3.35	3.13	3.09	3.09	3.14	3.05	3.46	**3.02**	**3.02**
	rmae	0.69	0.68	0.65	0.67	0.65	0.62	0.62	0.61	0.64	0.62	0.71	**0.60**	0.61
DE	smape	15.25	15.16	17.31	17.94	14.27	14.49	14.25	**14.20**	16.26	14.47	17.43	14.48	14.53
	mae	3.64	3.59	3.61	3.72	3.27	3.34	**3.22**	3.23	3.63	3.27	4.11	3.27	3.28
	dae	2.54	2.53	2.65	2.74	2.32	2.46	2.29	2.33	2.58	**2.27**	2.79	2.38	2.38
	rmae	0.50	0.49	0.50	0.51	0.45	0.46	**0.44**	**0.44**	0.50	0.45	0.57	0.45	0.45
BE	smape	17.02	17.32	17.20	17.75	15.77	14.79	15.77	15.17	15.20	14.59	15.37	14.50	**14.46**
	mae	7.28	7.32	6.68	6.73	6.84	6.37	6.76	6.50	6.43	**6.25**	6.55	6.41	**6.25**
	dae	5.18	5.20	4.84	4.91	5.15	4.67	5.00	4.75	**4.62**	4.50	4.79	4.77	**4.62**
	rmae	0.82	0.83	0.76	0.76	0.78	0.72	0.77	0.74	0.73	**0.71**	0.74	0.73	**0.71**

		Enriched datasets					Multi-market Models				
Markets	Metrics	CNN	DNN	RF	SVR		CNN	DNN	RF	SVR	
					Chain	Multi				Chain	Multi
FR	smape	10.80	11.12	11.81	**10.43**	10.56	11.07	10.95	12.44	10.66	10.67
	mae	3.79	3.89	4.11	**3.65**	3.67	3.92	3.85	4.30	3.71	3.71
	dae	2.67	2.71	2.90	2.57	**2.56**	2.81	2.78	2.99	2.61	2.57
	rmae	0.69	0.58	0.61	**0.54**	0.55	0.59	0.57	0.64	0.55	0.55
DE	smape	14.24	**13.56**	15.83	13.84	14.40	14.42	13.96	16.45	14.22	14.29
	mae	3.19	**3.12**	3.72	3.15	3.25	3.29	3.24	4.11	3.26	3.28
	dae	2.16	**2.12**	2.37	2.22	2.31	2.26	2.22	2.71	2.33	2.34
	rmae	0.44	**0.43**	0.51	**0.43**	0.45	0.45	0.45	0.57	0.45	0.45
BE	smape	14.12	14.82	15.22	14.35	14.28	13.8	13.60	15.50	13.50	**13.47**
	mae	6.14	6.33	6.50	6.11	6.14	6.01	5.87	6.66	**5.88**	5.90
	dae	4.51	4.69	4.85	4.58	4.54	4.28	4.20	4.98	**4.35**	4.29
	rmae	0.70	0.72	0.74	0.69	0.70	0.68	**0.67**	0.75	**0.67**	**0.67**

Table 3.4 – Metrics over the period T_1 for the SOTA datasets (TOP) and for the enriched datasets (BOTTOM). The multi-market models' metrics are reported market by market. It is clear that the enriched datasets outperform the SOTA datasets.

convolutions. *CNN* models are tailored for extracting meaningful patterns among raw features, such as basic geometric shapes on an image. We feed it with data such as production and generation forecasts which is a high-level representation of meteorological data. Also, our data is of size 32×24, which limits the usefulness of performing convolutions compared to usual *CNN* datasets.

Lastly, still considering Table 3.4 and the p-values in Table 3.5 (columns B and C), we see that the interest of jointly predicting the prices of several markets is mixed. The multi-market forecasting model reduces forecast quality by up to 5%. This reduction is significant on 4 of the 5 models in France and Germany (column C). However, it significantly increases the performance of 3 out of 5 models in Belgium (column B). Merging the three datasets did not add any crucial and previously unknown information to the French and German datasets. On the other hand, it allows the model to use Swiss prices to predict Belgian prices. We believe this explains the significant increase in Belgium's performance.

		A	B	C
Country	**Model**	$\mathbf{H_o}$: $\mathbf{m_{SOTA} > m_{enriched}}$	$\mathbf{H_o}$: $\mathbf{m_{enriched} > m_{multi}}$	$\mathbf{H_o}$: $\mathbf{m_{multi} > m_{enriched}}$
FR	**CNN**	**0**	1	**0**
	DNN	**0**	0.176	0.824
	RF	**0**	1	**0**
	ChainSVR	**0**	0.989	**0.011**
	MultiSVR	**0**	0.975	**0.025**
DE	**CNN**	**0**	1	**0**
	DNN	**0.001**	0.999	**0.001**
	RF	**0**	1	**0**
	ChainSVR	**0.003**	1	**0**
	MultiSVR	0.219	0.949	0.501
BE	**CNN**	**0**	1	**0**
	DNN	0.919	**0**	1
	RF	0.117	0.845	0.155
	ChainSVR	0.998	**0**	1
	MultiSVR	0.991	**0**	1

Table 3.5 – P-values of the Diebold & Mariano tests for the T_1 period. (A) the test compares models trained on SOTA datasets with the same trained on enriched datasets. The null hypothesis states the enriched dataset has lower metric values than SOTA dataset models. With a threshold $\alpha = 5\%$, models in bold are significantly better when trained on the enriched datasets. (B) compares the single country forecasting models with the multi-country ones. The null hypothesis states the multi-country forecasting models are better than single-country ones on enriched datasets (values in bold). (C) The null hypothesis states the single-country forecasting models are better than multi-country ones on enriched datasets (values in bold).

3.4.1.2 Period T2

We now study the robustness of these observations by considering the time period T_2. We present the metric values obtained for this period in Table 3.6.

First, the best absolute metrics, *MAE* & *DAE*, increased by almost a factor of two over the T_2 period. This is not surprising as price levels also drastically shifted on this period. Figure 2.5 illustrated this phenomenon in France in the preceding Chapter. However, the *RMAE* decreased from 0.55 to 0.46 for the French *MultiSVR*, from 0.44 to 0.42 for the German *DNN* and from 0.67 to 0.57 for the Belgian *MultiSVR*, which shows that the models are performing better against the baseline than for the previous period. The ML models can thus still be used when sudden changes occur in the market, if recalibration is performed.

Second, the availability of Belgian consumption and production forecasts made this market easier to predict than for the previous period. However, it is still the most difficult market to predict, probably because we aren't using Swiss prices, as this country does not border Belgium.

Third, the differences in performance between the models are greater over this period. We clearly identify that the *SVR* models are better on the French and Belgian datasets while the *DNN* is the best model on the German dataset. The DM test pvalues in Figure 3.5

Enriched Datasets						
Market	Metric	CNN	DNN	RF	SVR	
					Chain	Multi
FR	smape	19.75	15.97	17.33	**14.23**	**14.23**
	mae	10.40	7.96	9.41	6.86	**6.61**
	dae	7.65	5.70	7.06	5.10	**4.74**
	rmae	0.73	0.56	0.66	0.48	**0.46**
DE	smape	20.36	**18.79**	22.35	18.80	19.45
	mae	8.66	**7.66**	10.77	8.44	8.85
	dae	6.53	**5.13**	7.77	6.25	6.62
	rmae	0.47	**0.42**	0.58	0.46	0.48
BE	smape	24.85	21.65	21.60	**18.93**	19.17
	mae	14.18	11.86	12.30	**9.35**	9.51
	dae	10.09	9.37	9.68	**6.67**	6.78
	rmae	0.88	0.73	0.76	**0.58**	0.59

Table 3.6 – Performance metrics over period T_2.

confirm that this difference is significant. In this figure, colored squares at coordinates (i, j) indicates that the forecasts of model i are significantly more accurate than forecasts of model j. We clearly identify green columns for the *SVR* models in France and Belgium, indicating that the *MultiSVR* and *ChainSVR* significantly outperform other models. For these countries, the *DNN* model outperforms the *RF* and *CNN* models. Finally, the *CNN* model is significantly worse than all the other models. For Germany, only the *DNN* significantly outperforms all other models, while the *RF* model is significantly outperformed by all other models. Although the results were more mitigated for period T_1, it is interesting to notice that the same models for the same markets stood out in terms of metrics (*SVR* for France and Belgium and *DNN* for Germany). We further investigate why those models are better on specific markets in Section3.4.2.

3.4.2 Forecast Explanations

The ML models' performance are satisfying : they successfully improve over the baselines (the *RMAE* is lower than 1) and outperform the *LEAR* model. However, to increase the confidence in the predictions given by then models, it is necessary to be able to explain them and to identify the most important characteristics in the decision-making process. This allows us to better appreciate their quality and better understand the phenomena involved in price prediction. We have seen that adding features dramatically improves model performance for the vast majority of datasets and models. We now aim to explain this performance bump by finding which features increase performances and when. We consider the SHAP value approach [Lundberg et al., 2020], a method that for each sample, assigns each feature a value that reflects its importance in the prediction process. $\Phi_c^{(d,h)}$ designates a SHAP value and denotes the contribution of a column c to the predicted hour h on day d. Note that a column $c = (f, l, h')$ refers to the hour h' of a feature f with l days lag. We also divide the contribution

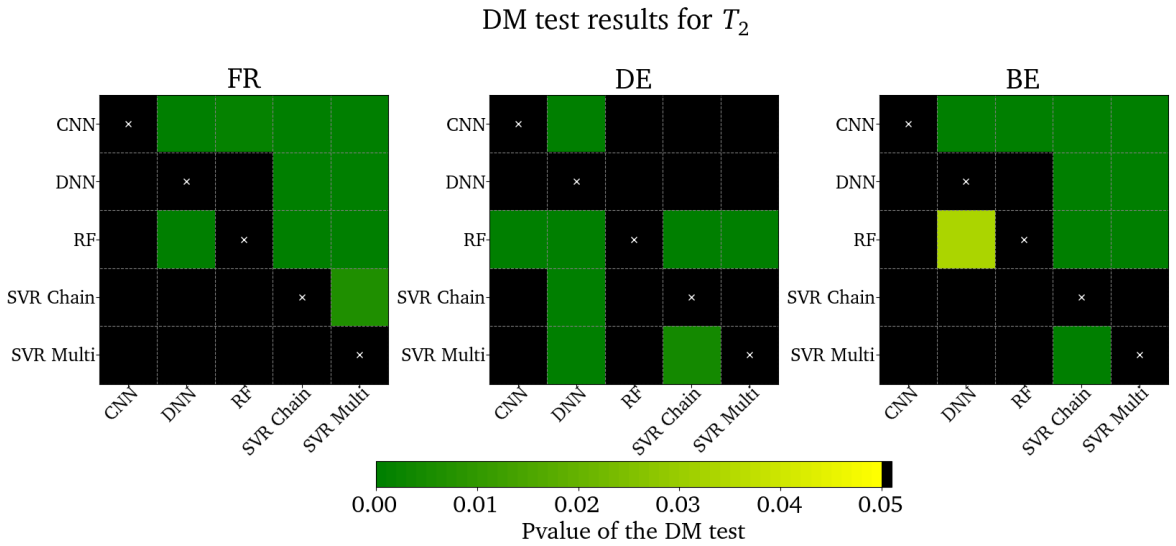


Figure 3.5 – P-values of the Diebold & Mariano tests computed on the recalibrated forecasts on period T_2. Colored squares in (i, j) indicates that the forecasts of model i are significantly more accurate than forecasts of model j. Green columns indicate that the corresponding models are significantly better than every other. Black lines indicate that the model on the y-axis' forecasts are significantly worse than every other.

of each column c so that they sum to one for each output (d, h):

$$\sum_{c=1}^{n_c} \Phi_c^{(d,h)} = 1$$

3.4.2.1 Global Explanations

We first focus on explaining the performance gaps between models. Results for period T_2 on the German dataset are presented on Figures 3.6, 3.7, 3.8. Each subplot corresponds to a feature f. The x-axis displays all possible lags in hours for this feature, while the outputs are shown on the y-axis. For a feature f, we display $\bar{\Phi}^{(h)}_{f,l,h'}$ at coordinates $[x = (l, h'), y = h]$, the average contribution over all days :

$$\bar{\Phi}^{(h)}_{f,l,h'} = \frac{1}{n_d} \sum_d \Phi^{(d,h)}_{f,l,h'}$$

This is equivalent to the AR model weight plots of Figure 2.6 described in the precedent Chapter, only organised by groups of features for a better readability, as it counts how much does a specific column $c = (f, l, h')$ participates in the forecast of a specific price h on average over the test set. Next, we report the feature importance over all samples, for all predicted hours, grouped by feature f or lags l. The average contribution for each feature $\bar{\Phi}_f$ is reported as a percentage of the total contribution in Table 3.7 :

$$\bar{\Phi}_f = \frac{1}{24 * n_d} \sum_{d=1,h=1}^{n_d,24} \sum_{l=1,h'=1}^{n_l,24} \Phi^{(d,h)}_{f,l,h'}$$

The average contribution for each lag day $\bar{\Phi}_l$ is shown in Table 3.8:

$$\bar{\Phi}_l = \frac{1}{24 * n_d} \sum_{d=1,h=1}^{n_d,24} \sum_{f=1,h'=1}^{n_f,24} \Phi_{f,l,h'}^{(d,h)}$$

Lastly, we study the evolution of feature contribution over time. Particularly, we are interested in the effects of the three Covid lock-downs in France on the Daily Average Unit Contribution (DAUC) that we define below:

$$\bar{\Phi}_f^{(d)} = \frac{1}{24} \sum_{h=1}^{24} \frac{1}{24 * n_l} \sum_{l=1,h'=1}^{n_l,24} \Phi_{f,l,h'}^{(d,h)}$$

The DAUC $\bar{\Phi}_f^{(d)}$ gives a value on how much on average does a feature group f brings to the forecasts of each day. It differs from feature importance $\bar{\Phi}_f$ because the values are not averaged over the test period but are divided by the number of columns referring to the feature f. This allows to fairly compare the contribution of French price whose number of columns $n_c = 24 * n_l = 96$ with the contribution of Gas Prices that takes only 1 column for instance. Figure 3.9 displays these measures.

	Model	Consumption Forecast			Generation Forecast			Past Prices					Current prices	Local	Foreign	Date Indicators
		FR	DE	BE	FR	DE	BE	FR	DE	BE	NL	ES	CH			& Gaz Price
	CNN	7.11	7.59	5.13	6.65	5.45	22.18	19.38	5.09	3.45	4.59	3.37	8.13	33.14	64.97	1.89
	DNN	7.10	4.28	5.15	5.13	5.61	7.55	15.49	5.60	3.82	4.78	5.63	28.39	27.71	70.80	1.49
FR	**RF**	1.33	1.58	1.33	1.00	1.13	2.19	**15.49**	1.21	2.30	1.03	1.06	**70.09**	17.82	81.92	0.26
	SVR Chain	4.10	2.86	4.23	9.81	4.18	6.22	17.46	6.16	4.47	6.90	5.62	26.56	31.37	67.21	1.42
	SVR Multi	4.10	2.68	4.06	9.06	4.55	6.90	17.16	5.69	4.36	6.16	5.65	27.98	30.32	68.03	1.65
	CNN	4.39	6.44	6.03	7.11	18.13	6.23	5.06	22.64	3.90	8.24	-	9.46	47.21	50.42	2.37
	DNN	4.11	5.08	4.78	4.98	18.10	6.89	3.88	20.95	3.67	9.27	-	16.66	44.13	54.24	1.63
DE	**RF**	0.91	1.85	2.54	0.87	11.17	1.77	3.53	**32.84**	1.44	2.76	-	**39.97**	45.86	53.79	0.35
	SVR Chain	3.15	3.92	4.70	5.09	13.83	6.20	4.87	25.13	5.44	10.05	-	16.56	42.87	56.04	1.09
	SVR Multi	2.69	4.01	3.74	5.17	15.82	6.10	4.12	25.33	4.57	10.68	-	16.68	45.16	53.74	1.10
	CNN	3.82	4.84	5.90	5.55	6.48	23.46	10.93	7.83	20.35	8.31	-	-	49.71	47.76	2.53
	DNN	7.77	6.55	7.07	6.88	7.82	11.37	10.96	8.82	20.00	10.78	-	-	38.44	59.57	1.99
BE	**RF**	2.14	3.35	11.70	2.75	2.66	4.94	**33.21**	2.01	**30.47**	5.44	-	-	47.11	51.56	1.33
	SVR Chain	7.79	4.12	5.47	6.20	7.05	10.04	16.22	6.79	22.80	10.07	-	-	38.31	58.23	3.46
	SVR Multi	7.66	4.08	5.62	5.76	7.00	10.43	15.72	6.56	23.55	10.33	-	-	39.59	57.11	3.30

Table 3.7 – Summary of average contributions by feature $\bar{\Phi}_f$ over time period T_2. The contributions are summed for all targets, all times, and all lags for each category. The last two columns display the weight of the characteristics of foreign countries in the total contribution.

Figure 3.6 presents the feature contribution of the Random Forest model for the German dataset. We can observe that most of the feature contributions are close to 0 or are used uniformly to predict all hourly prices over 24, forming vertical lines of red squares ($\bar{\Phi}_{f,l,h'}^{(h)}$ is high $\forall\, h = 1:24$). We relate this observation to the way RF models are trained: the Multi-Ouptut Decision Tree algorithm chooses a division that satisfies the split criterion for all target variables. Therefore, at least on the higher nodes, the same characteristics are used to determine all the target features and their contributions are thus important. Moreover, we see in Table 3.7 that the majority of the contributions are made by the Swiss prices and by the country-specific prices (for French and German datasets). Finally, from Table 3.8, we observe that they barely use the feature with two, three or seven lag days. We believe that these three facts explain why RF models perform significantly worse than any other model on every market.

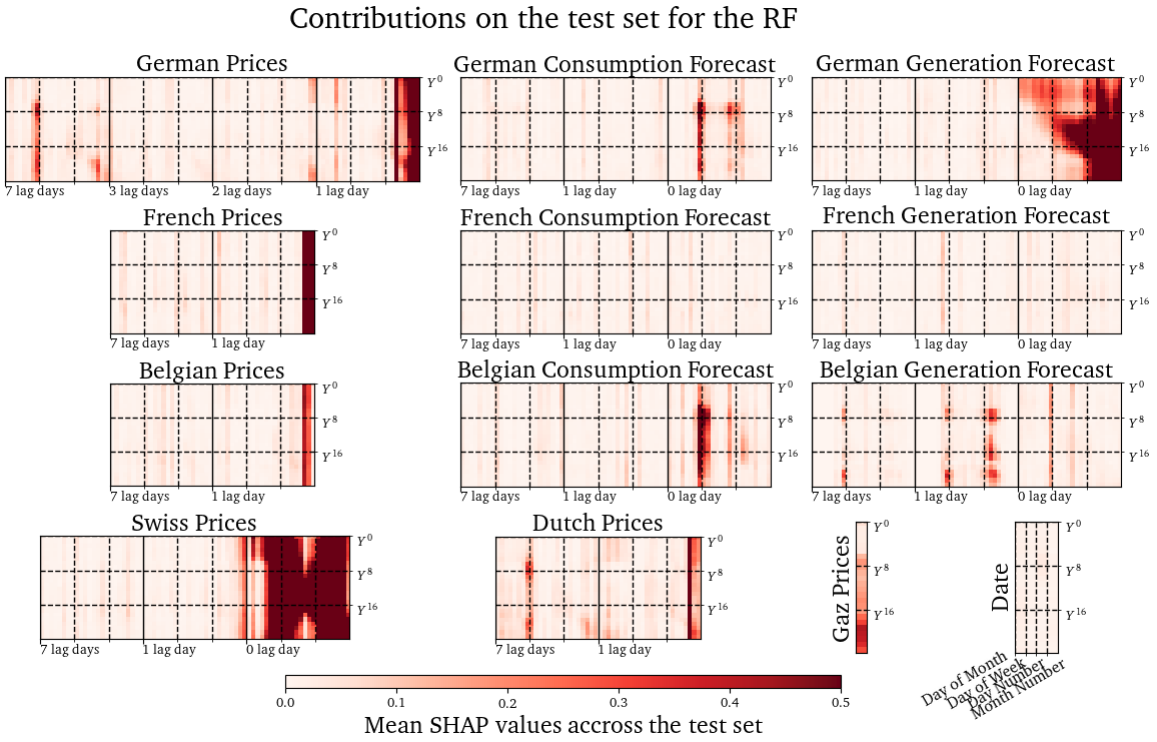


Figure 3.6 – Average feature contribution of the *RF* model for the German dataset. Each subplot displays the contributions of a single feature on all target variables. The target variables are on the y-axis, and the lag days and hours are represented on the x-axis. A red square on subplot f at the coordinates $[(l, h'), h]$ means that the contribution of feature f with l lag days at hour h' is high on average for the output h. We observe numerous contributions close to 0, meaning that some features have been omitted.

MultiSVR contributions, shown in Figure 3.7, display diagonals of red squares that occur when $\bar{\Phi}^{(h)}_{f,l,h'}$ is high $\forall h' = h$. This means that a column $c = (f, l, h')$ contributes to target variable h only if $h = h'$. This is most visible for the German generation forecast for the day to predict : market players usually try to take into account Renewable Generation for making their order books and it helps in estimating the prices.Patterns are hard to identify in the foreign features with lag days such as French consumption or generation forecasts, even though the contributions for these features are not null. We can observe partial diagonals in German, Dutch or Swiss price features. Due to market coupling, prices at a given hour from neighboring countries are sometimes identical, hence they constitute an important feature for prediction. Lastly, we notice strong contributions from evening prices with a 1-day lag, not only when $h = 24$ as observed by [Ziel, 2016]. The previous evening's prices are closest in time to the prices we aim to predict and are therefore an interesting feature well captured by the model.

The DNN model contributions in Figure 3.8 display centered group of red squares: high

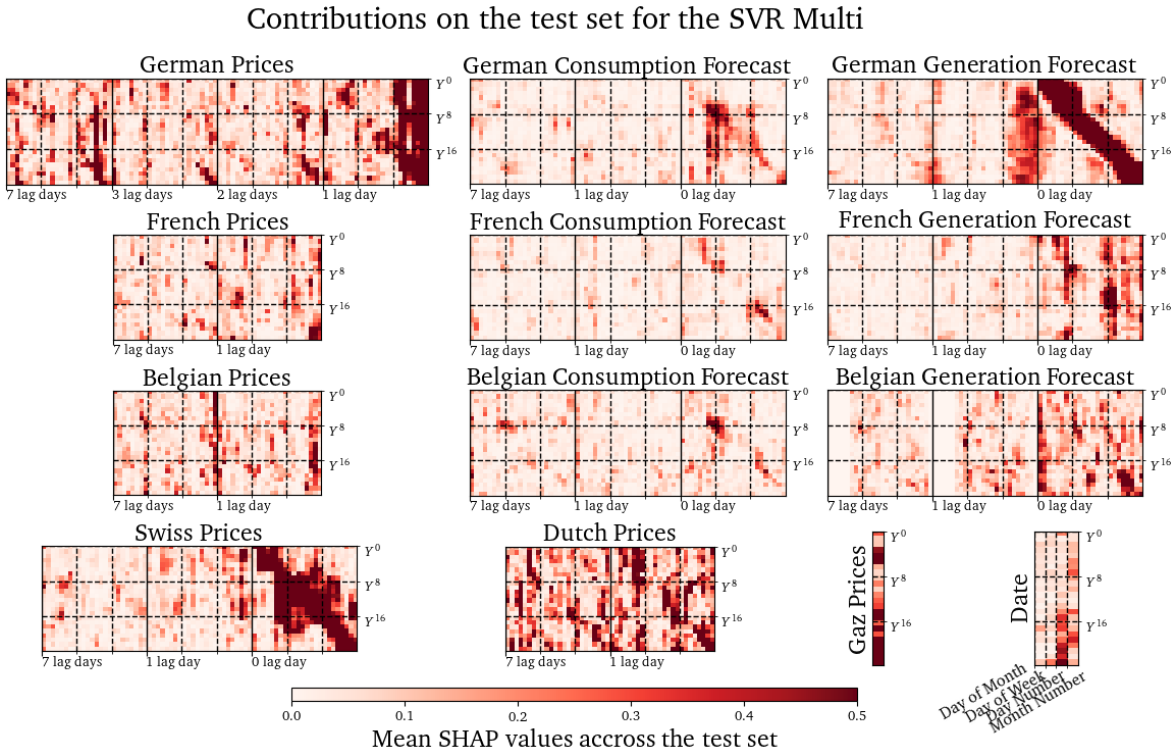


Figure 3.7 – Average feature contribution of the *MultiSVR* model for the German dataset. We observe diagonal lines of high contributions. This means that features at hour h' contribute mainly for predicting the output h if $h = h'$.

$\bar{\Phi}^{(h)}_{f,l,h'}$ for $h' = 8$ am to 8 pm and target variables $h = 8$ am to 8 pm. For instance, the German consumption forecasts from 8 am to 8 pm highly contributes to the German prices forecasts from 8 am to 8 pm. Peak-load specific orders can be issued by market players during those hours, and this is most used by power plant owners to allow them to either turn on their plant or shut it down during those 12 hours. We also identify diagonal patterns for several features such as the Generation forecasts or German and Dutch prices. The patterns observed on this model give a finer representation of its use of input features, and attest to its capacity to integrate complex phenomena. It also helps explaining the performance gap between all models. Similar figures for the *CNN*, *ChainSVR* and for other countries can be found in our repository.

We observe from Table 3.7 that both *SVR* and *DNN* models use foreign features for more than half of their total contribution (right-hand side columns). German renewable forecasts account for almost one fifth of the total contribution for predicting the German prices. This is the highest contribution after Swiss and German prices and it is almost twice the contribution of the French generation forecast for predicting French prices (or Belgian generation forecast for predicting the Belgian prices) that reach 10% at maximum. We explain this observation by the difference in energy mix between these countries. Nuclear electricity is produced according to the pricing algorithm, and is, in a sense, a result of the Day-Ahead price computation. Renewables are generated independently and thermal plants are dependant on the

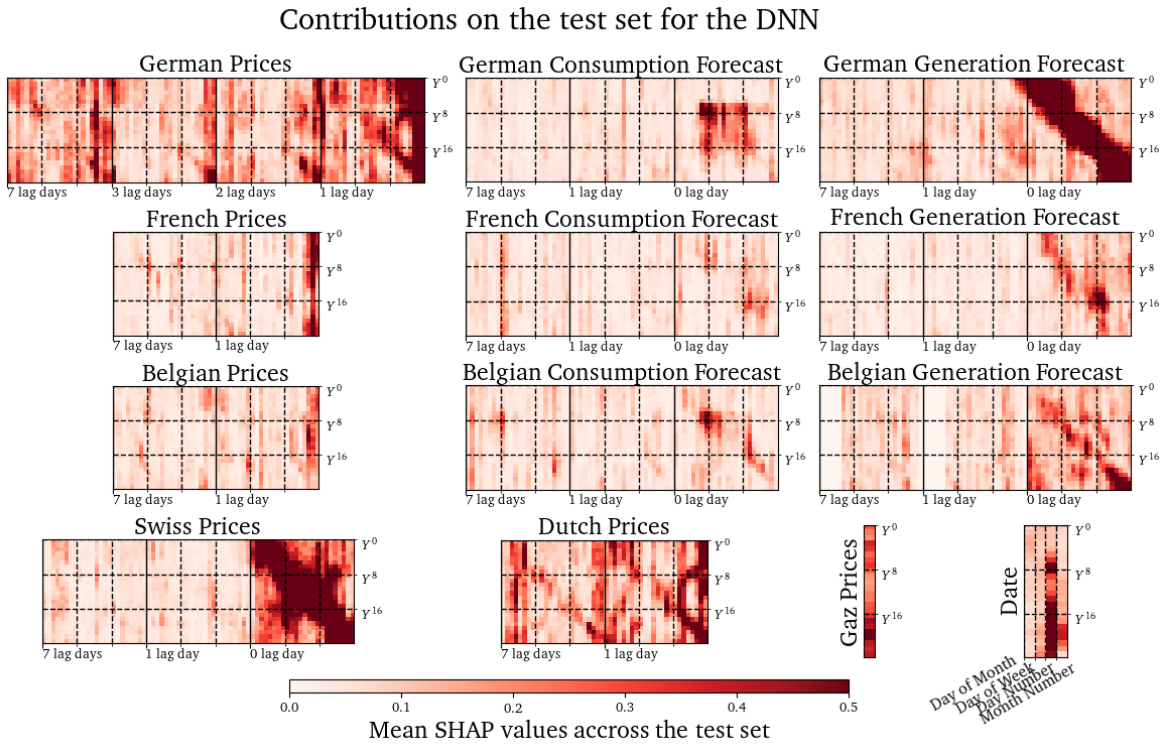


Figure 3.8 – Average feature contribution of the *DNN* model for the German dataset. We observe centered squares of high contributions between hours from 8 am to 8 pm for French consumption and generation forecasts. The Neural Network models display peak-load related patterns in their feature contributions.

price of other commodities. As a result, models have to put more weight to those features for the German market.

We also observe that the French consumption forecast contributes to predict French prices more than the German consumption forecast for German prices (and Belgian consumption forecast for the Belgian prices). It reflects the thermosensibility of the French market, that makes consumption more volatile and more determinant for setting the prices. Our studied models consider that this feature is more decisive in setting prices and gives it more weight.

In addition, it is clear form Table 3.7 that Swiss prices account for an important part of the feature contribution for all datasets that contain them. They are not part of the EUPHEMIA algorithm and can therefore be used by market participants to create their order books. Owners of power plants can use these prices to plan their production. Thanks to cross-border energy flows, market players can also exchange energy from and to this country using its price as a reference. Swiss prices thus constitute a good overview of the price level that will be reached in its neighboring countries.

Finally, from Table 3.8, we can conclude that features with two or three day lag contribute

	Model	D	D-1	D-2	D-3	D-7
	CNN	20.27	37.59	4.05	4.21	33.88
	DNN	46.71	29.83	2.41	2.32	18.73
FR	**RF**	73.81	21.05	0.30	0.34	4.49
	ChainSVR	42.34	31.42	2.72	2.63	20.89
	MultiSVR	44.87	30.74	2.85	2.12	19.42
	CNN	38.65	33.48	3.89	3.06	20.93
	DNN	43.53	30.01	3.50	2.92	20.04
DE	**RF**	54.12	40.13	0.49	0.39	4.87
	ChainSVR	36.28	37.52	2.92	2.92	20.36
	MultiSVR	38.62	37.12	2.55	2.48	19.24
	CNN	19.89	40.91	4.21	5.18	29.80
	DNN	28.39	39.72	3.99	3.89	24.01
BE	**RF**	21.55	64.20	1.25	1.31	11.69
	ChainSVR	29.55	43.65	3.90	3.27	19.63
	MultiSVR	29.61	43.72	3.93	3.30	19.44

Table 3.8 – Summary of average contributions per lag $\bar{\Phi}_l$ across all datasets for the first period T_2. Contributions are summed for all targets, all times, and features for each lag.

very little to the decision. They never contribute more than 5% of the total contribution. However, features with one or seven days lag are an important part of the model decision process. Different seasonalities can be observed in electricity prices. Among them, the weekly seasonality is one of the most important: the prices of the week before are generally more similar to the prices of the previous 2 or 3 days during then week-ends. In addition, prices are fixed daily by the EUPHEMIA algorithm, that uses as input the order books specific to the corresponding day. Thus, the data of the previous days, although similar to the current data, are not decisive in setting the prices. Contribution weights assigned to features with two and three day lags confirm that our models are also mostly based on current data.

3.4.2.2 Individual explanations

Lastly, we focus on the Daily Average Unit Contribution (DAUC) of the features and their evolution along time period T_2. Figure 3.9 displays these values for the best performing model for the French country: the *MultiSVR*. Grey areas delimit the periods of confinement. We have displayed the *SMAPE* values in red, and we first observe its evolution during the three lock-downs.

The model hardly adapts to the first lockdown (March 17, 2020 - May 11, 2020) and the highest error is reached in the middle of it. The second and third lockdown (November 2020 and April 2021) show no significant increase in errors overall. We explain this evolution by two factors: 1) The first confinement was more brutal for the French market. Because the industry had come to a standstill, prices and consumption had fallen. This is not the case for the other two lock-downs. 2) The first lockdown was a completely new situation for the model, which was not the case for the following. This also explains why the *SMAPE* starts dropping after half the first lockdown: the model has integrated several data samples from the confinement period and is able to adapt. The next two confinements are also more easily dealt with.

Next, we focus on the period following the first lockdown (from May 11, 2020 to October 1, 2020). This period was characterized by a slow French industry recovery and warm temperatures with consumption still below the standards. The nuclear production fleet was sufficient to cover the consumption and no gas-fired plant was needed. We see that the model correctly balances the trade-off between French production forecasts and gas prices by lowering the importance of Swiss and gas prices for the benefit of production forecasts and date indicators.

The period following the third lockdown (May 2, 2021 - December 31, 2021) is opposite to the previous one. High volatility in gas prices, due partly to the economic recovery in China, causes extreme volatility in electricity prices (see Figure 2.5). In this context, we observe the evolution of the contributions of Swiss and gas prices. These two features are the two most important characteristics and their movements are opposite: when one decreases, the other increases. Periods when the gas price contribution increases and the Swiss price contribution decreases are marked by high *SMAPE*, and oppositely. The model captures the importance of the gas price as shown by the strong increase in the contribution, but this price is so volatile that using gas price from two days ago results in a high error. A good way to avoid error spikes over this period would be to find a more reliable value than the EGSI index that is an estimate of price of gas two days before. Using the last value traded on the gas market could be an alternative.

Lastly, we notice little variation in contribution among the other features. The French consumption forecast contributed more from January to March 2020 and from November 2020 to April 2021. During these winter periods, the model gave more weight to the consumption forecasts to account for thermosensibility. This is not the case during the winter of 2021. Very low French nuclear production, due to the maintenance of the power stations, obliges the gas park to ensure the balance between consumption and production. The proportion of gas-generated electricity in the total mix was so high that small variations in consumption did not affect prices because all gas-fired power plants have the same marginal cost: the price of gas.

3.5 Synthesis, Discussion and Future Work

In this section, we summarize our conclusions and observations from the results of our experiments. First, we see that including new features in the predictive dataset dramatically increases model performance. Among these added features, the most discriminating are the features without lag days: production and consumption forecasts, and Swiss prices. We believe that the Belgian dataset is more difficult to grasp as it lacks the forecasts for the period T_1 and the Swiss prices for the period T_2. We also observe that the feature contributions depend on the considered dataset. These differences reflect the specificities of the European market such as the temperature sensitivity of consumption in France, or the intermittency of production in Germany.

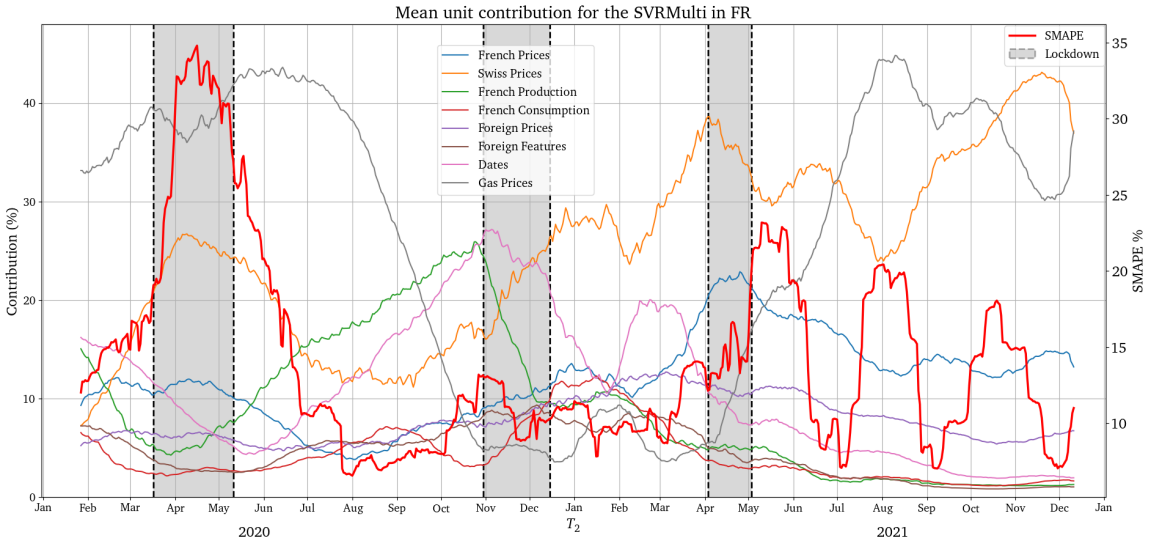


Figure 3.9 – Evolution of the daily mean unit contribution of each feature $\bar{\Phi}_f^{(d)}$ for the prediction of French prices over the period T_2 in the *DNN* model. The daily *SMAPE* is displayed in red.

Second, we report significant inequalities in the performance of ML models. *RF* and *CNN* are not suitable for the EPF paradigm we study. These models incorrectly incorporate input features and therefore we cannot identify significant patterns in their contribution analysis. In contrast, *DNN* and *SVR* extract meaningful information from features and display diagonal and peak load patterns in their contributions. As a result, these models are better over the three considered markets and the two time periods. Further analysis of the contribution revealed that they are able to react to significant market changes by updating the weight of discriminating features such as gas price when necessary. Although this adaptation is not instantaneous and a short period of performance deterioration is observed, the models produce accurate predictions in new situations. For example, performance has increased during the second semester of 2021 for the French market even though prices are more volatile.

Due to the high computation times and the difficulty of acquiring new data, several experiments were left for future work. The integration of new EPF features such as coal, oil or carbon prices, or the use of more data from foreign countries such as Spain, Italy, Austria or Denmark could be considered as future work. Given the importance of Swiss and gas prices in the total contribution, it will also be interesting to include other prices without lag days available before the close of the EPEX market, such as EXAA prices or UK prices. Moreover, the available transfer capacities are essential to understand the cross-border energy flows that are necessary to explain the price differences between countries. Their inclusion in our datasets should increase the accuracy of the multi-country forecasting framework.

Finally, we observed a slight degradation in performance while forecasting multiple countries at once. The tabular data structure that we use for feeding our models is not suited for managing the European network topology. Additionally, the *SVR* models scale badly

with the number of markets n_z to tackle since one has to train 1 model per output, thus $24 * n_z$ models in total. An appropriate model for pan-European forecasting is modeling the data as a Graph and using a Graph Neural Network model.

Chapter 4

Forecasting Electricity Prices: an Optimize then Predict-based approach

In the last Chapter, we assessed several Data-Driven models for EPF and explained their representation of the different markets using Shap values. We have concluded that while foreign features played an important role in price prediction, the Data-Driven models could not be directly employed for multi-market forecasts. In this Chapter, we focus on solving the pan-European price forecast problem and focus on the topological aspect of the network. We start by considering the flow-estimation problem where energy exchanges between markets are modeled with an optimization problem. Then, we represent the European network using a Graph where each node is a market whose prices have to be predicted. Estimated flows are used as edge attributes. We show that Graph Neural Networks successfully tackles forecasting prices for all considered markets, and that flow estimates are beneficial for EPF, especially for markets at the center of the Graph.

4.1 Introduction

Being able to forecast day-ahead electricity prices is crucial to control its production and ensure a successful energy transition. In the preceding Chapter 3, we have sought to produce the most accurate price prediction models possible. We also aimed to predicting the prices of different zones jointly. Although we did not obtain a significant improvement in the forecasts, the analysis of the contributions of the variables highlighted the importance of integrating data from foreign countries for the price forecast. For example, we have shown that Swiss prices contribute significantly to increase the accuracy of French, Belgian and German price forecasts. We concluded that we had not used enough information to correctly model the European network, in particular that we had not sufficiently taken into account transfer capacities and cross-border energy flows in our models.

We propose to overcome these limitations by putting forward different ways to integrate cross-border flows into predictive models. Flows $F_{z,z'}$ between two zones z and z' are constrained by the Available Transfer Capacity (ATC) Γ between two countries that share a border : $F_{z,z'} \leqslant \Gamma_{z,z'}$. However, this maximum capacity is not always fully utilized and knowing the flows between countries would undoubtedly improve the prediction models. Also, in the last Chapter, we limited our study to 3 of the most popular markets (France, Germany and Belgium). The price-fixing algorithm EUPHEMIA computes prices jointly for more than 40 zones across Europe (see Figure 2.7 of the first Chapter).

4.1.1 Contributions

We propose to take advantage of domain knowledge to estimate cross-border flows by a combinatorial optimization model. Using the definition of EUPHEMIA, we design two distinct optimization problems. One approach consists in a constrained linear problem (LP), the second is a quadratic problem (QP). We also investigate two means of combining the flow estimates to improve performances.

We then incorporate the flow estimates as input features of a ML model. Our research hypothesis is that we can improve the model predictions by enriching the input data thanks to domain knowledge. This approach is reversed from the predict-then-optimize approaches presented in Section 2.4 used to solve many decision-making problems by combining Machine Learning and combinatorial optimization. In this framework, some parameters of a combinatorial optimization problem are estimated from other features based on historical data. Our approach uses a combinatorial optimization model to estimate features that are then used to train a Machine Learning model.

Lastly, from the entsoe platform [1], we collect data from 35 markets and represent the European network as a Graph, where each market is a node and each connection an edge. Using appropriate models (*DNN*, *CNN*, *GNN*), we forecast the prices of all markets altogether.

[1] https://transparency.entsoe.eu/

4.1.2 Chapter Structure

In the first Section 4.2 of this Chapter, we detail how EUPHEMIA works and its variables and notations. Then, we introduce the flow estimation problem in Section 4.3. We explain our formalization using linear and quadratic programming. We also cover our combination methods. We use the results of these optimization problems in a multi-market forecasting model that predicts prices for 35 distinct zones of the European market (Section 4.4). The experimental evaluation (Section 4.5) confirms that the cross-border flows estimation makes it possible to improve the model performance. We then conclude with a broader discussion and a forward look (Section 4.6).

4.2 Preliminaries

We start this Chapter by presenting the variables of interest and their notations. We also briefly present the EUPHEMIA algorithm.

4.2.1 Variables & Notations

To solve the EPF problem, we consider the European market on day d where each zone has day-ahead prices $Y_z^{(d,h)} \in \mathbb{R}$ to be predicted for all $h \in [1, 24]$. For simplifying the notations, we drop the temporal indicator (d, h) on only write Y_z. The other features that interest us are:

- The Renewable generation forecast for the next day $R_z \in \mathbb{R}$
- The Programmable generation forecast [2] for the next day $G_z \in \mathbb{R}$
- The maximum generation capacity for the next day $V_z \in \mathbb{R}$ is the sum of capacities of all plants.
- The Price of the current day $P_z \in \mathbb{R}$. When forecasting the next day's prices, the current day's prices are available.
- The Available Transfer Capacities for the next day $\Gamma_{z,z'} \in \mathbb{R}$ which is the maximum amount of energy that can be sent from z to z'

Additionally, we consider the required amount of energy to be produced using controllable plants $E_z \in \mathbb{R}$, and the day-ahead flows $F_{z,z'} \in \mathbb{R}$. Hence, C, R, G, P, V and Γ are known at prediction time, while Y, E and F are unknown. In what follows, we propose to take advantage of knowledge from the field of electricity pricing to estimate the flows F between zones by combinatorial optimization, before using those results to forecast the day-ahead prices Y.

[2] Since renewable energy production is subject to external factors that are not controllable (wind speed, solar radiation, etc...), we distinguish the two types of source by R_z and G_z.

4.2.2 Euphemia

Before delving into the flow estimation problem, we quickly recall some key elements of the EUPHEMIA algorithm. More details are available in Chapter2.1.2. On the European market, this algorithm determines hourly prices by matching demand, production and exchanges across Europe in a way to maximize the social welfare while taking into account the market and network constraints. In this Chapter, we consider the following:

- The energy balance must be zero for all zones at all times, that is consumption, generation and cross-zonal flows balance:
$$G_z + R_z + \sum_{z'} F_{z',z} = C_z + \sum_{z'} F_{z,z'}$$
- The flow of energy between two zones must not exceed the maximum transfer capacity between these two zones:
$$F_{z',z} \leqslant \Gamma_{z',z}$$
- Where possible, the energy flow between two areas is maximized to generate more profit from congestion rents T.
$$\max_F \sum_{z,z'} F_{z,z'} T_{z,z'}$$

This algorithm runs daily at noon and determines the day-ahead prices Y_z, required energy production E_z and energy flows $F_{z,z'}$ of all the European zones, and it also satisfies the listed constraints. This leads to sophisticated and counter-intuitive flows between zones, some zones playing the role of transit zones to make possible energy exchanges between two other zones. For example, Switzerland is generally used to send energy from France to Germany or vice versa. Also, as the connected areas have multiple energy lines, bilateral flows can occur and one can have both $F_{z,z'} > 0$ and $F_{z',z} > 0$. To better model these dynamics, we use domain knowledge to approximate day-ahead flows $F_{z,z'}$ and use them as predictive variables into EPF models.

Limitations

The proposed optimization problems are simplified version of EUPHEMIA that we only use to estimate the cross-border flows. As seen in Chapter 2.1.2, EUPHEMIA is a complex MIQP algorithm that considers thousands of orders from several markets. In our approach, we only consider the total consumption and generation forecasts for a given period, rather than computing which orders are accepted by maximizing the *social welfare*. Moreover, we do not consider the Flow-Based Capacity network constraints and treat all connection lines using Available Transfer Capacities (ATCs).

4.3 Estimate cross-border flows by combinatorial optimization

In this Section, we present our two optimizations problems for estimating cross-border flows. We also describe two methods for combining flow estimates.

4.3.1 A straightforward approach with linear programming

The most natural way to formulate the flow optimization problem is to write the EUPHEMIA algorithm as a linear programming problem. In doing so, the network constraints are explicitly enforced, and the Day-Ahead Flow F is computed:

$$Fs = \arg\max_{F_{z,z'}} \sum_{z,z'} F_{z,z'} T_{z,z'} \tag{4.1}$$

$$\textit{under const.} \begin{cases} G_z + R_z - C_z + \sum_{z'} F_{z',z} - \sum_{z'} F_{z,z'} = 0 & \forall z \\ F_{z,z'} \leqslant \Gamma_{z,z'} & \forall z, z' \end{cases} \tag{4.2}$$

However, when we first tried to solve this problem on real data, we encountered an important problem with the Energy Balance constraint (4.2). For many data instances, the problem is not feasible because the flows are not enough to restore energy balance to zero. In the Appendix, Figure D.1 displays the distribution discrepancy of C_z and $G_z + R_z$ and it becomes clear that impossible cases can occur. To quantitize this phenomenon, we count the number of data instances where one of the two cases occur:

$$G_z + R_z > C_z + \sum_{z'} \Gamma_{z,z'} + \epsilon$$

$$G_z + R_z < C_z - \sum_{z'} \Gamma_{z',z} - \epsilon$$

for different values of ϵ. When $\epsilon = 0$, this corresponds to extreme cases where the difference $G_z + R_z - C_z$ can't be filled even fully using the full capacity of all available lines. When increasing ϵ, this gives the number of cases that could be solved while relaxing the Energy Balance constraint. We gather the results for ϵ up to 5 GWh and display them as percentages in Figure 4.1. First, we see that many important markets (France (FR), Belgium (BE), Great-Britain (GB), Spain (ES), Portugal (PT) and Germany (DE)) can't fulfill the Energy Balance condition because they produce too much (blue line). Oppositely, only a few markets can't cover their consumption (Greece (GR), Sicily (IT-SICI), Northern Italy (IT-NORTH)). Even though most market only have less than 5 GWh out-of-balance energy, the problem stays impossible to solve for many data instances.

4.3.2 Reformulating the problem

4.3.2.1 A formalization by linear programming

The impossibility to enforce Energy Balance comes from the Programmable Production Forecast G_z : it is too inaccurate and we would better consider it as a result of the price fixation algorithm. Indeed, EUPHEMIA determines which plant has to be turned on to match consumption (check Section2.1.2 for more details). A first approach to circumvent this

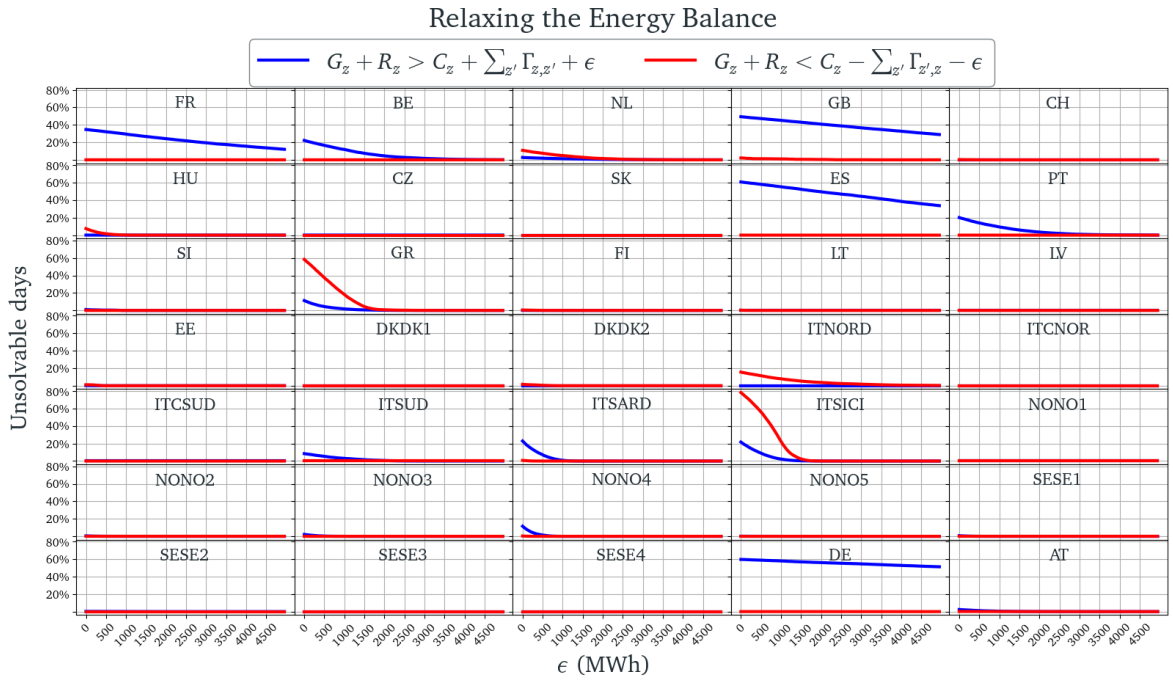


Figure 4.1 – Percentage of data instances on the period 2020-2021 where Energy Balance requirement can't be met, because (BLUE) the generation is too high (RED) the consumption is too low. This percentage usually goes down to 0 when allowing a 5 GWh gap.

problem is to use the required generation E_z as an optimization variable instead of G_z:

$$Flin = \underset{F_{z,z'},\, E_z}{\arg\max} \sum_{z,z'} F_{z,z'}(P_{z'} - P_z)$$

$$under\ const. \begin{cases} E_z + R_z - C_z + \sum_{z'} F_{z,z'} - \sum_{z'} F_{z',z} = 0 & \forall z \\ 0 \leqslant F_{z,z'} \leqslant A_{z,z'} & \forall z, z' \\ 0 \leqslant E_z \leqslant V_z & \forall z \end{cases}$$

This allows the generation to be determined to match $C_z - R_z$. E_z is only constrained by the maximal generation capacity of the market V_z, that is largely sufficient to cover consumption for all data instances. Also, R_z is never superior to C_z thus the problem is solvable for all data instances. As tariffs from the lines $T_{z,z'}$ are not available, we weight a flow $F_{z,z'}$ with the price difference between origin and destination market. This encourages flows from a zone with lower prices to a zone with higher prices that increases Social Welfare, and thus matches the logic of EUPHEMIA.

4.3.2.2 Formalizing the problem by a least-squares loss

The formulation of the problem by linear programming has a major drawback. It allow the generation of zone to expand to its maximum capacity $E_z \leqslant V_z$ without penalty to the cost. In practice, increasing the amount of generation has a cost that is not linear with respect to the generated volume (Figure 2.10 illustrates this phenomenon). We thus propose to rewrite the problem by transforming the energy balance constraint into a cost to be minimized :

Problem	*CC*	**MAE** (MWh	**SMAPE** (%)
Flin	0.153	944.63	120.32
Flsq	**0.389**	**418.05**	**105.93**

Table 4.1 – *CC*, *MAE* and *SMAPE* metrics between *Flin* and *Flsq* optimized flows and actual day-ahead flow values F on the train dataset.

$$Flsq = \underset{F_{z,z'}}{\arg\min} \sum_{z} \left(R_z + G_z - C_z + \sum_{z'} F_{z',z} - \sum_{z'} F_{z,z'} \right)^2$$
$$\textit{under constraint} \;\; 0 \leqslant F_{z,z'} \leqslant A_{z,z'} \;\forall_{z,z'}$$

The squared loss ensures that unbalanced zones are heavily penalized. By doing so, we can also remove the determination of E_z from the problem and we use the fixed programmable generation forecast G_z instead. This is equivalent to penalize the gap ϵ between $R_z + G_z$ and C_z presented in Section4.3.1 by a square loss.

4.3.3 Improving flow estimates

To study the quality of estimation of these two models, we solved the two optimization problems for each hour of our train dataset (see Section 4.4.1) using scipy[3]. As flow values are known a posteriori, we can evaluate the quality of the estimation on the train set using standard measures.

4.3.3.1 Combining the two formalizations

The metrics obtained are reported in Table 4.1. It is obvious that *Flsq* outperforms *Flin* on the dataset. However, by analyzing the estimations with a lower granularity, we observe that the performances vary according to graph edges. For example, the flow on the edge between Norway-5 and Norway-1 is well handled by problem *Flsq* as shown in Figure 4.2 (left) while the flow on edge between France and Germany is better handled by problem *Flin* (see Figure 4.2 right). To take advantage of these two models, we sought to identify the market conditions that allow us to differentiate these two scenarios. For this, we first define the loss difference between the results of the two problems as

$$L^{(t)}(z,z') = |F^{(t)}_{z,z'} - Flsq^{(t)}_{z,z'}| - |F^{(t)}_{z,z'} - Flin^{(t)}_{z,z'}|$$

where $t = (d,h)$ is one of the $24 * n_d$ possible time-steps. We analyze the relationship between $L^{(t)}(z,z')$ and the characteristics of the market $x \in \{C_z, C_{z'}, R_z, R_{z'}, P_z, P_{z'}\}$. We break down x into 100 quantiles x_q and compute the average loss for each (x,q):

$$L(z,z',x,q) = \frac{1}{24 * n_d} \sum_{t \in T(x,q)} L^{(t)}(z,z')$$

with $T(x,q) = \{t \mid x^{(t)} \in [x_q, x_{q+1}]\}$. Market conditions where $L(z,z',x,q) > 0$ correspond to situations where it is preferable to use *Flin* instead of *Flsq*. We name the results of this

[3] https://scipy.org/

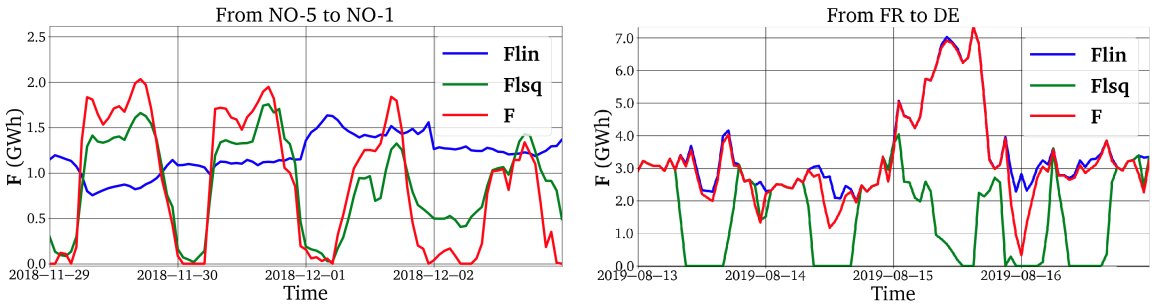


Figure 4.2 – Optimized flows *Flin* (blue), *Flsq* (green) and the actual Day-Ahead flows F (red) between the Norway-5 and Norway-1 zones (left) and the France and Germany zones (right). On the left, we observe that *Flsq* comes close enough to the Day-Ahead flow F, while *Flin* does not. On the right, we observe the opposite.

combination *Fcmb*. To generate *Fcmb* on the test dataset, we keep the same market conditions (z, z', x, q) as found on the train dataset. This prevents data leaks related to the use of posterior data for a prediction.

4.3.3.2 One-sided flows

In all the proposed formalizations, we enabled bilateral flows between two zones, i.e. $F_{z,z'} > 0$ and $F_{z',z} > 0$ can both occur, which matches the logic of EUPHEMIA. However, in practice, a majority of connections never have bilateral flows. Let's define

$$PF(z, z') = \{(d, h) | F^{(d,h)}_{z,z'} > 0\}$$

the set of instances where there is a positive flow between z and z'. The set

$$BF(z, z') = PF(z, z') \cap PF(z', z)$$

is the set of instances where there are bilateral flows. Figure 4.3 highlights the percentage of bilateral flows

$$\frac{100\%}{24 * n_d} |BF(z, z')|$$

for each connections. Only 17 out of 63 connections have a positive percentage. This mostly concerns central countries such as Germany (DE), Switzerland (CH) or Austria (AT) that act as transit zones for other markets. To further improve our flow modeling, we identify one-sided connections and apply one-sideness in our flow estimates.

$$OS(z, z') = PF(z, z') \backslash PF(z', z)$$

is the set of instances where the flows between z and z' is one-sided. If

$$\frac{100\%}{24 * n_d} |OS(z, z')| > 75\%$$

then we consider the edge (z, z') as always one-sided. For this, we keep the most important predicted flow from which we subtract the least important flow. We set the latter to 0. In this way, the energy balance in the two zones remains the same. We apply this transformation to *Fcmb* and call the result *Fos*.

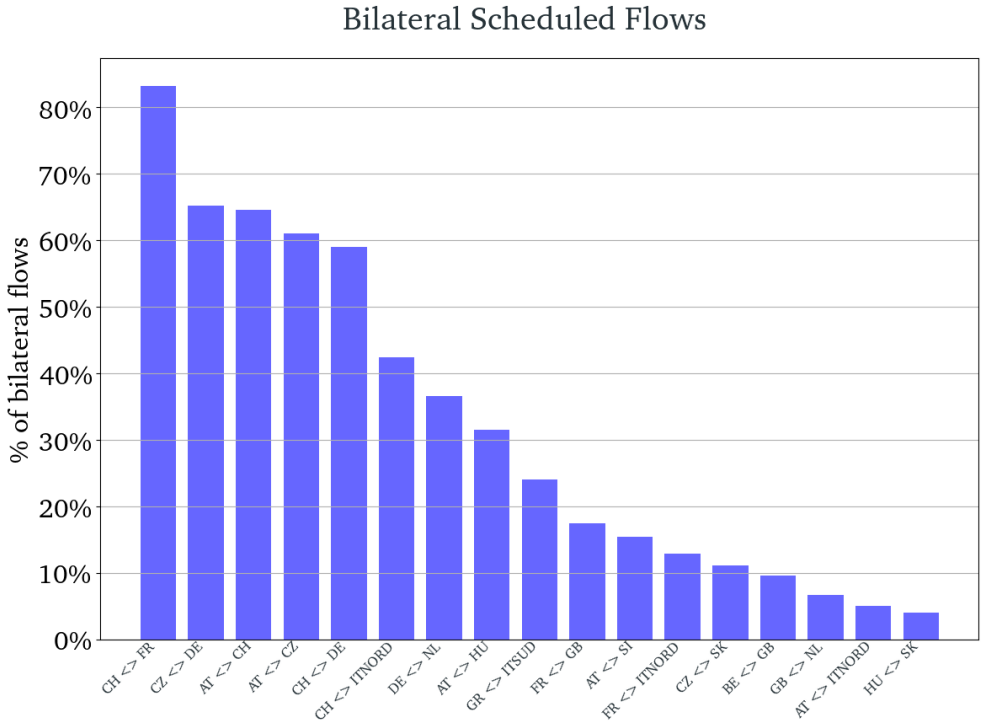


Figure 4.3 – Percentage of Bilateral flows over the 2020-2021 period.

4.4 Electricity price forecasting models

4.4.1 The dataset

In this section, we tackle the EPF problem on the European market. We collected the data for 35 markets, linked by 63 connections, summarized in Figure 4.4. For each zone z, the attributes are

$$X_z^{(d)} = (C_z^{(d)}, R_z^{(d)}, G_z^{(d)}, P_z^{(d)}) \in \mathbb{R}^{96}$$

Hence, each day is described by 35×96 predictive features and the targets to be predicted are the 24 hourly prices for each zone. We exclude the Swiss and Great-Britain prices from the prediction task. Although they are part of the network, their prices are determined prior to the closing of EUPHEMIA. Due to their important contributions in the forecasts computed in the previous Chapter 3.4.2, we prefer to use them as predictive variables. We predict the 24 prices of the remaining 33 zones : $Y^{(d)} \in \mathbb{R}^{792}$. Our dataset spans from 01/01/2016 to 31/12/2021. We use the last two years (2020, 2021) as test set. The year 2019 is kept as a validation set for hyper-parameter search.

In addition to the 35×96 predictive features cited above, we consider the Available Transfer Capacities (ATC) for each connection Γ or instead one of the flow estimates

$$\widetilde{F}^{(d)} \in \{\Gamma^{(d)}, Flin^{(d)}, Flsq^{(d)}, Fcmb^{(d)}, Fos^{(d)}\}$$

for each link, leading to 126×24 additional predictive variables. Each line of our dataset

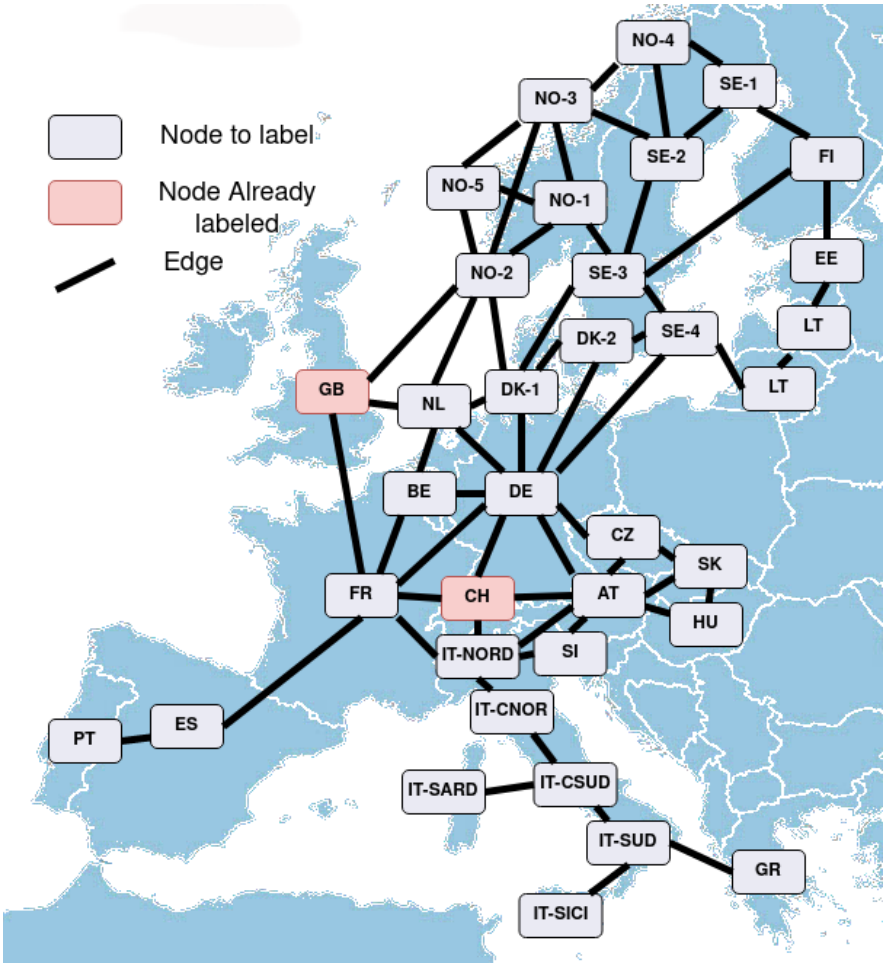


Figure 4.4 – The European market as a Graph. Each market is a node to label, except for the nodes highlighted in red that are already labeled because price fixation happens before noon. Each connection between market is a Graph Edge.

corresponds to a day and has 6384 values. Lastly, our data is re-arranged as a Graph where each zone z is a node with attributes $X_z^{(d)}$ and labels $Y_z^{(d)}$, and each connection is an edge with attribute the flow estimates or ATC, and labels the real Day-Ahead flows. A graphical summary is displayed in Figure 4.4

4.4.2 The machine learning models

4.4.2.1 Data-Driven approaches

We use Deep Neural Network and Convolutional Neural Network to predict the electricity prices. Deep Neural Networks (*DNN*) [Lago et al., 2018a,b, 2021; Mosbah and El-Hawary, 2016] are the most commonly used models in EPF. Their training samples are vectors $X^{(d)} \in \mathbb{R}^{6384}$. Convolutional Neural Networks (*CNN*) have also seen a growing interest in EPF over the past years [Cheng et al., 2020; Khan et al., 2020; Lago et al., 2018a]. We

compute the convolutions along time and each sample is avector $X^{(d)} \in \mathbb{R}^{(35+126)\times 24}$. We employ the same methodology for fitting those models as what was described in the precedent Chapters (Sections 2.2.1and 3.3.1). The dimensionality of the data to predict $Y^{(d)} \in \mathbb{R}^{792}$ discourages us to use the SVR models that require fitting 1 model per output.

4.4.2.2 Graph Neural Networks

Lastly, we propose to use a Graph Neural Network (*GNN*), which is new for the EPF domain. The *GNNs* have been proven capableof solving many complex tasks such as traffic forecasting problems, recommendation systems or biology [Wu et al., 2020] [Xu et al., 2018] because it can process data generated from non-Euclidean space. In the context of EPF, we want to exploit the European network's structural data : $\mathcal{G} = (\mathcal{V}, \mathcal{E}) = (X, \widetilde{F})$. We perform a Node Labeling task where graph convolution layers are stacked and update the node embeddings. This is followed by independent linear layers that map embeddings to their price prediction for each node to predict. We use TensorFlow and PyTorch-Geometric libraries [4] for training and evaluation of the *GNNs*.

4.4.2.3 Hyper-parameter search

Each model (*DNN*, *CNN*, *GNN*) is trained on 5 different versions of our dataset based on the method used to estimate F: Γ, *Flin*, *Flsq*, *Fcmb*, *Fos*. To ensure fairness in our experiments, we set a time limit for the hyper-parameter search. More precisely, we let our program explore the hyper-parameter grid for 24 hours for each model with $\widetilde{F} = \Gamma$ on a 20 cpus computer and use the same configuration for all variants of F. This introduces a slight bias, as the resulting best configuration is chosen for its performance on the Γ dataset. After finding the optimal configuration, we calculate forecasts on the test dataset using recalibration. It consists in re-training the model using the most recent data before making forecasts. Once a test set sample has been predicted, we can integrate its predictions into the training dataset and retrain the model. We recalibrate our models every 30 days.

4.5 Experiments

We compare the values of the predicted $\widehat{Y}_z$ and the real Y_z target variables for the different zones z, and the values between estimated flows $\widetilde{F}$ and real flows F for all connections (z, z'). We use standard measures *MAE*, *SMAPE* and *CC* (the average correlation coefficient over the target variables) that is defined as:

$$CC(y, \widehat{y}) = \frac{1}{n_z} \sum_z \frac{\sum_i (y_{i,z} - \bar{y}_z)(\widehat{y}_{i,z} - \bar{\widehat{y}}_z)}{\sqrt{\sum_i (y_{i,z} - \bar{y}_z)^2 \sum_i (\widehat{y}_{i,z} - \bar{\widehat{y}}_z)^2}}$$

With $\bar{y}_z = \frac{1}{24 * n_d} \sum_i y_{i,z}$ the average predicted value for zone z and $\bar{\widehat{y}}_z = \frac{1}{24 * n_d} \sum_i \widehat{y}_{i,z}$ the average prediction for zone z. For the flow estimation error, y_z is replaced by $\widetilde{F}_{z,z'}$. To

[4] `https://www.tensorflow.org/`, `https://pytorch-geometric.readthedocs.io`

Problem	CC	MAE (MWh)	SMAPE (%)
Γ	0.14	917.59	111.43
Flin	0.116	876.51	111.12
Flsq	**0.380**	388.95	105.35
Fcmb	0.367	396.46	102.52
Fos	0.375	**314.5**	**81.19**

Problem	Γ	Flin	Flsq	Fcmb	Fos
Γ	-	1.0	1.0	1.0	1.0
Flin	**0.0**	-	1.0	1.0	1.0
Flsq	**0.0**	**0.0**	-	1.0	1.0
Fcmb	**0.0**	**0.0**	**0.0**	-	1.0
Fos	**0.0**	**0.0**	**0.0**	**0.0**	-

Table 4.2 – (LEFT) Metrics for flow estimation on the test dataset for the different methods. The *Flsq* method outperforms the *Flin* methods. The *Fcmb* method does not improve the metrics, while the *Fos* method improves performances. (RIGHT) p-values of the DM test using the *SMAPE* loss. Values in bold on coordinate (i, j) inferior to 5% indicate that model i is significantly more performing than model j.

check the statistical significance of the results, we use the Diebold & Mariano (DM) test with threshold of 0.05 (see Section 3.3.2 for more details). We use *SMAPE* as *Loss* to better account for the different price scales. The search space and best configuration of the Hyper-Parameters are detailed in the Appendix D. For reproducibility purpose, the source code and the data are made available[5].

4.5.1 Results

4.5.1.1 Flow estimate

The results of the flow estimation problems on the test set are first presented in Table 4.2 (LEFT). We make the same observation as for the train set: *Flin* barely improves the quality of the flows while *Flsq* dramatically reduces the error. Then, their combination *Fcmb* does not show notable metric improvement while setting up one-sided flows *Fos* does. We perform DM tests and display the p-values in Table 4.2(RIGHT). They confirm that the flow estimate quality increases with the complexity of the estimation method, i.e. *Fos* outperforms every method, *Fcmb* outperforms every method except *Fos* and *Flsq* is better than *Flin*.

4.5.1.2 Price forecast

Global analysis

The results of the EPF problem on the test period are presented in Table 4.3. The left part displays the metrics, while the right part details the P-values of the DM tests. On each line, we first compare the model in that line with the same model using other flow estimates (first 5 columns), then we compare it to other models using the same flow estimate (last 3 columns). We can for instance confirm that the *DNN* model using the network constraints Γ is significantly more efficient than the *CNN* using Γ (first line) because the p-value is zero.

The *CNN* models are less competitive. They obtain the worst metrics and the DM test confirms that they are significantly less efficient than other models using the same flows (penultimate column). The *GNN* models are the most adequate models for this problem. Their metrics are better and the DM test statistically confirms that they outperform other

[5] `https://github.com/Leonardbcm/OPALE.git`

models using the same flows (last column). The *DNN* models thus stand in between. We now analyse the performance variations with respect to the flow estimation method.

Next, we compare results obtained using the network constraints Γ and those using estimation methods $\widetilde{F}$ (1st column of the RIGHT table). We notice that for the *DNN*, estimating the flows significantly improves performances. For the *CNN*, the performance increase is only significant for more sophisticated *Fcmb* and *Fos* methods, while for the *GNN*, only for *Fcmb*. Moreover, the *Flin* method is generally never significantly more efficient than other methods (2nd column of the RIGHT table). However, which flow estimation method is better for all models remains unclear: no method significantly outperform every other for a given model.

Analysis by markets

Figure 4.5 details the *SMAPE* value for each considered zone. It explains the performance gap of the *GNN* model compared to the *DNN* and *CNN*. In Scandinavia (Denmark (DK), Norway (NO), Sweden (SE)), the latter fail to achieve correct results. The *GNN* obtain a *SMAPE* that is lower than 35% for all markets and lower than those obtained by the *DNN* or *CNN*. Figure 4.6 highlights the significance of using a flow estimate in place of the ATC. We observe that replacing Γ by *Flsq*, *Fcmb* or *Fos* leads to overall improvements, even though local decreases can occur. Almost all zones profit from using $\widetilde{F}$ for the *DNN*. Using *Flin* improves performance less often than other methods and can degrade forecasts on multiple neighboring areas (Italy (IT) and a central block : Germany (DE), Slovenia (SI), Austria (AT), Hungary (HU), Czechia (CZ), Slovakia (SK) for the CNN). Using an inappropriate flow estimate with the GNN can degrade performance for a few markets (France (FR), Germany (DE), Slovenia (SI), Norway-5 (NO-5)). *Fcmb* shows the biggest improvements and the lowest decrease for all models.

4.5.2 SHAP Values

It is possible to further analyze our models and determine the impact of the different groups of features on the predictions. To that end, we consider the SHAP value approach [Lundberg and Lee, 2017], a feature attribution method that assigns to each feature a value that reflects its contribution in the prediction process. We denote the contribution of a column c to the hourly price h of a zone z on day d as $\Phi_{z,c}^{(d,h)}$. A column $c = (f, h', z')$ refers to the feature f at hour h' for zone z' or pair of zones (z', z'') if f is an edge attribute. Hence, the contribution tensor $\Phi \in \mathbb{R}^{731\times 792\times 6385}$ is made of 3.7 billion values. For computational reasons, we only compute 500 most important SHAP values on the first 30 days of the test dataset. We normalize the results so that the sum of each contribution equals 1 for each target of a given day (check Chapter 3.4.2 for a more detailed explanation). We are interested in the sum of the contributions of each feature f:

$$\Phi_f = \sum_{d,z,h,h',z'} \Phi_{z,f,h',z'}^{(d,h)}$$

Models	CC	MAE (€/MWh)	SMAPE (%)
DNN_Γ	0.893	13.97	29.76
DNN_Flin	0.903	13.44	28.51
DNN_Flsq	0.904	12.96	28.26
DNN_Fcmb	0.906	13.07	28.38
DNN_Fos	0.909	13.2	28.84
CNN_Γ	0.866	14.41	32.17
CNN_Flin	0.865	14.54	32.23
CNN_Flsq	0.875	14.19	32.01
CNN_Fcmb	0.867	14.04	31.81
CNN_Fos	0.872	14.26	31.87
GNN_Γ	0.925	10.23	24.59
GNN_Flin	0.926	10.22	24.6
GNN_Flsq	0.925	10.17	24.6
GNN_Fcmb	0.926	10.18	**24.46**
GNN_Fos	**0.926**	**10.14**	24.52

Γ	Flin	Flsq	Fcmb	Fos	DNN	CNN	GNN
-	1.0	1.0	1.0	1.0	-	**0.0**	1.0
0.0	-	0.995	0.887	**0.04**	-	**0.0**	1.0
0.0	**0.05**	-	0.06	**0.0**	-	**0.0**	1.0
0.0	0.113	0.94	-	**0.0**	-	**0.0**	1.0
0.0	0.996	1.0	1.0	-	-	**0.0**	1.0
-	0.35	0.847	0.977	0.964	1.0	-	1.0
0.65	-	0.904	0.99	0.976	1.0	-	1.0
0.153	0.096	-	0.942	0.855	1.0	-	1.0
0.023	**0.01**	0.058	-	0.33	1.0	-	1.0
0.036	**0.024**	0.145	0.67	-	1.0	-	1.0
-	0.431	0.457	0.989	0.882	**0.0**	**0.0**	-
0.569	-	0.504	0.999	0.945	**0.0**	**0.0**	-
0.543	0.496	-	0.999	0.953	**0.0**	**0.0**	-
0.011	**0.001**	**0.001**	-	0.089	**0.0**	**0.0**	-
0.118	0.055	**0.047**	0.91	-	**0.0**	**0.0**	-

Table 4.3 – (Left) Metrics on the test period. (Right) DM test P-values. For each trained model (line), the p-value is computed against the same model with other flows (first 5 columns) and against other models with the same flows (last 3 columns). The null hypothesis states that the column model outperforms the row model. With a threshold of 0.05, the bold values indicate that the row model outperforms the column model.

We compute Φ_f for each $f \in (C, G, R, P, \widetilde{F})$ and display them in Table 4.4. First, we observe that the *GNN*'s top contributing features are the prices that explain 30% of the forecasts, against approximately 20% for the other models. The *GNN* also uses $\widetilde{F}$ the least (14% against 18-20%). Next, we observe that the *DNN* model favors the use of $\widetilde{F}$ at the expense of C, G and R as we use more sophisticated flow estimate (*Flsq*, *Fcmb*, *Fos*). In contrast, the average contribution of $\widetilde{F}$ in the *CNN* and *GNN* does not show a clear trend. To detail this observation, we display in Figure 4.7 the difference of contribution between Γ and the used estimate $\widetilde{F}$. Green squares on coordinate (i, j) indicate that the contribution of $\widetilde{F}$ is more important than the contribution of Γ for predicting the zone i for model j. We observe that the *Flin* contribution differences are mostly negative; that is, models rely less on *Flin* than Γ for forecasting prices. Next, we see that the *DNN* increases the contribution of $\widetilde{F}$ for almost all zones. This is particularly marked for *Fos*. Lastly, the *GNN* always lowers the contribution of Latvia (LT), Estonia (EE), Lithuania (LT), Finland (FI) and Greece (GR). Those zones are characterized by having few (1 or 2) connections and being far from the center of the network.

4.5.3 Discussion

The joint analysis of the model's performances and SHAP values of *Flin* shows us significant degradation of the flow estimation and price forecasts, and less contribution for the forecast than Γ. This leads us to conclude that the *Flin* method is not a good flow estimation approach. Apart from *Flin*, other flow estimation methods are all beneficial for the EPF task, but can locally degrade performances for stronger models (GNN). Using *Fcmb* in a EPF model seems to be a reasonable default choice for any model.

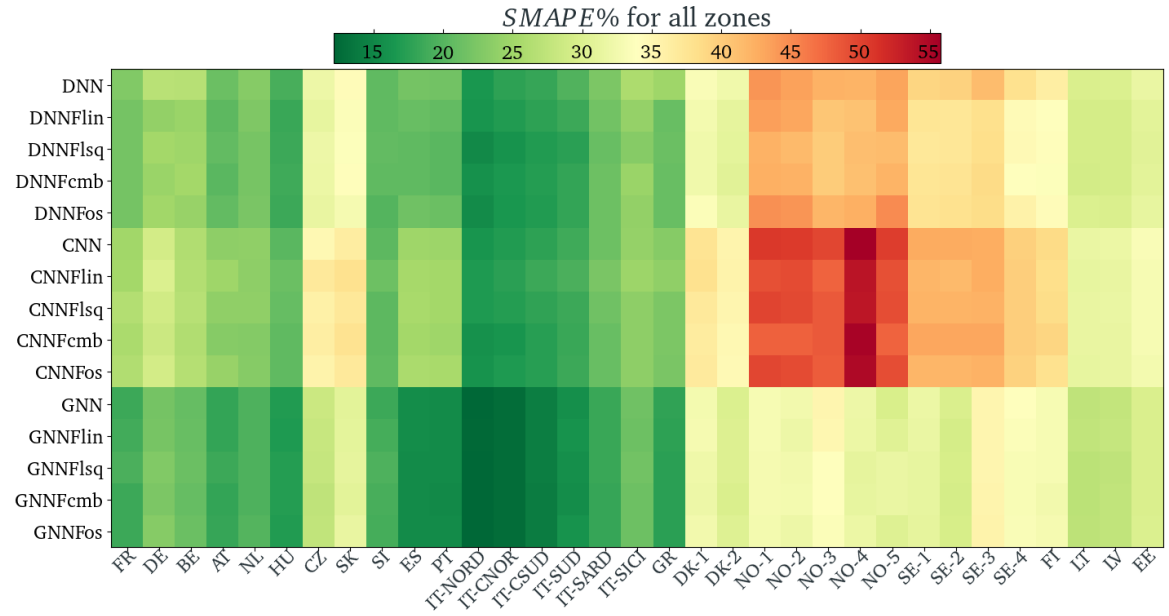


Figure 4.5 – *SMAPE* for all considered markets. While DNN and CNN fail to produce accurate results in Scandinavia, the GNN can handle it.

Model	DNN					CNN					GNN				
	Γ	Flin	Flsq	Fcmb	Fos	Γ	Flin	Flsq	Fcmb	Fos	Γ	Flin	Flsq	Fcmb	Fos
C	19.4	19.2	19.3	19.3	18.8	19.3	19.2	18.9	18.8	18.6	16.7	17.0	16.7	16.7	17.0
G	20.4	20.5	20.0	20.0	19.5	20.1	20.6	10.7	20.5	20.0	18.1	18.0	18.0	18.0	18.3
R	21.5	21.6	20.7	20.7	20.1	22.1	21.5	21.0	20.9	22.3	19.4	19.1	19.5	19.6	19.5
P	20.5	20.5	20.7	20.5	20.3	20.1	20.5	21.3	21.7	20.4	31.6	31.5	31.0	30.9	30.6
F	18.5	18.2	19.3	19.4	21.2	18.4	18.1	18.1	18.2	18.7	14.2	14.4	14.8	14.8	14.5

Table 4.4 – Average contribution (%) for the predictions grouped by feature. For the *DNN* and *CNN* models, we observe that the average contribution of the flows $\widetilde{F}$ increases as we use more sophisticated estimation methods.

The *DNN* is the leat sophisticated model and takes as input tabular data. However, observing both a significant performance improvement and an increase of the average contribution of any $\widetilde{F}$ over Γ for almost all zones, we infer that the *DNN* model takes benefit from using flow estimation methods. The inability of *DNN* to model the network is balanced by topology-rich estimated flows.

Next, the *CNN* uses a matrix of arbitrary-arranged input features and a convolution kernel and dilation rate inconsistent with the European network. Hence, *CNN* is the least tailored model for EPF and zones and flows are not properly associated. The incorporation of basic flow estimates (*Flin*, *Flsq*) in place of ATC can significantly deteriorate the price forecasts in well-connected central markets, misleading the model.

The *GNN* model is the best model for EPF at the European scale. The Performance gap

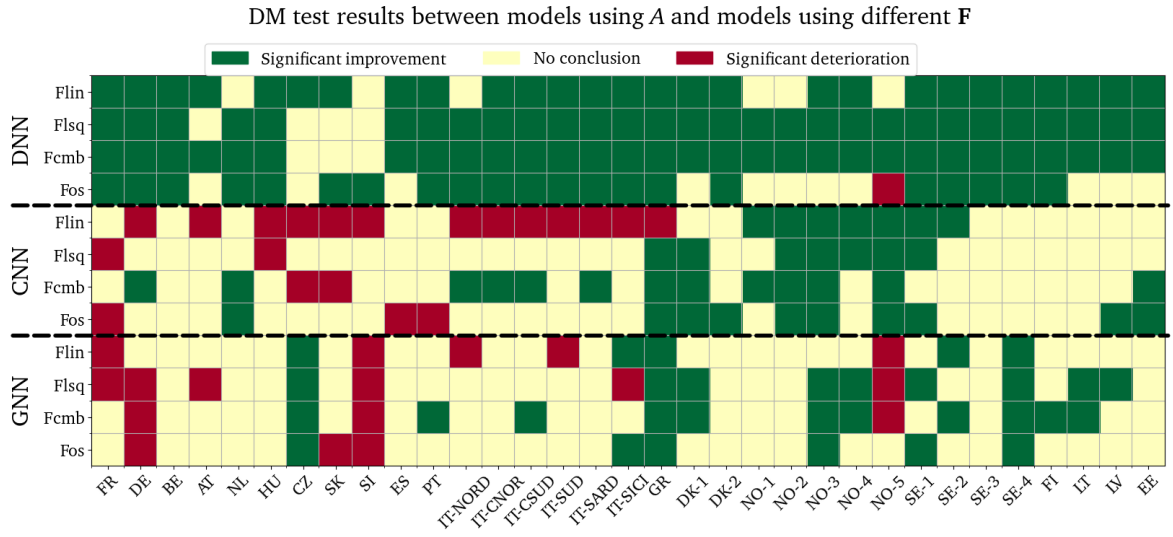


Figure 4.6 – DM test between models using the network constraints Γ and models using different flow estimates. A green square indicates that using estimated flows significantly improved performance, while red squares indicate a significant deterioration in performance. Yellow squares indicate that no conclusion can be reached.

is even more visible in the Scandinavian markets. Due to the separation of those countries in several zones, the Scandinavian area consist in smaller markets. Moreover, prices can be much lower than those in other markets because of the importance of hydraulic power plants and the low electricity consumption. Lastly, this area is far from the Swiss and British markets, whose Day-Ahead prices play an important role in the prediction of close markets. Because the *GNN* updates nodes embeddings at each training step rather than summarizing the precedent input using a lower number of neurons, market specificites are conserved and information from further markets is better incorporated. This ability lowers the contribution of Γ or $\widetilde{F}$ in the forecast since flows are internally modeled by the weighting of a node's neighbor information. This is even more the case for isolated zones as their relationships with other zones are simpler.

4.6 Conclusion

While many works have focused on the construction of increasingly sophisticated models for specific regions of the European market, we proposed to tackle the problem of day-ahead electricity price forecasting at the European scale. Due to its ability to preserve the market's specificities, the Graph Neural Network is the best model available to solve this task. Many options are still to be explored : model the problem as a Graph Labeling task, consider the Edge Labeling task to estimate the flows or searching for more optimal value of hyper-parameters. Nevertheless, we hope that our novel approach for solving the EPF task will benefit the EPF community.

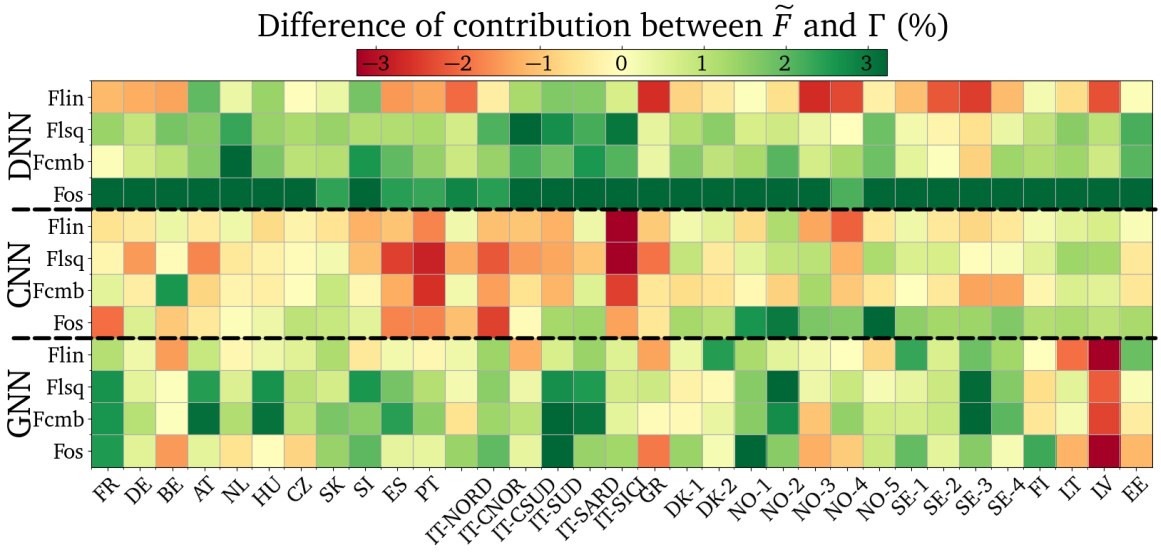


Figure 4.7 – Difference in contribution made by the flow estimates $\widetilde{F}$ compared to the available transfer capacity Γ for the different models and zones. The green (resp. red) squares indicate that $\widetilde{F}$ contributes more (resp. less) than Γ.

Moreover, we presented a way of estimating features based on domain knowledge, and this upstream of learning. Our results on the flow estimation task are promising since we showed that this new feature significantly increases the performances on the price forecasting task for less sophisticated models. A next step on this direction would be to assess the effects of including flow estimates in simple and single-market models, such as the SVR models, to improve their performances even more.

Furthermore, our *optimize then predict* strategy brings the EPF models closer to the EUPHEMIA price-fixing algorithm. We have considered market interdependence caused by price regulation mechanisms and investigated the possibility of solving EUPHEMIA prior to learning to explicitly enforce the different network constraints such as the Energy Balance or the capacities of the connection lines. While our flow estimates showed to increase performances, this approach has the drawback of being independent of model training. In future work, we will study ways of taking into account EUPHEMIA during model training.

Chapter 5

Electricity Price Forecasting based on Order Books: a differentiable optimization approach

In the literature, there is no Data-Driven approach that considers the EUPHEMIA algorithm. In the preceding Chapter, we integrated elements from the Price-Fixing Algorithm by replicating the flow estimation procedure. We significantly improved performances, but had to diverge from the EUPHEMIA logic by omitting the order books. In this Chapter, our approach is to forecast Order Books using a NN model, then use EUPHEMIA to compute the prices. To this aim, we build our own version of EUPHEMIA that computes the Day-Ahead Price from an Order Book, using a dichotomy search. This method has the advantage of being differentiable, and can thus be considered during the training of a Neural Network. In practice, we faced many implementation details that mitigate our conclusion. Our approach fails to improve accuracy on markets dominated by nuclear plants (France) and has to be combined with a standard *DNN* to significantly improve performances on other markets. Nevertheless, this differentiable optimization approach is a significant addition to the literature because it is the first to consider EUPHEMIA in a Data-Driven model.

5.1 Introduction

5.1.1 Motivations

As electricity is difficult to store, many specialists have aimed at anticipating electricity prices to better adjust production. In Section 2.3, we have seen that the most accurate price forecasting models were black-box Machine Learning models. In Chapters 3 and 4, we have shown how to increase both accuracy and consistency of the models by using explainability tools or including domain-knowledge as input features. However, the methods that we have used are both limited because they don't fully consider the price-fixing algorithm. Because of the complexity of EUPHEMIA, it's possible to estimate the price with a Data-Driven approach only up to a certain accuracy. The solution for improving accuracy and consistency even more is to consider EUPHEMIA.

From the literature review (see Chapter 2.3), we observed that Expert models approaches, that aim at replicating EUPHEMIA, yield inaccurate results. This is mainly because those approaches consider fixed relationships between price and its drivers. These relationships are subject to heavy incertitude that feed back to the price estimates. Data-Driven, on the other hand, are adapted to the drivers volatility. Mixing them with Expert models seems to be an adequate solution that has been barely explored. In Chapter 4, we paved the way for combining Expert models and Data-Driven approaches by solving the flow estimation problem prior to learning. In this approach, we simplified the *social welfare* expression to only consider cross-markets flows. Also, the solving occurred prior to learning. In the present Chapter, we aim at taking a step further in that direction by fully considering the EUPHEMIA algorithm during training.

5.1.2 Differentiable optimization approach

The general idea is to forecast the Order Books (the inputs of EUPHEMIA) using a ML model, before solving the problem. Various works have studied the coupling of a prediction model with an optimization task (see Chapter 2.4). In the predict-then-optimize approach [El Balghiti et al., 2019], a predictive model is first built and then used to optimize decision-making. However, the learning of the model is not guided by the prediction errors on the final task related to the optimization problem. Conversely, the predict-and-optimize framework proposes to learn a predictive model by directly minimizing the error related to the downstream decision-making task [Mandi et al., 2022]. This requires differentiating through the solution of the optimization problem. In Section 2.4.4 we concluded that the current state-of-the art of Constrained Optimization Learning was not applicable to EUPHEMIA.

In this chapter, three models are considered. First, a standard machine learning approach predicts day-ahead prices based on exogenous variables. The loss used minimizes the difference between predicted and real prices. The second model predicts order books by minimizing the difference between predictions and actual order books. The last model predicts order books with a neural network and then solve the EUPHEMIA optimization problem. The resulting prices are then compared to real prices. During training, the derivation

of the loss after optimisation is used to adjust the parameters of the neural network. Finally, by making these three models share a common neural network, it is possible to combine them using a loss resulting from the linear combination of the three previous losses. We can then evaluate empirically and on different data sets the impact of each model on the accuracy of the predictions. We aim to show that end-to-end predict-and-optimize methods are beneficial for this problem.

5.1.3 Contributions

Our contributions are summarized as follows:

1. We formalize EUPHEMIA optimization problem, a Mixed-Integer Quadratic Programming problem with linear orders, as the maximization of *social welfare* measured by the difference between market price and order book prices, for all accepted orders (supply orders with prices inferior to the market price, and demand orders prices superior to the market price).

2. We present the dual optimization problem that links EUPHEMIA to the market prices. We then show how to analytically solve this problem by setting it's derivative to zero using a dichotomy search. To the best of our knowledge, our work is the first decision-focused learning method to predict the electricity price, taking into account the EUPHEMIA optimization through differentiable optimization.

3. We then give the implementation details of our solver. Using the torch[1] software, we implement a batched and differentiable dichotomy search.

4. We explain how to integrate the optimization problem into the neural network model, by deriving the calculations of the backward pass. Several neural networks and their associated loss allow the model to be adjusted according to real prices or order books. We present experiments performing a deep analysis of seven configurations on four datasets. We provide an extensive discussion of our results in context.

5.1.4 Chapter structure

The remainder of the Chapter is structured as follows. In Section 5.2 the EUPHEMIA optimization problem is framed based on its definition. We show how to compute the *social welfare* of a linear supply order and generalize its formula. Assuming that model predictions can be improved by tightly coupling the problem with Order Book prediction from exogenous data, a method for solving EUPHEMIA to determine the electricity price from order books is proposed in (Section 5.3). The next Section 5.4 is dedicated to explain the implementation of the differentiable dichotomy search. Then, three strategies of combining an optimal decision process while learning the order book prediction model are presented in Section 5.5.1. The experimental evaluation (Section 5.5.4) subsequently compares several configurations of this end-to-end predict-and-optimize model on 4 different European datasets. Afterwards, a qualitative analysis determines what information has been captured by the differentiable

[1] https://pytorch.org/docs/stable/torch.html

optimization problem using Shap values. Finally, in Section 5.6, we discuss our approach and results, and we provide future work.

5.2 EUPHEMIA optimization problem

In constrast with the Optimization problems described in Chapter 4.2, we consider the Order Book concept from the original EUPHEMIA algorithm. As electricity cannot be stored, the market is regularized to guarantee the balance between supply and demand at European scale, by prioritizing the least expensive means of production (see Section 2.1.2 and 2.3.3 for broader explanations). This price harmonization aims at maximizing the *social welfare* among all participants. In this section, we present the formalization of EUPHEMIA that we will solve in later sections.

5.2.1 Computing the *social welfare* of an order

EUPHEMIA maximizes the sum of each order's *social welfare*, and enforces constraints of the network. We start by detailing the determination of *social welfare* for a supply linear order represented in Figure 5.1 by using :

- A price range $[Po_i, Po_i + P_i]$
- A volume V_i
- A variable $A_i \in [0, 1]$ which state whether the order i is accepted or not.

If the optimal price P^* is strictly lower than Po_i, then it's not worth it for the supplier to turn on its plants and the order is fully rejected ($A_i = 0$, LEFT plot). Oppositely, if P^* is above $Po_i + P_i$, then all the volume V_i is sold and $A_i = 1$ (RIGHT plot). Lastly, if P^* is comprised in the price range, a volume linearly dependant on the price is sold and A_i is between 0 and 1 (CENTER plot). The *social welfare* corresponds to the commercial *surplus* of an order : what the player receives minus what he was willing to be paid EP. As P^* is used for exchanging all orders, the actual payment is always $A_i V_i P^*$. For a plant owner, EP is the marginal cost of a plant. By submitting an order, a supplier expresses his wish to sell energy at a price $P(v_i)$ that is linearly dependant on the volume he sells $v_i = A_i V_i$:

$$P(v_i) = \frac{P_i}{V_i} v_i + Po_i$$

The expected payment $EP(v_i)$ can then be found by taking the integral of $P(v_i)$:

$$\begin{aligned} EP(v_i) &= \int P(v_i)\, dv_i \\ &= \frac{1}{2}\frac{P_i}{V_i} v_i^2 + Po_i v_i + \theta_i \end{aligned}$$

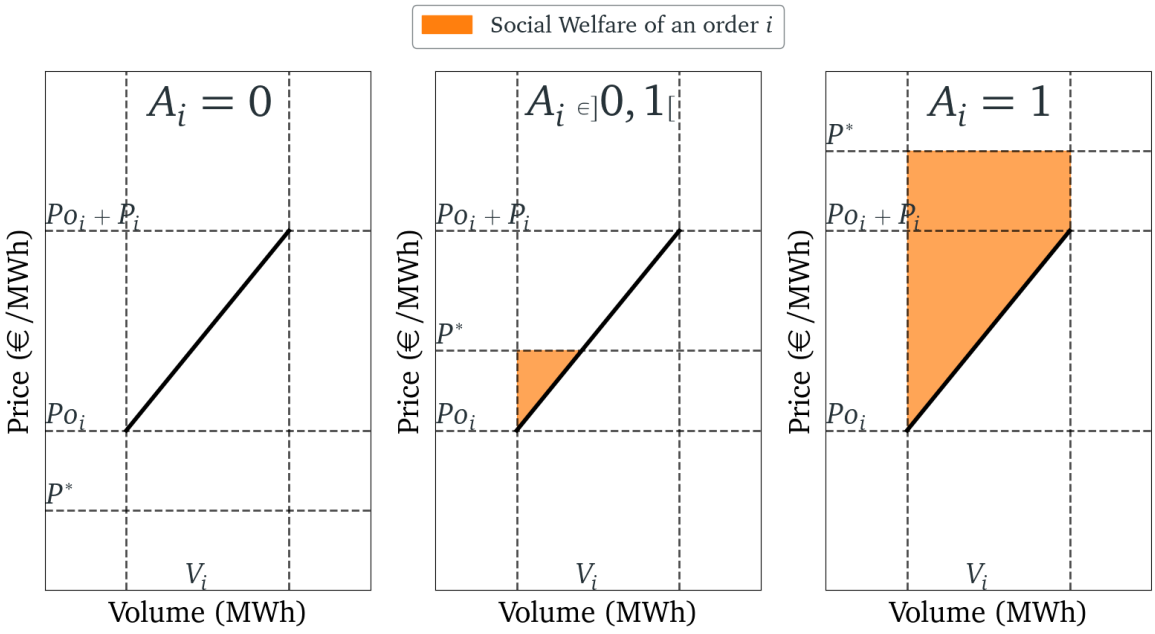


Figure 5.1 – A linear supply Order is defined by a price range $[Po_i, Po_i + P_i]$ and volume V_i. (LEFT) $P^* < Po_i$, the order is rejected. (CENTER) $P^* \in [Po_i, P_i]$, the order is partially accepted. (RIGHT) $P^* > Po_i + P_i$, the order is accepted. The curve equation for the price of selling is $P(v_i) = \frac{P_i}{V_i} v_i + Po_i$. The area in orange represents the *social welfare* brought by the order.

with constant θ_i independent on v_i. We can then reinject the expression of v_i on the previous equation and express the $SW(i)$ by substracting $EP(v_i)$ to the real payment $A_i V_i P^*$:

$$EP(v_i) = \frac{1}{2} A_i^2 V_i P_i + A_i V_i Po_i + \theta_i \tag{5.1}$$

$$SW(i) = A_i V_i P^* - \frac{1}{2} A_i^2 V_i P_i - A_i V_i Po_i - \theta_i \tag{5.2}$$

5.2.2 Definition of EUPHEMIA

The determination of the *social welfare* for a demand order follows the same logic, and leads to the same expression if using negative volumes V_i and price range P_i. By summing expression (5.2) for all orders, we obtain:

Definition 1 (EUPHEMIA optimization problem). EUPHEMIA *is defined by the following convex*

quadratic optimization problem :

$$\max_{A} \sum_{i \in OB} \left(-\frac{1}{2} A_i^2 V_i P_i - A_i V_i P_{o_i} \right) \tag{5.3}$$

$$u.c. \quad \sum_{i \in OB} A_i V_i = 0, \tag{5.4}$$

$$-A_i \leqslant 0, \tag{5.5}$$

$$A_i - 1 \leqslant 0 \tag{5.6}$$

Since the constants θ_i are not affected by the decision variables A_i we remove them . Also, using the *energy balance* constraint $\sum_{i \in OB} A_i V_i = 0$ (5.4), we can remove the reward part $P^* \sum_{i \in OB} A_i V_i$.

5.3 Finding the optimal price from an order book

In order to improve electricity price forecasts, we want to integrate the EUPHEMIA pricing mechanism into the predictive model, as we believe that this will increase the accuracy of the predictions. Market players submit their orders before 12 a.m. each day. EUPHEMIA then calculates Day-Ahead prices that maximize *social welfare*. While EUPHEMIA is reproducible to some extent, the Order Book is not known until the prices are released. Hence, we propose to predict the Order Book using fundamental variables and use the predicted orders as input to EUPHEMIA optimization problem to get the price predictions.

5.3.1 Dual EUPHEMIA optimization problem

From Definition 1, the relationship between the Day-Ahead Price and the optimal solution of EUPHEMIA remains unclear. To obtain the prices, we have to consider the dual problem. For a deeper economic interpretation, one can refer to MIT's lecture about Optimization Methods in Management Science [Orlin, 2023]. Appendix E.1 gives graphical insights on why the solution to the dual problem is the Day-Ahead Price, and detailed computations are available. To determine the dual problem from Definition 1, we introduce $\lambda \in \mathbb{R}$, $M \in \mathbb{R}^N$ and $K \in \mathbb{R}^N$ as the dual variables associated to the constraints in equations (5.4)-(5.6) and consider the Lagrangian:

$$\mathcal{L}(A, \lambda, M, K) = \sum_{i \in OB} \left(-\frac{A_i^2 V_i P_i}{2} - A_i V_i Po_i + \lambda A_i V_i - M_i A_i + K_i (A_i - 1) \right)$$

Leveraging the different Karush–Kuhn–Tucker necessary conditions for optimality, we obtain the dual problem of EUPHEMIA (Appendix E.2):

Definition 2 (Dual of EUPHEMIA)**.**

$$\min_{\lambda} \sum_{i \in OB} \mathcal{D}_i(\lambda) \tag{5.7}$$

$$with\ \mathcal{D}_i(\lambda) = \begin{cases} (1)\ 0, & if\ V_i(Po_i - \lambda) > 0 \\ (2)\ V_i(\lambda - \frac{P_i}{2} - Po_i), & if\ V_i(\lambda - P_i - Po_i)) > 0 \\ (3)\ \frac{V_i}{2P_i}(\lambda - Po_i)^2, & if\ \lambda \in [Po_i, Po_i + P_i] \end{cases} \tag{5.8}$$

It becomes clear that the optimal dual variable $\lambda^\star$, is the day-ahead price. Then, (1) corresponds to the situation of a fully rejected order where the optimal price $\lambda^\star$ is lower than Po_i for the supply orders ($V_i > 0$) and higher for the demand orders ($V_i < 0$). Inversely, (2) corresponds to fully accepted orders. (3) happens when the order is partially accepted ($\lambda^\star$ is in the price range and the proportion $\frac{\lambda^\star - Po_i}{P_i}$ of volume V_i is exchanged).

5.3.2 Computing the optimal price $\lambda^\star$

The minimum of $\mathcal{D}(\lambda)$ can be obtained by looking for the values for which the derivative of $\mathcal{D}$ vanishes: $\frac{\partial \mathcal{D}}{\partial \lambda} = 0$. Each segment of the piecewise function $\mathcal{D}_i$ is differentiable or equals 0. Only the inflection points Po_i and $Po_i + P_i$ have to be examined by considering their limits. Let us consider supply orders (similar results can be obtained for demand orders). Using the generic expression

$$\frac{\mathcal{D}_i(\lambda + h) - \mathcal{D}_i(\lambda)}{h} = \frac{2V_i}{P_i}\left(\frac{h}{2} + \lambda - Po_i\right)$$

we can compute the limits for the inflection points with $h \mapsto 0^-, h \mapsto 0^+$. We find that the limit at $\lambda = Po_i$ is 0 for both sides of h, and the limit at $\lambda = Po_i + P_i$ is V_i (Appendix E.3). Hence, $\mathcal{D}_i$ is differentiable for all values of λ and the derivative of $\mathcal{D}$ is

$$\mathcal{D}_i'(\lambda) = \begin{cases} 0 & \text{if } V_i(Po_i - \lambda) > 0 \\ V_i & \text{if } V_i(\lambda - P_i - Po_i) > 0 \\ \frac{V_i}{P_i}(\lambda - Po_i) & \text{if } \lambda \in [Po_i, Po_i + P_i] \end{cases}$$

Lastly, we use the Heaviside function:

$$H(x) = \begin{cases} 0 \text{ if } x < 0 \\ 1 \text{ if } x \geqslant 0 \end{cases}$$

to rewrite $\mathcal{D}'$. Computations (E.4) leads us to:

$$\mathcal{D}'(\lambda) = \sum_i \frac{x_i H(x_i) - y_i H(y_i)}{P_i}$$

with $x_i = V_i(\lambda - Po_i)$ and $y_i = V_i(\lambda - Po_i - P_i)$. This function is strictly increasing (Appendix E.5) and we can use a dichotomy search to solve $\mathcal{D}'(\lambda^\star) = 0$. Using lb, ub as the lower and upper bounds initialized at the market prices limits fixed by EPEX, $\lambda^\star$ is computed with Algorithm 1. Using the Heaviside function allows us to differentiate through IF statements (Appendix E.6) that analytically derives to the Dirac function:

$$\delta(x) = \begin{cases} +\infty & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases}$$

5.4 A differentiable optimization approach for EPF

In this Section, we present in how to integrate the solution of the optimization problem into the price prediction neural network model. First, we present how the difference between the optimal price calculated by Algorithm 1 and the real price is back-propagated to adjust the

Algorithm 1 Differentiable dichotomy search.

lb ← -500€/MWh
ub ← 3000€/MWh
found ←**False**
while (found = **False**) and $(ub - lb > 2 * 0.01)$ **do**
  $M \leftarrow \frac{ub+lb}{2}$
  $D_M \leftarrow D'(M)$
  found ← $D_M = 0$
  $ub \leftarrow ub - H(D_M) * (ub - M)$
  $lb \leftarrow M - H(D_M) * (M - lb)$
end while

parameters of the neural network. Then, we go over several implementation tricks that are necessary to make our solution competitive.

5.4.1 Integrating the optimization process into the forward and backward passes

Figure 5.2 presents the model architecture. Exogenous variables X are provided to a neural network that is trained to predict Order Book. The order book is then used to determine the optimal price, $\lambda^\star$, using Algorithm 1. A loss function L, between $\widehat{Y}$, the optimal price based on estimated Order Book $\widehat{OB}$, and the real price Y, evaluates the error made on the predictions. This error has to be back-propagated through the network that learns $\widehat{OB}$ from exogenous features X:

$$\frac{\partial L(\widehat{Y}, Y)}{\partial X} = \frac{\partial L}{\partial \widehat{Y}} \times \frac{\partial \widehat{Y}}{\partial \widehat{OB}} \times \frac{\partial \widehat{OB}}{\partial X}$$

In this expression, the first term is the gradient of the loss, and the third term is the standard back-propagation. The second term, $\frac{\partial \widehat{Y}}{\partial \widehat{OB}}$ is obtained by differentiating the dual problem formulation. Therefore, Algorithm 1 is implemented using PyTorch [2] and is used to solve EUPHEMIA during the forward pass. It is also used to compute the derivative of the order books with respect to the optimal price $\frac{\partial \widehat{Y}}{\partial \widehat{OB}}$ during the backward pass. Note that the additive property of the gradient decompose the final $\frac{\partial \widehat{Y}}{\partial \widehat{OB}}$ into a sum of gradient brought by each variable m that is computed using $\widehat{OB}$, in our case the value of $\mathcal{D}'$ for each search iteration k:

$$\begin{aligned}\frac{\partial \widehat{Y}}{\partial \widehat{OB}} &= \sum_m \nabla_m \frac{\partial m}{\partial \widehat{OB}} \\ &= \sum_{k=1}^{N-1} \nabla_{\mathcal{D}'_k} \frac{\partial \mathcal{D}'_k}{\partial \widehat{OB}}\end{aligned}$$

Thus, it's not necessary to explicitly formulate $\frac{\partial \widehat{Y}}{\partial \widehat{OB}}$.

[2] https://pytorch.org/docs/stable/index.html

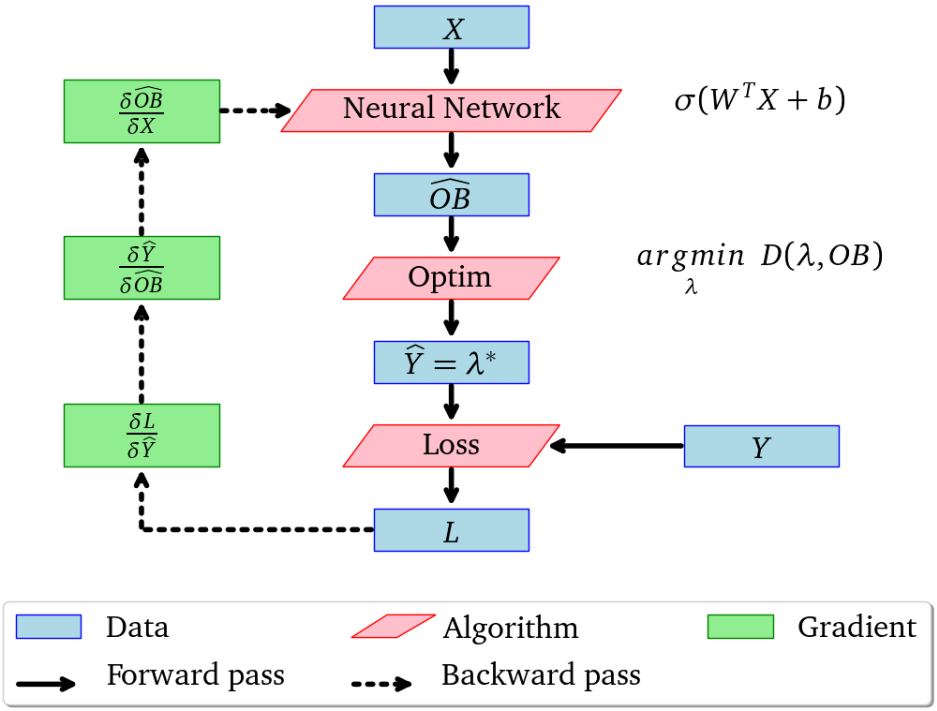


Figure 5.2 – Differentiable optimization: Predict the order book variables $\widehat{OB}$, find optimal prices given the order book, and back-propagate the errors on the prediction model.

5.4.2 Implementation tricks

5.4.2.1 Approximating the Heaviside function

The implementation of Algorithm 1 is not straightforward. Because we go from the continuous analytic space to the discrete numeric space, some assumptions such as the differentiability of the Heaviside function don't hold. We thus replace the Heaviside function by the Sigmoid function:

$$H(x) = \lim_{k \to \infty} \sigma(kx)$$

The parameter k controls the sharpness of the slope. The derivative of sigmoid is:

$$\sigma'(kx) = \sigma(kx) * (1 - \sigma(kx))$$

In Figure 5.3, we ran Algorithm 1 using the real OB and computed the difference with the real prices Y, using several values of k and the Heaviside function. We find that the price difference is neglectable (inferior to 3cts/MWh) even for small values of k. As convergence to results using Heaviside is reached when $k = 100$, we used this parameter throughout our experiments.

5.4.2.2 Batch solver

PyTorch is implemented to perform efficient operations on batches of data. Solving the optimization problem independently for each data instance drastically increases the training

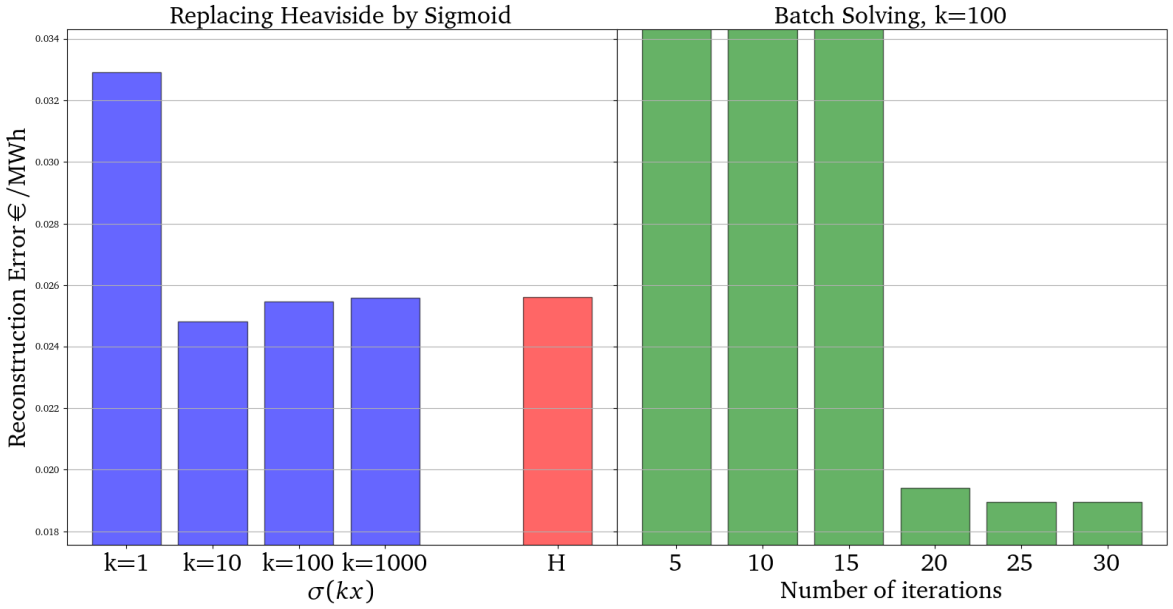


Figure 5.3 – (LEFT) Heaviside approximation using sigmoid (RIGHT) evaluation of the Batch Solver.

time. Hence, we propose to rewrite Algorithm 1 to perform the dichotomy search on batches of data. The only way to vectorize operations is to determine a fixed number of iterations N for the search loop. Figure 5.3 displays the price error using various values of N. We observe that only 20 iterations are sufficient to achieve better performances than with the for-loop. This is due to two reasons:

- The search converges. While finding $\mathcal{D}'(M) = 0$, $H(\mathcal{D}'_M) = \frac{1}{2}$ and the update of ub, lb does not change their values.
- The search space ranges from -500€/MWh to 3000€/MWh with a cent step, which is fully explored in 19 iterations by an $O(log2(n))$ algorithm.

In our experiments, we pick $N = 30$ iterations and solve the problem on batches of $30 * 24$ data instances.

5.4.2.3 Shrinking Order Book

Order Book are complex data structure that is not straightforward to use as labels to predict in a ML model. They consist in tensors $OB_z^{(d,h)} \in \mathbb{R}^{n_o \times 3}$ with n_o a variable number of orders, typically comprised between 500 and 2500. We thus start by defining a mapping $\mathcal{S}$:

$$\mathcal{S} : \Phi \mapsto \mathbb{R}^{n_t \times 3}$$
$$OB_z^{(d,h)} \to \mathcal{S}(OB_z^{(d,h)})$$
$$\text{such that } \mathcal{E}\left(\mathcal{S}(OB_z^{(d,h)})\right) = \mathcal{E}\left(OB_z^{(d,h)}\right)$$

that reduces the OB to a much lower and fixed dimmensionality $n_t << n_o$ while not changing the solution of EUPHEMIA $\mathcal{E}$. For a desired dimension of size $n_t = n_S + n_D + 4$, we select the n_S supply orders and n_D demand orders closest to the intersection price. The orders not selected are either: supply orders before O_{SB} or after intersection O_{SA}, or demand orders before O_{DB} or after intersection O_{DA}. The four sets of unselected orders are each summarized by a fictitious orders, defined by the following values:

Variable	O_{SB} and O_{SA}	O_{DB} and O_{DA}
V	$\sum_{i \in O} V_i$	$\sum_{i \in O} V_i$
Po	$\min_{i \in O} Po_i$	$\max_{i \in O} Po_i$
P	$\min_{i \in O} (Po_i + P_i) - Po$	$\max_{i \in O} (Po_i + P_i) - Po$

Graphically, this replaces the step curve of the portion of unselected orders by a line, as displayed in the left-hand side of Figure 5.4.

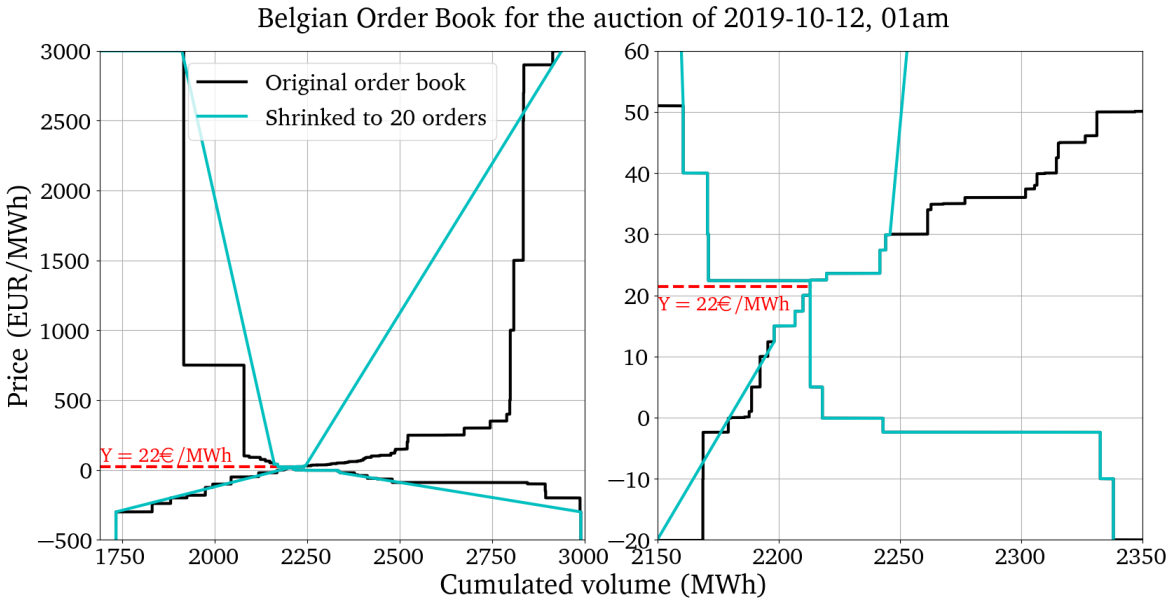


Figure 5.4 – The order book of the 2019/10/12, 1am auction in Belgium (black), and its reduced version to 20 orders (blue). Around the intersection, the real and reduced order books are the same (right). Orders far from the intersection are replaced by a straight line. Both real and reduced order books cover the same volume range and solve to the same price (left).

5.5 Experiments

In this section, we describe a series of experiments with multiple objectives. First, we want to compare our Order Book model and differentiable EUPHEMIA method to the baseline (direct neural network) and determine if they improve the performance. We then analyze more precisely the impact of varying the weight of the optimization problem in the loss to understand its links with the model performance. Lastly, we seek to link the effects

of differentiable optimization to the input variables. For this, we perform a contribution analysis using Shap values to identify which features have been put forward by adding differentiable optimization.

5.5.1 Varying the impact of optimization problem on model learning

To be able to vary and evaluate the impact of the integration of the optimization process within the predictive model, we propose to consider three scenarios (see Figure 5.5). The first one, is a traditional neural network used to solve EPF problem. The neural network directly predicts prices $\widehat{Y}_{DNN}$ from exogenous variables without considering order books and compares them to real prices, computing L_{DNN}. The second branch, uses the exogenous variables and order books from the last day $OB^{(d-1)}$ to predict order books $\widehat{OB}^{(d)}$. The loss function L_{OB} evaluates the difference between the predicted order books and the true ones. The last scenario, is the differentiable optimization approach that uses $\widehat{OB}$ to solve EUPHEMIA and get the price estimates $\widehat{Y}_{DO}$, then computes the loss L_{DO} between solved prices and real prices. As these three models share a neural network NN, it is possible to combine them. We propose to evaluate different linear combinations of them. We define the general loss as a linear combination of the three losses: $L = \alpha L_{DNN} + \beta L_{DO} + \gamma L_{OB}$. By using the *SMAPE* for the different losses, they produces values in the same range and their weights α, β, γ in the combination are consistent. The final price predictions are computed as:

$$\widehat{Y} = \frac{\alpha\widehat{Y_{DNN}} + \beta\widehat{Y_{DO}}}{\alpha + \beta}$$

5.5.2 Datasets

We consider the EPF problem on the European market where the data is available free of charge[3]. We forecast the prices for 4 countries: France (FR), Germany (DE), Belgium (BE) and the Netherlands (NL). As predictive variables, we use the consumption forecasts, the generation forecasts, the renewable generation forecasts and the current prices of nine European countries: France, Germany, Belgium, the Netherlands, Austria, Italy, Spain, Switzerland and England. To those variables, we add the reference gas price, as well as the date indicators (day, day of week, week, month) that are circularly encoded. Hence, each day is described by $9 + 36 \times 24$ predictive features and the targets to be predicted are the 24 hourly prices for each country. The variables can be grouped into families: domestic variables (variable of the country being predicted), foreign variables (variable of another country), gas price and date. The Swiss and English prices are also considered separately because they are available at 11 am and can be used in the training set. Additionally, it is possible to include order books from the previous day as predictive features, made available by the Epex exchange against a fee. Our dataset spans from 01/01/2016 to 31/12/2019. We use the last year (2019) as test set, to account for the prices seasonality.

[3] `https://transparency.entsoe.eu`

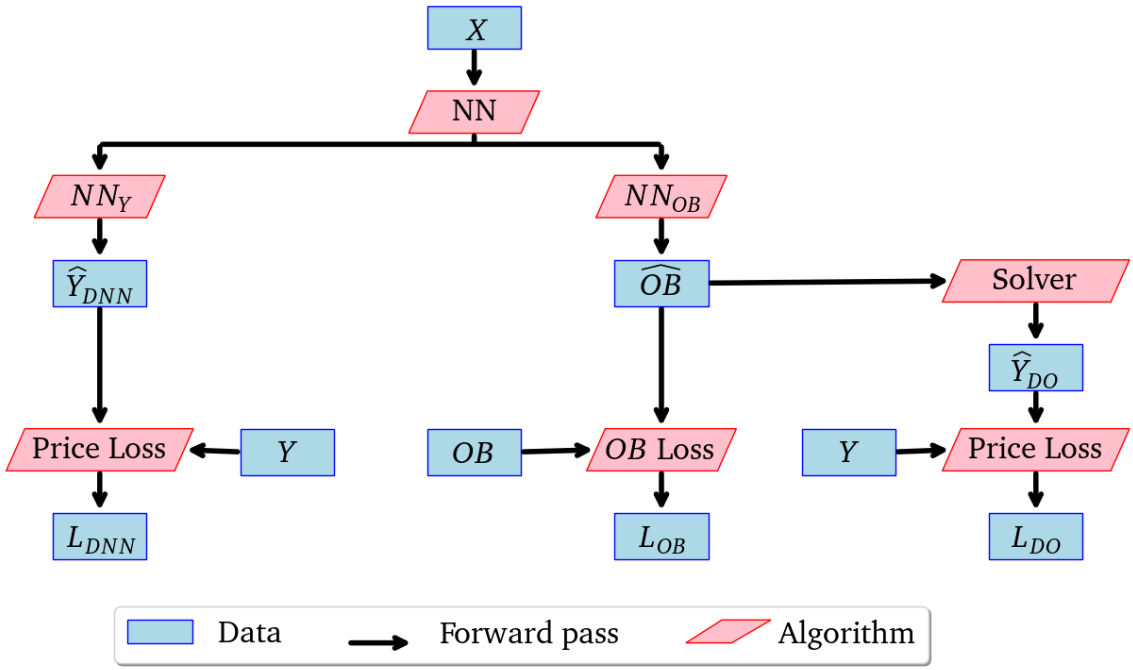


Figure 5.5 – Three models architecture combined in one. The left branch directly forecasts the prices from the exogenous variables. The middle branch forecasts the OB and the right branch solves the EUPHEMIA problem. The final loss is $L = \alpha L_{DNN} + \beta L_{DO} + \gamma L_{OB}$.

5.5.3 Models' implementation

We elaborate on the models introduced in Figure 5.5. The NN part is a dense layer with 873 inputs and 888 outputs, followed by batch normalization, dropout and ReLU activation layers. The NN_Y part is a dense layer with 888 inputs and 24 outputs. The NN_{OB} part is detailed on Figure 5.6. This network receives as input X the output of the shared neural network NN. The input data is first reshaped to hourly granularity, and then sent through a dense layer with 37 inputs and 37 outputs, followed by batch normalization, dropout and ReLU activation layers. This is followed by 6 distinct dense layers with 37 inputs and 20 outputs that forecast the components of the supply and demand sides.

We establish two baseline models and four models to test, by setting α, β or γ to 0. We define these models in Table 5.1, where the first two are common DNN models from the literature (see [Lago et al., 2021]) since they do not use the differentiable EUPHEMIA solver ($\beta = 0$). The last 4 models are the novel models ($\beta > 0$).

5.5.4 Results

5.5.4.1 Configurations

We compare the values of the predicted $\widehat{Y}$ with the real Y target variable. We use standard measures such as $MAE(Y, \widehat{Y})$, $DAE(Y, \widehat{Y})$, $RMAE(Y, \widehat{Y})$ and $SMAPE(Y, \widehat{Y})$. To check the statistical significance of the results, we use the Diebold & Mariano (DM) test [Diebold and

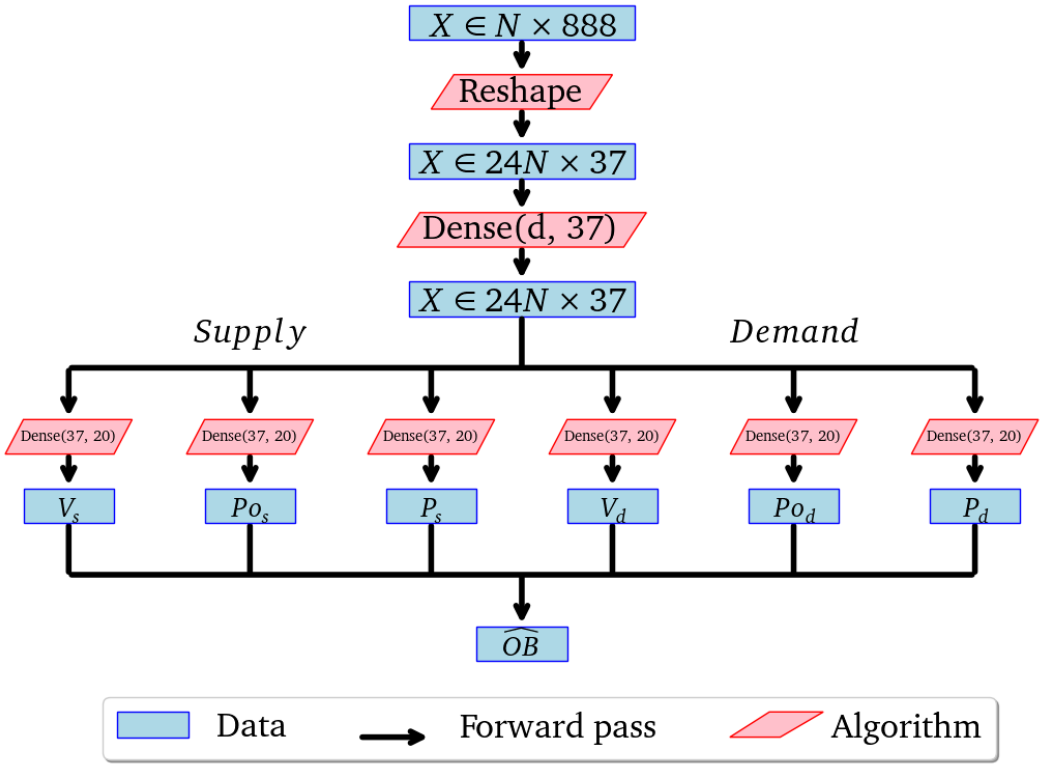


Figure 5.6 – Implementation details of the NN_{OB} network. The input X, coming from the common NN part, is reshaped to hourly granularity and goes through a dense layer. Then, it is fed to 6 different linear layers to forecast supply and demand components of the order book.

Mariano, 2002] with a fixed threshold of 0.05. The source code and the free data are made available.[4] The metrics computed on the test period are displayed in Table 5.2. The p-values of the Diebold & Mariano test are displayed in Figure 5.7. Our observations are the following:

- On the French market, dominated by nuclear energy production, where the marginal cost is independent of other variables, the order books are more difficult to predict and less significant. Consequently, the metrics of the baseline models DNN_Y and $\text{DNN}_{Y,OB}$ are significantly better : the order book modeling plus differentiable optimization models fail to improve performances.
- On the German, Dutch and Belgian datasets, models DO + DNN_Y and DO + $\text{DNN}_{Y,OB}$ outperform other models with statistical significance. These markets are characterized by a significant gas or coal-fired electricity generation, whose marginal costs are tied to the commodity prices. Such pricing produces more relevant order books, and hence, the models using differentiable optimization are more adapted to these markets.
- Models DO and DO + DNN_{OB} that do not use direct forecast ($\alpha = 0$) are often outperformed by others on every datasets. Models using the differentiable optimization are

[4]CODE: `https://github.com/Leonardbcm/MOB`, DATA: `https://rb.gy/v9ui3`

Model	α	γ	β	Description
DNN_Y	1	0	0	Predict $\widehat{Y}$ using a standard DNN.
$DNN_{Y,OB}$	$\frac{1}{2}$	$\frac{1}{2}$	0	Predict $\widehat{OB}$ and $\widehat{Y}$ without differentiable optimization.
DO	0	0	1	Pure differentiable optimization model: there is no loss on $\widehat{OB}$, and $\widehat{Y}$ is only obtained through the solving the optimization problem.
DO + DNN_{OB}	0	$\frac{1}{2}$	$\frac{1}{2}$	A loss is applied to $\widehat{OB}$, but $\widehat{Y}$ is only obtained through solving.
DO + DNN_Y	$\frac{1}{2}$	0	$\frac{1}{2}$	No $\widehat{OB}$ forecast loss. $\widehat{Y}$ is obtained directly, but also by solving the optimization problem.
DO + $DNN_{Y,OB}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	The three models are combined.

Table 5.1 – The different models derived from our architecture. Models DNN_Y and $DNN_{Y,OB}$ are the baselines that do not use the differentiable optimization.

better trained when the loss is also computed using direct prediction. This suggests a deeper analysis of the α and β parameters.

5.5.4.2 Varying the β parameter

In this experiment, we focused on the Belgian dataset. For this dataset, we have seen that model DNN_Y ($\alpha = 1$) is outperformed by model DO ($\beta = 1$), but both models are outperformed by model DO + DNN_Y ($\alpha = \frac{1}{2}$ and $\beta = \frac{1}{2}$) and DO + $DNN_{Y,OB}$ ($\alpha = \frac{1}{3}$ and $\beta = \frac{1}{3}$). Our purpose is to find even more adequate values for α and β parameters. To this aim, we start from the configuration of DNN_Y ($\alpha = 1$) and increase β by steps of 5%, while decreasing α by steps of 5%. Results are displayed in Figure 5.8. It is clear that not considering the optimization problem during training ($\beta = 0$) does not yield the best results. However, considering only the differentiable optimization part ($\beta = 1$) is not an adequate solution either. We see in Fig. 5.8 that adding the optimization loss even with a very small weight in the global loss (5%) increases all the considered metrics.

5.5.4.3 Contribution Analysis

We now perform a contribution analysis using SHAP [Lundberg and Lee, 2017]. Our aim is to determine which features have been prioritized by adding the differentiable optimization problem, and how the weight of the differentiation in the loss affects features contribution. To this aim, we compute the difference of contribution between a variable with $\beta > 0$ and its contribution when $\beta = 0$. For each value of β, we compute 1000 SHAP values on the test set. For a clearer analysis, we regroup the contributions by families of variable and we display them on Fig. 5.9. Colored squares quantify the variation of contribution between $\beta = 0$ and $\beta > 0$. We observe almost no variation on the domestic features (first 4 columns), with only a slight increases of the price contribution at the expense of the generation forecast (gen). Changes for Foreign features are more pronounced. The contribution of the Foreign Prices (F. price) increases for smaller values of β while higher values favor the Swiss and English

Country	Model	MAE	DAE	RMAE	SMAPE
BE	DNN_Y	7.74	5.79	0.941	21.27
	$DNN_{Y,OB}$	9.63	4.17	1.17	26.48
	DO	7.27	4.37	0.884	19.73
	DO + DNN_{OB}	19.85	19.49	2.425	42.19
	DO + DNN_Y	6.85	4.32	0.832	20.35
	DO + $DNN_{Y,OB}$	**6.28**	**3.44**	**0.763**	**17.28**
DE	DNN_Y	7.28	6.67	0.778	29.83
	$DNN_{Y,OB}$	8.87	6.52	0.946	30.36
	DO	9.01	6.96	0.958	29.87
	DO + DNN_{OB}	9.24	7.1	0.983	31.24
	DO + DNN_Y	6.99	5.15	0.745	25.97
	DO + $DNN_{Y,OB}$	**6.91**	**4.53**	**0.735**	**25.53**
FR	DNN_Y	**4.54**	3.06	**0.653**	15.5
	$DNN_{Y,OB}$	5.11	**3.03**	0.734	**15.21**
	DO	6.47	4.8	0.93	20.31
	DO + DNN_{OB}	5.92	3.5	0.849	18.25
	DO + DNN_Y	5.3	3.22	0.759	16.2
	DO + $DNN_{Y,OB}$	5.79	3.87	0.831	19.51
NL	DNN_Y	6.32	4.43	1.057	18.84
	$DNN_{Y,OB}$	5.77	3.81	0.965	15.5
	DO	6.53	3.96	1.092	16.47
	DO + DNN_{OB}	10.97	10.34	1.838	25.18
	DO + DNN_Y	**5.22**	**3.49**	**0.874**	**13.4**
	DO + $DNN_{Y,OB}$	5.79	4.47	0.968	14.41

Table 5.2 – Metrics obtained on the test period for different model configurations. Bold values indicate the best metric among all model configurations for a given dataset.

prices (CH price, UK price). We also note an important decrease in the contribution of the Foreign generation forecast (F. ren gen) for all values of β, and of the contribution of the Foreign generation and consumption forecast (F. gen) and (F. conso) for higher values of β.

5.5.5 Discussion

In the scope of our study, the order book modeling and differentiation of the optimization problem have led to significantly better results on the datasets where the energy mix is eclectic. For the Belgian dataset, the analysis of the β hyper-parameter shows that the best model is a combination of the traditional approach where prices are directly forecast from exogenous features $\beta = 0$ and the differentiable optimization approach ($\beta > 0$). This experiment suggests that the optimal value of β could be found using hyper-parameter search methods. The variation in the contribution also reveals that adding differentiable optimization to the model guides it to a more refined representation of the variables. The Domestic and Foreign consumption, generation and renewable generation forecasts are less regarded, in favor of prices. Indeed, while those forecasts are to an extent correlated to the prices, this correlation does not hold with the order books whose complex structure is more

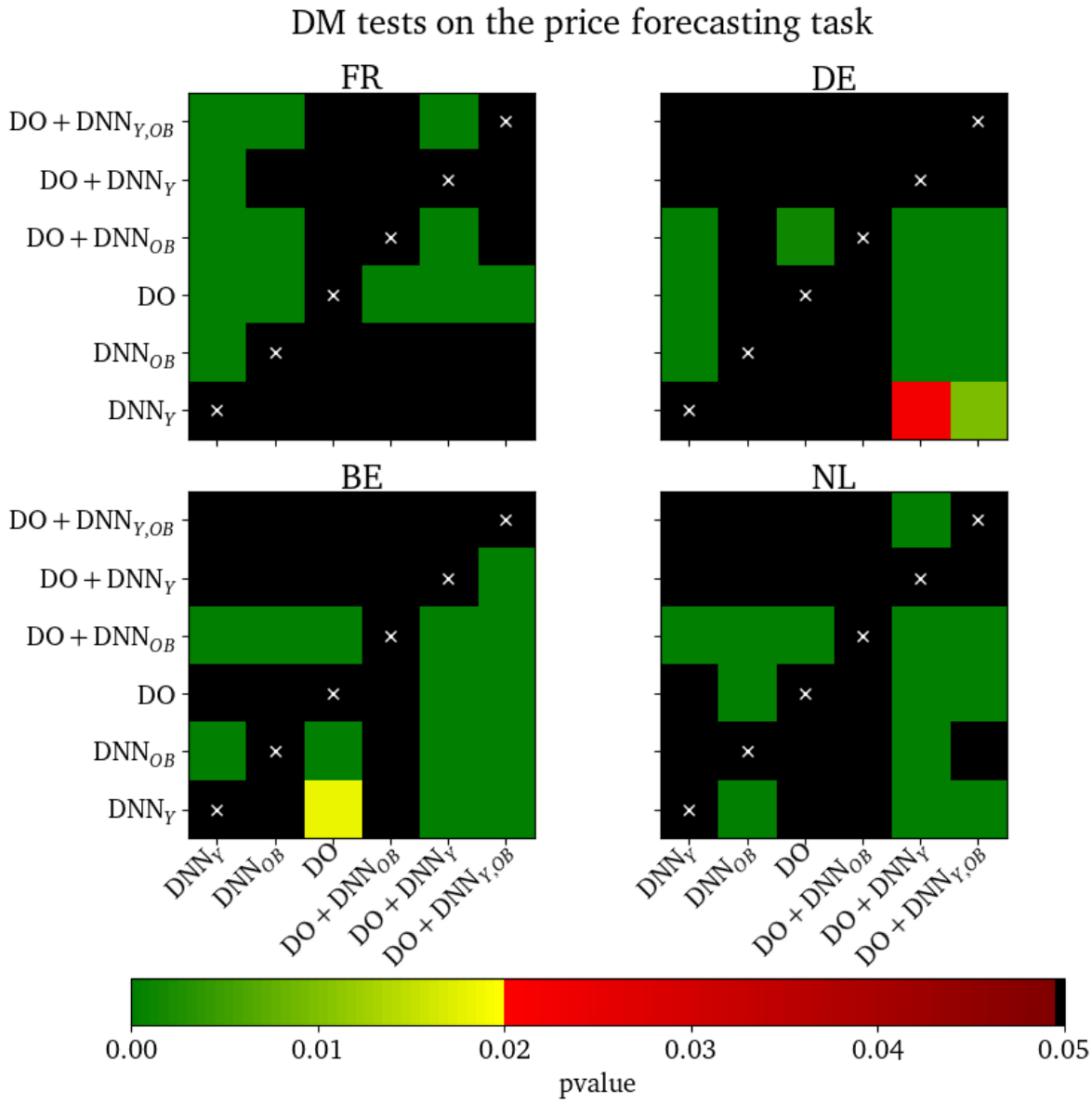


Figure 5.7 – p-values of the Diebold & Mariano test for the price forecast task. Colored squares in (i, j) indicate that the forecasts of model i are significantly more accurate than forecasts of model j. Green columns indicate that the corresponding models are significantly better than every other. Black lines indicate that the model on the y-axis' forecasts are significantly worse than every other.

price-dependent. This is especially true for Swiss and English prices which are set at 11 am and thus available for market players to form their order books.

To illustrate this finding, we provide an extract of the test set predictions in Figure 5.10, where the Belgian day-ahead prices from 10/10/2019 to 12/10/2019 are displayed on the top diagram along with the forecasts of different models ($\beta = 0$ in red, $\beta = \frac{1}{2}$ in yellow and $\beta = 1$

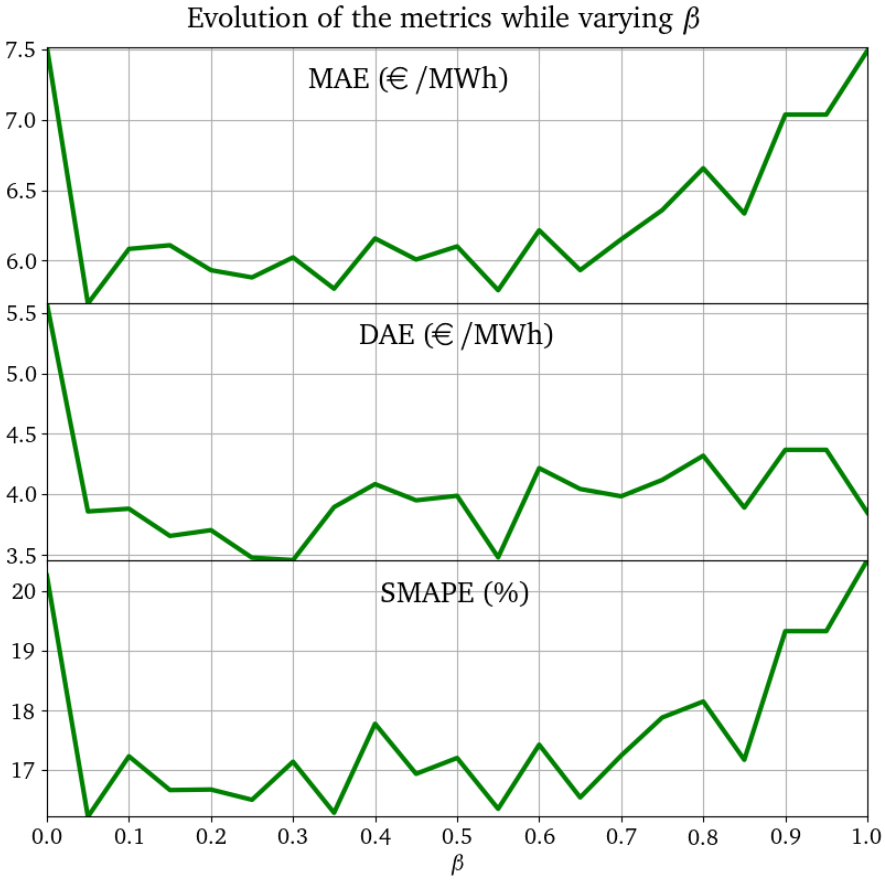


Figure 5.8 – Quality of the obtained predictions according to β on the Belgian dataset: MAE (top), DAE (middle) and SMAPE (bottom).

in green). On the bottom diagram, the market conditions are displayed (forecast generation and forecast residual load), expressed as % of deviation from their normal values. Focusing on the end-of-day auctions of the 11/10/2019: 1 am to 4 am and 11 pm to 3 am of the 12/10/2019, we note that the model without differentiable optimization ($\beta = 0$), given that the residual load is 40% below the normal, greatly underestimates the price. The real sensibility of the price to consumption variations, given by the order book displayed in Figure 5.4, cannot be captured by the model without differentiable optimization, while the other two models with $\beta = \frac{1}{2}$ and $\beta = 1$ predict the price accurately.

5.6 Conclusion

In this Chapter, we addressed the problem of Electricity Price Forecasting (EPF) by combining a prediction model with an optimization task based on the EUPHEMIA algorithm. We aimed to demonstrate the benefits of the end-to-end predict-and-optimize approach for this problem. Our research hypothesis was that tightly coupling the EUPHEMIA optimization task with order book prediction from exogenous data would improve the accuracy of the model predictions. To achieve this, we first formalized the EUPHEMIA optimization problem and

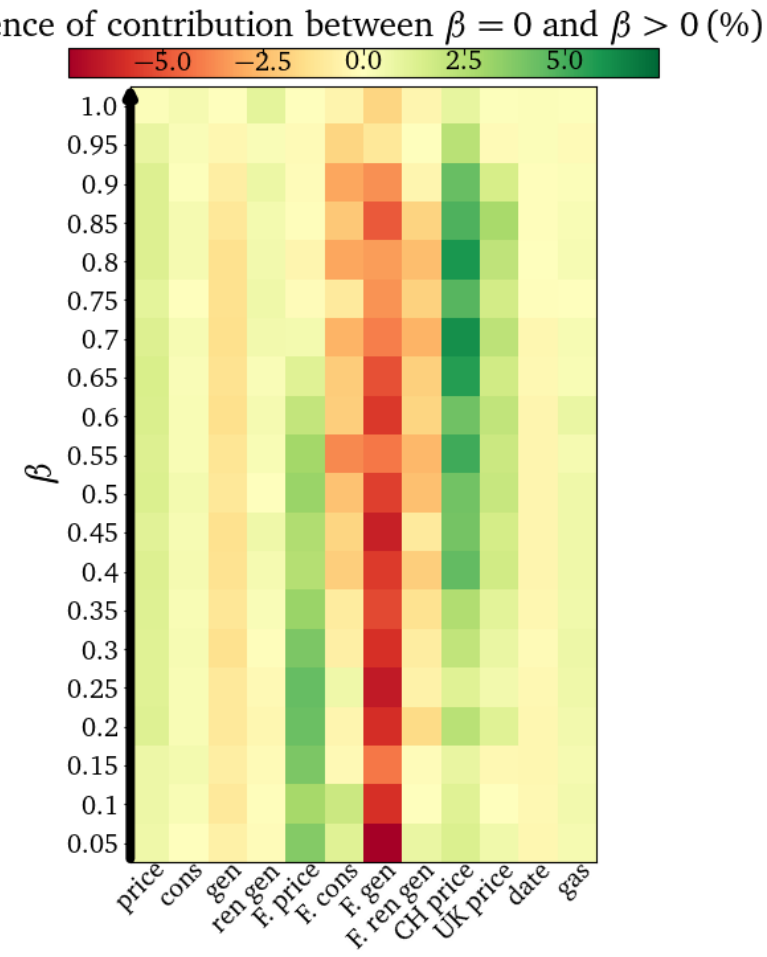


Figure 5.9 – Variation of the feature contributions while increasing β on the Belgian dataset. A green square at coordinate (i, j) indicates that the model trained with $\beta = i$ increase the contribution of feature j compared to the model with $\beta = 0$. A red square indicates the opposite. On the x-axis, the features are regrouped by category. For instance, F. CONS is the sum of contribution of all Foreign Consumption Forecasts (for Belgium, it is the French, German, Dutch, Spanish, Italian, English, Austrian and Swiss prices).

proposed a method for solving it to determine electricity prices from order books. Our approach consisted of a differentiable batch dichotomy search. We then explored different ways of integrating the optimal decision process while learning the order book prediction model. By directly minimizing the error related to the downstream decision-making task, we aimed to improve the overall performance of the EPF model. In the experimental evaluation, we compared several configurations of the end-to-end decision-focused model using four different European datasets. The results demonstrated the effectiveness of our approach, showing improvements in the accuracy of the predictions compared to traditional methods for the datasets in which the exogenous variables make it possible to finely estimate the order books. The tight coupling of the EUPHEMIA optimization task with the order book prediction task then allows the model to capture more relevant information and make more

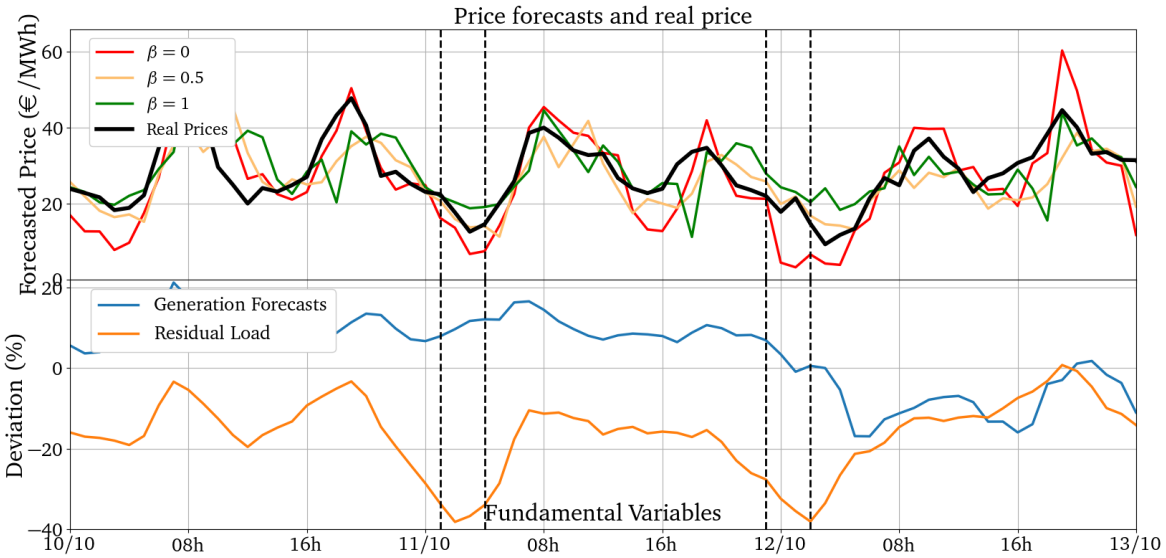


Figure 5.10 – Day-Ahead prices on the Belgian dataset from 10/10/2019 to 12/10/2019 (black line) and their predicted values for various β (Top). Fundamental variables of the Belgian market for the same period, expressed as percentage of deviation from the average (Bottom).

accurate price predictions. Furthermore, we conducted a qualitative analysis using Shap values to understand the factors and features that influenced the predictions and decisions made by the predict-and-optimize model. This analysis provided insights into the captured information and shed light on the relationships between the input features and the final decision-making process.

Because our work is a novel approach to differential optimization, it deserves additional work. For instance, explicitly formulating the gradient of the optimal variable for all steps of the search as a recurrent sequence could speed up computations. Furthermore, our tested architecture for the different *NN* parts, the number of dimension n_t for the Order Book, as well as the values of weighting parameters α, β and γ should be included in a more thorough hyper-parameter search. The next step in this direction will also be to include more sophisticated orders to the problem formulation. Block and Complex orders 2.1.2 constitute an additional difficulty to tackle because the problem has to be solved at daily granularity but are crucial for obtaining better price estimates. Additionally, integrating results of our previous Chapter 4 i.e. flow estimates in EUPHEMIA and solving it across the entire European Network will be a consistent step.

Nevertheless, our study highlights the benefits of the end-to-end differentiable optimization approach for Electricity Price Forecasting. By integrating the EUPHEMIA optimization task with order book prediction, we achieved improved accuracy in price forecasting, which has significant implications for the efficient management of electricity production and consumption. Our findings pave the way for further research and development in this area, including exploring other optimization algorithms and refining the prediction models to enhance the overall performance of EPF systems.

Chapter 6

Conclusion

Electricity price forecasts are necessary for many applications that implement the energy transition such as self-consumption or battery optimization, but also for generating profit on the market (1.2). Yet, the European electricity market is regulated by the Price-Fixing Algorithm EUPHEMIA that at the same time protects the price of electricity from speculation and makes its relationship to supply and demand intricated. Because of the unsteadiness of electricity, its prices are hard to estimate. Moreover, only the Price-Fixing Algorithm gives the true relationship between fundamentals, transfer capacities and the price, and this at the European scale (1.3).

In the literature, two families of approaches are in conflict while trying to improve both the correctness (how close are the forecasts to the real prices) and the consistency (how close the model is to the Price-Fixing Algorithm) of price forecasts (see Section 2.3). Expert models aim at replicating the Price-Fixing Algorithm but fail to adapt to real data, while Data-Driven methods occult it and directly link fundamentals to prices using black-box models. Mixing both Expert models and Data-Driven approaches by employing a differentiable optimization framework has never been tried for EPF, whereas it takes the best from both approaches: considering the Price-Fixing Algorithm and accounting for data volatility. The challenge of differentiable optimization lies in taking into account the *regret* (the error in optimal solutions) while training the Neural Network 2.4.

In this thesis, we have presented various ways of tackling the EPF problem. In Chapter 3, we focus on tuning Data-Driven models for the prediction of three well-known markets. Using our domain-knowledge and the feature contribution analysis tool Shap, we drew conclusions about our results. First, we questioned the current state of the art studies [Lago et al., 2018a] and [Lago et al., 2021] by noticing that the most accurate model for a given market was tied to the inherent market specificities (*SVR* for the French market that is based on steady nuclear plants, *DNN* for the German market, based on more volatile renewables energy). We also identified that using features from $d-2$, $d-3$ or $d-7$ for predicting prices of a day d was not beneficial, whereas using data from non-neighboring markets could significantly improve performances (3.4). We also explored the scenario where we predicted the prices of the three markets jointly. This resulted in a performance increase for only one market (Belgium), that was due to its access to the Swiss prices (3.5).

From these observations, we strove for jointly predicting the prices of all European markets in Chapter 4. To this aim, we first considered the flow estimation problem that we tackled by solving an optimization problem similar to EUPHEMIA prior to learning (4.3). The idea is to restore energy balance from the consumption and generation forecasts using the connection lines between zones. We employed several means of penalizing the deviation from the equilibrium, notably a linear and a quadratic cost. Then, we also modeled the network as a Graph where each market is a node to label (4.4). The flow estimates are used as edge attributes. The Graph structure is able to capture the topology of the network and can preserve market specificities. This translates as significantly more accurate price forecasts for the *GNN* model, compared to standard *CNN* or *DNN* models (4.5). Despite the *GNN* already implicitly considering cross-zonal flows via the training of message weights between nodes, we have observed in the contribution patterns that using the flow estimated with a QP problem could still improve performances, particularly in the central areas where the cross-border flows are more sophisticated (4.5.3).

Lastly in Chapter 5, we attacked the core of EPF that is the EUPHEMIA algorithm. We started by explicitly defining the *social welfare* of a linear order, that we extended to an entire Order Book (5.2). We have seen that the Day-Ahead Price could be obtained by solving the dual problem. In the aim of tracking the gradients of the price with respect to the Order Book, solved the dual problem by finding the point where its derivative reaches 0 using a dichotomy search (5.3). This block is then put on top of an Order Books forecasting network : the Day-Ahead Price is determined by solving EUPHEMIA using estimated Order Books. The result is an hybrid model, combining Data-Driven and Expert models approaches (5.4). We then analysed the impact of modifying the importance of each component in the final loss and have concluded that the differentiable optimization part could lead to significant improvement (5.5). However, pure differentiable optimization was still outperformed by mixed approaches, suggesting a deeper analysis of the model's hyperparameters. We have also seen that the model relies less on fundamental forecasts (consumption, generation) while increasing the weight of the differentiable optimization in the loss. This is explained because the true relationship between fundamentals and the price is captured by the Order Book, not by the point forecasts that are fed to the model (5.6).

6.1 Industrial Approach

The methods we have explored during this thesis can be adapted for use in an industrial context. The tool that we developed runs on a daily basis. Everyday before noon, it acquires the prediction data, forecasts the 24 daily prices of the next day, and stores the results in a database. In particular, we use the models described in Chapter 2 to predict the prices of France and Germany. Other companies use the forecasted prices of Germany in the scope of the Islander project. The forecasted prices of France are used for two activities. First, the hourly price curve is considered to optimize the use of a battery. The battery is charged when the price forecasts are low and discharged when high.

Second, the daily average price forecast is used by traders. The model is assessed in real

conditions using a *Paper Trade*. This means that the trader record the decision taken based on a forecasted price and a market value at a given time, as explained in Section 1.2.2, without actually bidding. When the real price is published, the Profit & Loss (P&L) is computed. As this measure requires the action of a trader, week-end days are excluded. Two versions of the model are used. The first runs at 8.30 am, and thus can't use the Swiss prices. The second runs at 11.30 am and integrates the Swiss prices. Two metrics are monitored for the last 30 days using a dashboard : the Daily Average Error is displayed in Figure 6.1, while the P&L is displayed in Figure 6.2. As of writing these lines, the average P&L of the model using Swiss price (11.30 am) indicates that traders could gain on average 20€/MWh every working day. Since players usually trade 25MWh, this represents a significant value that is very promising for a simple model.

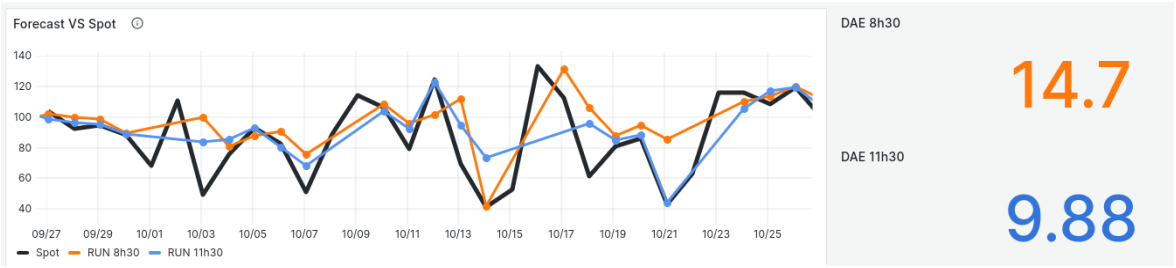


Figure 6.1 – Screenshot of the dashboard monitoring SVR model used for predicting the French prices, taken on the 26-10-2023. The left part displays the real price (in black), the forecast without Swiss prices (orange) and with Swiss Prices (blue). On the right, the DAE of the displayed period is computed.

Figure 6.2 – Screenshot of the dashboard monitoring the SVR model used for predicting the French prices, taken on the 26-10-2023. On the left, the daily P&L of the model without the Swiss prices is displayed in orange, and in blue for the model using the Swiss prices. On the right, 3 metrics are displayed for both models. First, the percentage of cases were the daily P&L is positive. Then, the total P&L over the 30 days period. Lastly, the daily average P&L over the 30 days.

As this requires additional work, we have not yet converted all our research into tools like this one we presented. Our first step will be to produce a relevant explanation along with the forecasted value. We believe that a force diagram, indicating which feature pushed the price forecast the most, could be a game-changing tool for trading on the day-ahead market. Using our approaches of Chapter 4 and Chapter 5 for day-to-day applications will be less straightforward as it requires handling multi-source data acquisition (35 zones for the graph and thousands of orders for the differentiable optimization). However, since our

experimental results were promising and the approaches innovative, adapting them will constitute our next significant task.

6.2 Future Works

Throughout this thesis, we have explored means of improving both the consistency and the accuracy of the price forecasts. We started from simple ML models for which we explained the predictions, to a differentiable optimization framework where the Price-Fixing Algorithm is explicitly solved and Order Book are modeled. However, this approach left a lot of room for improvement and we detail our possible future steps in this last section.

6.2.1 Task-specific forecasts

The first category of future directions lies in the set of possibilities offered by differentiable optimization. Indeed, we focused on incorporating the EUPHEMIA problem into a price forecasting network. This solution is suitable for task-agnostic price forecasts. However, most of the time, forecasts are used for a given application where lowering the price *MAE* is not the end objective. Similarly to Order Book errors that reverberates on the price forecasts, a small price forecast could drastically impact the end-task. For instance, setting-up a differentiable optimization approach for managing a battery, sounds promising. The neural network would forecast the prices but the end goal is to maximize the battery profit while ensuring that users can use their energy at a given time.
Similarly, trading on the Day-Ahead market can be viewed as a knapsack problem where players have a limited capacity (the traded volume is limited) and the goods they want to pick have an unknown value (the profit of a trade is unknown without the real price). We could consider optimizing the profit based on price estimates, with the price prediction network fit to maximize the profit.
Lastly, many other real-world problems would benefit from Differentiable Optimization methods. Although our approach presented in Chapter 5 is specific to EUPHEMIA, it could be adapted to other problems. If the Dual function of an Optimization problem admits a derivative that is monotonous, then our dichotomy search method can be employed.

6.2.2 Order Books embeddings

Our differentiable optimization approach relies on forecasting Order Books and using them to solve EUPHEMIA. Many steps of the forecasting task could be improved. First, the size of the embedding is constrained by the number of neurons in the hidden dimension of the network, and enforced by applying a shrinking operation (as defined in Section 5.4.2.3). In our experiment, we use a fixed value of 20 orders per book, whereas this number should be treated as a hyper-parameter to optimize.
Then, we scale order books using regular min-max scaler prior to learning, then apply batch normalization during training. This allows the neural network to fit its weights optimally and prevents over-fitting. However, this makes the embedding Order Books hard to interpret. Solving EUPHEMIA on the embedded Order Books should yield the scaled prices. In practice, this is not the case because we use an Order Book scaler independent of the price scaler. We think our model correct this by predicting biased Order Books and perhaps

this is why increasing the weight of Order Book forecasting error in the global loss does not leads to better results (5.5.4). Defining a suitable Order Book scaling operation would be an important step towards improving the forecasts, but also a very important tool for business practitioners that would benefit from having access to the Order Books forecasts. Doing so, they could graphically appreciate the price sensitivity to supply and demand shifts.
Lastly, we considered only LINEAR and STEP orders. In practice, there are other order types that adds complexity to EUPHEMIA but that could improve our price forecasts. BLOCK and COMPLEX orders span across several hours, that makes it mandatory to consider an entire day in a single optimization problem : the number of decision variables and constraints is multiplied by 24. On top of that, the acceptance ratio of a block order is a binary variable. This makes the problem MIQP thus much harder to solve, and even harder to differentiate.

6.2.3 Solving EUPHEMIA for several markets

Another step would be to combine our works on modeling the network as a graph and solving EUPHEMIA. Instead of forecasting the Day-Ahead Price of each node of the graph, we could forecast the associated Order Books. Then, two options would be available. The first and easier one would be to solve each Order Book separately using our differentiable dichotomy search. A more sophisticated option could consist in modifying the search to solve the problem for several zones at once. To this aim, we would have to combine the optimization problems of Chapter 4 with our simplification of EUPHEMIA constructed in Chapter 5. This would only increase the problem dimensions (number of constraints and decision variables), not its complexity since the cross-markets flows are considered linearly in the energy balance constraint:

$$\max_{\substack{A_{z,i}\in[0,1]\\ F_{z,z'}\geqslant 0}} \sum_{z} \sum_{i\in OB_z} \left(-\frac{1}{2} A_{z,i}^2 V_{z,i} P_{z,i} - A_{z,i} V_{z,i} Po_{z,i} \right) + \sum_{z,z'} T_{z,z'} F_{z,z'} \tag{6.1}$$

$$\text{u.c.} \quad \sum_{i\in OB_z} A_{z,i} V_{z,i} + \sum_{z'} F_{z',z} - \sum_{z'} F_{z,z'} = 0, \ \forall z \tag{6.2}$$

$$F_{z,z'} \leqslant \Gamma_{z,z'}, \ \forall z, z' \tag{6.3}$$

The *social welfare* of Equation (6.1) has to be accumulated on all considered zones, and the congestion rents taken into account. The *energy balance* of Equation (6.2) considers accepted orders and cross-zonal flows. Lastly, Equation (6.3) ensures that chosen flows respect the network constraints.

The EPF task is a very dense subject with multiple possible strategies for tackling it. The diversity of applications, the complexity of the Price-Fixing Algorithm and the abundance of potential price drivers contribute towards making EPF both difficult and fascinating. Our work surely paved the way for novel ways of tackling the problem, and we will be excited to keep working on those promising approaches.

Appendix A

Appendix - Notation Tables

General notations	
Symbol	**Variable**
Y	Day-Ahead Price
$\overline{Y}^{(d)}$	Average daily price
X	Predictive dataset
x	Data instance
$.^{(d,h)}$	Timestamp (day, hour)
$\widehat{\cdot}$	Forecasted value
$\widetilde{\cdot}$	Estimated value
$\overline{\cdot}$	Averaged value
$.^{\star}$	Optimal Value
z	Zone or market
l	Lag days of a variable
f	Feature
L	Loss function

Table A.1 – Notations used throughout this thesis. Most of them are defined in 2.1.

Feature notations	
Symbol	**Variable**
n	Number of data instances
n_z	Number of zones
n_f	Number of features
n_d	Number of data instances
n_c	Number of columns in a data instance
n_f^l	Number of lags for a Feature
P	Current Price
C	Consumption forecast
R	Renewables forecast
G	Generation forecast
V	Maximal Generation capacity
E	Required programmable generation
F	Cross-market Flows
Γ	Available Transfer Capacity
T	Congestion Rents

Table A.2 – Noatation used for the features, introduced in Chapter 4.2

Shap Values notations	
Symbol	**Variable**
$\mathbb{E}$	Expectation function
$\mathcal{F}$	Set of all features
S	Coallition of features
ϕ	Shapley value
m	Predictive model
$\widetilde{m}_x(S)$	Marginalised preditction function over S
g_x	Additive explanatory model
z	Binary encoding of a coalition
$\bar{\Phi}^{(h)}_{f,l,h'}$	Average weight of each column for predicting a label
$\bar{\Phi}_f$	Average contribution of feature f
$\bar{\Phi}_l$	Average contribution of lag l
$\bar{\Phi}^{(d)}_f$	Daily Average Unit Contribution
h'	Hour of a column

Table A.3 – Notations used for Shap Values, defined in Chapter 3.3.6 and 3.4.2

EUPHEMIA notations	
Symbol	**Variable**
OB	Order Book
Φ	Order Book domain
$\mathcal{E}$	EUPHEMIA algorithm
A	Acceptance ratio of an order
P	Price limit of a step Order
$[Po, Po + P]$	Price range of a Linear Order
V	Volume of an Order
v	Sold volume of an Order
λ	Dual Variable associated to the *energy balance* constraint
M, K	Dual Variable associated to the inequality constraints
N	Number of orders
$\mathcal{L}$	Lagrangian function
$\mathcal{D}$	Dual function
$H(x)$	Heaviside function
$\delta(x)$	Dirac function
$\sigma(x)$	Sigmoid function
x_i	$V_i(\lambda - Po_i)$
y_i	$V_i(\lambda - Po_i - P_i)$

Table A.4 – Notation for solving EUPHEMIA. Some variables are introduced in Chapter 2.1.2 and extended in Chapter 5.2.

Differentiable Optimization notations	
Symbol	**Variable**
L_{DNN}	Direct forecast price loss
L_{OB}	Direct forecast Order Book loss
L_{DO}	Differentiable optimization price loss
α	Weight of L_{DNN} is the total loss
β	Weight of L_{DO} is the total loss
γ	Weight of L_{OB} is the total loss
Y_{DNN}	Direct price forecasts
Y_{DO}	Differentiable optimization price forecasts
NN_Y	Direct price forecasts layers
NN_{OB}	Order Book forecasts layers
NN	Shared layers
DNN_Y	Direct price forecasts model
$DNN_{Y,OB}$	Direct price + Order Book forecast model
DO	Differentiable Optimization model

Table A.5 – Notations used for the differentiable optimization model of Chapter 5.

Appendix B

Appendix - Chapter 1

Country Code	Description
AT	Austria
BE	Belgium
BG	Bulgaria
CH	Switzerland
CZ	Czechia
DE	Germany
DK1	Western Denmark
DK2	Eastern Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GB	United Kingdom
GR	Greece
HR	Croatia
HU	Hungary
IE	Ireland
IT-CNOR	Northern-Central Italy
IT-CSUD	Southern-Central Italy
IT-NORD	Northern Italy

Country Code	Description
IT-SARD	Italy - Sardinia
IT-SICI	Italy - Sicilia
IT-SUD	Southern Italy
LT	Lithuania
LV	Latvia
NO1	Norway - Oslo
NO2	Norway - Kristiansand
NO3	Norway - Trondheim
NO4	Norway - Tromsø
NO5	Norway - Bergen
NL	the Netherlands
PL	Poland
PT	Portugal
RO	Romania
SE1	Sweden - Luleå
SE2	Sweden - Sundsvall
SE3	Sweden - Stockholm
SE4	Sweden - Malmö
SI	Slovenia
SK	Slovakia

Table B.1 – Conversion table for associating zonal codes to physical regions.

Appendix C

Appendix - Chapter 3

Random Forest				
Parameter	**Search Range**	**Dataset**	**Market**	**Value in best configuration**
min_sample_split	[2, 50]	SOTA	FR	4
			DE	2
			BE	3
		Enriched	FR	21
			DE	2
			BE	3
n_estimators	[10, 1500]	SOTA	FR	91
			DE	1228
			BE	502
		Enriched	FR	461
			DE	359
			BE	416

Table C.1 – Hyper-parameters grid and best configuration for the *RF* model used in Chapter 3.4.

SVRChain				
Parameter	**Search Range**	**Dataset**	**Market**	**Value in best configuration**
C	$[10^{-9}, 10^{9}]$	SOTA	FR	28.688861
			DE	9666.061474
			BE	2907.102329
		Enriched	FR	28.688861
			DE	268354179.083
			BE	5734.823533
gamma	$[10^{-9}, 10^{9}]$	SOTA	FR	0.000036
			DE	0.0000001583
			BE	0.0000001385
		Enriched	FR	0.000036
			DE	0.000438
			BE	0.0000000615

Table C.2 – Hyper-parameters grid and best configuration for the *SVRChain* model used in Chapter 3.4.

SVRMulti				
Parameter	**Search Range**	**Dataset**	**Market**	**Value in best configuration**
C	$[10^{-9}, 10^{9}]$	SOTA	FR	18.770937
			DE	573.647277
			BE	1000.08756
		Enriched	FR	434.409029
			DE	10483.318948
			BE	17.81966
gamma	$[10^{-9}, 10^{9}]$	SOTA	FR	0.000131
			DE	0.000003
			BE	0.000001
		Enriched	FR	0.000005
			DE	0.0000000916
			BE	0.0000007897

Table C.3 – Hyper-parameters grid and best configuration for the *SVRMulti* model used in Chapter 3.4.

DNN				
Parameter	**Search Range**	**Dataset**	**Market**	**Value in best configuration**
batch_norm	True or False	SOTA	FR	True
			DE	True
			BE	True
		Enriched	FR	True
			DE	False
			BE	False
batch_size Enriched	[10, n]	SOTA	FR	744
			DE	367
			BE	1200
			FR	153
			DE	736
			BE	17
dropout	[0, 0.5]	SOTA	FR	0.076741
			DE	0.180403
			BE	0.067465
		Enriched	FR	0.160375
			DE	0.0
			BE	0.430954
hidden_layers	[1, 4]	SOTA	FR	1
			DE	1
			BE	1
		Enriched	FR	1
			DE	2
			BE	1
neurons_per_layer	[10, 500]	SOTA	FR	(157,)
			DE	(444,)
			BE	(124,)
		Enriched	FR	(440,)
			DE	(254, 231)
			BE	(370,)

Table C.4 – Hyper-parameters grid and best configuration for the *DNN* model used in Chapter 3.4. n is the number of data instances in the training set

CNN				
Parameter	**Search Range**	**Dataset**	**Market**	**Value in best configuration**
batch_norm	True or False	SOTA	FR	False
			DE	False
			BE	False
		Enriched	FR	False
			DE	False
			BE	False
batch_size	[10, *n*]	SOTA	FR	360
			DE	819
			BE	235
		Enriched	FR	*n*
			DE	468
			BE	17
dropout	[0, 0.5]	SOTA	FR	0.073378
			DE	0.21907
			BE	0.057622
		Enriched	FR	0.1
			DE	0.434814
			BE	0.0
fully_connected_layers	[0, 2]	SOTA	FR	1
			DE	0
			BE	0
		Enriched	FR	1
			DE	0
			BE	0
neurons_per_layer	[10, 500]	SOTA	FR	(470,)
			DE	()
			BE	()
		Enriched	FR	(50)
			DE	()
			BE	()
convolutional_layers	[1, 4	SOTA	FR	1
			DE	1
			BE	1
		Enriched	FR	2
			DE	1
			BE	1
filters	[3, 25]	SOTA	FR	20
			DE	20
			BE	4
		Enriched	FR	(8, 6)
			DE	7
			BE	19
dilation_rate	([1, 5], [1, 5])	SOTA	FR	(4, 4)
			DE	(2, 1)
			BE	(4, 2)
		Enriched	FR	(4, 4), (2, 2)
			DE	(4, 3)
			BE	(1, 2)
kernel_size	([3, 11], [3, 25])	SOTA	FR	(10, 15)
			DE	(5, 21)
			BE	(3, 14)
		Enriched	FR	(9, 9), (9, 9)
			DE	(9, 16)
			BE	(10, 13)
pooling_size	([0, 4], [0, 4])	SOTA	FR	(0, 0)
			DE	(0, 0)
			BE	(2, 3)
		Enriched	FR	(2, 2), (2, 2)
			DE	(0, 0)
			BE	(0, 0)
strides	([0, 4], [0, 4])	SOTA	FR	(0, 0)
			DE	(0, 0)
			BE	(2, 3)
		Enriched	FR	(1, 1), (1, 1)
			DE	(0, 0)
			BE	(0, 0)

Table C.5 – Hyper-parameters grid and best configuration for the *CNN* model used in Chapter 3.4.

Appendix D

Appendix - Chapter 4

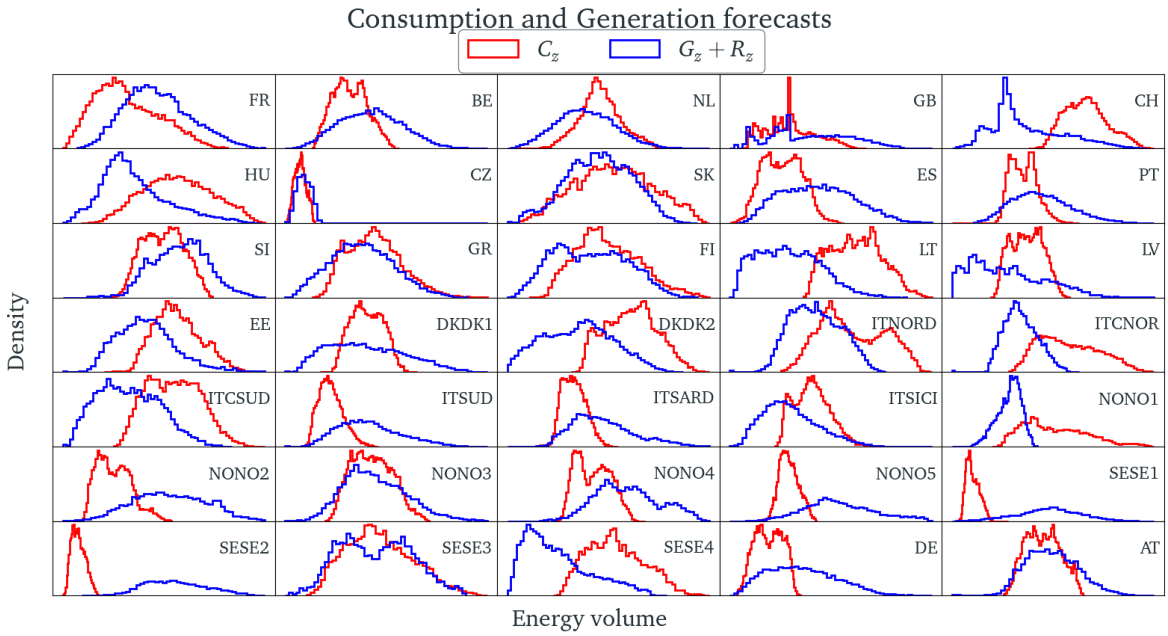


Figure D.1 – (RED) consumption forecasts histogram (BLUE) Production (programmable + renewable) forecasts for the period 2020-2021. Data comes from the entsoe website.

DNN		
Parameter	**Search Range**	**Value in best configuration**
batch_norm	True or False	False
batch_size	[10, n]	78
dropout	0, 0.5	0.089953
hidden_layers	[1, 4]	1
neurons_per_layer	[10, 1000]	(732,)

Table D.1 – Hyper-parameters grid and best configuration for the *DNN* model used in Chapter 4.5. n is the number of data instances in the training dataset

CNN		
Parameter	**Search Range**	**Value in best configuration**
batch_norm	True or False	False
batch_size	[10, n]	300
dropout	0, 0.5	0
dense_hidden_layers	[1, 4]	0
neurons_per_layer	[10, 1000]	()
convolutional_layers	[]	3
filters	[3, 25]	(24, 12, 6)
dilation_rate	([1, 8] [1, 6])	([1, 1], [1, 1], [1, 1])
kernel_size	([1, 3], [1, 2])	([10, 3], [5, 3], [3, 2])
pooling_size	([4, 12], (2, 6))	([4, 1], [4, 2], [3, 2])
strides	([4, 16], [2, 8])	([4, 1], [4, 2], [3, 2])

Table D.2 – Hyper-parameters grid and best configuration for the *CNN* model used in Chapter 4.5. n is the number of data instances in the training dataset

GNN		
Parameter	**Search Range**	**Value in best configuration**
batch_norm	True or False	False
batch_size	[10, n]	n
dropout	0, 0.5	0.0
fully_connected_layers	[0, 2]	1
neurons_per_layer	[24, 96]	(81,)
convolution_layers	[1, 4]	2
hidden_channels	[24, 96]	(32, 31)
heads	[1, 20]	15

Table D.3 – Hyper-parameters grid and best configuration for the *GNN* model used in Chapter 4.5. n is the number of data instances in the training dataset.

Appendix E

Appendix - Chapter 5

E.1 $\lambda^\star$ is the Day-Ahead price : Gaphical insights

In this section, we explain why the optimal solution of EUPHEMIA dual problem is the Day-Ahead price. We give a proof for a very simple case that can be extended to more sophisticated examples. We consider only 1 zone on 1 hour. The Order Book only contain one linear supply order defined by points (V_{S1}, P_{S1}) $(V_{S1} + V_S, P_{S1} + P_S)$ and one linear demand order (V_{D1}, P_{D1}) $(V_{D1} + V_D, P_{D1} + P_D)$. Graphically, finding the optimal price is rather simple since it's the y-coordinate of the segment's intersection, as in displayed in Figure E.1. Mathematicaly, this translates as:

CURVE EQUATIONS:

$$\begin{cases} y_D = P_{D1} + \frac{P_D}{V_D}(x - V_{D1}) \\ y_S = P_{S1} + \frac{P_S}{V_S}(x - V_{S1}) \end{cases}$$

INTERSECTION:

$$\begin{aligned} & y_D = y_s \\ \Leftrightarrow \quad & P_{D1} + \frac{P_D}{V_D}(x^* - V_{D1}) = P_{S1} + \frac{P_S}{V_S}(x^* - V_{S1}) \\ \Leftrightarrow \quad & x^*(\frac{P_D}{V_D} - \frac{P_S}{V_S}) = P_{S1} - P_{D1} + \frac{P_D V_{D1}}{V_D} - \frac{P_S V_{S1}}{V_S} \\ \Leftrightarrow \quad & x^* \frac{P_D V_S - P_S V_D}{V_D V_S} = \frac{V_D V_S (P_{S1} - P_{D1}) + V_S P_D V_{D1} - V_D P_S V_{S1}}{V_D V_S} \\ \Leftrightarrow \quad & x^* = \frac{V_D V_S (P_{S1} - P_{D1}) + V_S P_D V_{D1} - V_D P_S V_{S1}}{V_S P_D - V_D P_S} \end{aligned}$$

PROJECTION:

$$
\begin{aligned}
& y^* = P_{D1} + \frac{P_D}{V_D}(x^* - V_{D1}) \\
\Leftrightarrow \quad & y^* = P_{D1} + \frac{P_D}{V_D}\left(\frac{V_D V_S (P_{S1} - P_{D1}) + V_S P_D V_{D1} - V_D P_S V_{S1}}{V_S P_D - V_D P_S} - V_{D1}\right) \\
\Leftrightarrow \quad & y^* = P_{D1} + \frac{P_D}{V_D}\left(\frac{V_D V_S (P_{S1} - P_{D1}) + V_S P_D V_{D1} - V_D P_S V_{S1}}{V_S P_D - V_D P_S} - \frac{V_S P_D V_{D1} - V_D P_S V_{D1}}{V_S P_D - V_D P_S}\right) \\
\Leftrightarrow \quad & y^* = \frac{V_S P_D P_{D1} - V_D P_S P_{D1}}{V_S P_D - V_D P_S} + \frac{V_S P_D (P_{S1} - P_{D1}) - P_D P_S V_{S1} + P_D P_S V_{D1}}{V_S P_D - V_D P_S} \\
\Leftrightarrow \quad & y^* = \frac{V_S P_D P_{S1} - V_D P_S P_{D1} + P_S P_D (V_{D1} - V_{S1})}{V_S P_D - V_D P_S}
\end{aligned}
$$

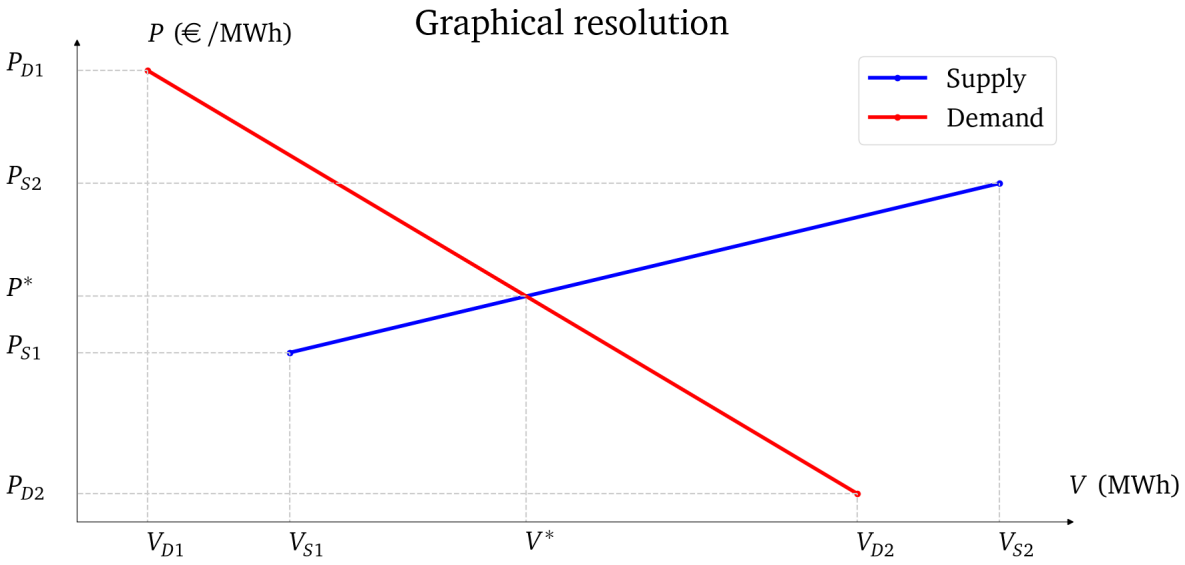


Figure E.1 – Finding the optimal price : Graphical Solution

Now let's write EUPHEMIA with only 1 supply order and 1 demand order:

$$
\begin{aligned}
\max_{a_S, a_D} \quad & \frac{1}{2} a_D^2 V_D P_D + a_D V_D P_{D1} - \frac{1}{2} a_S^2 V_S P_S - a_S V_S P_{S1} \\
\text{u.c.} \quad & V_{S1} + a_S V_S - V_{D1} - a_D V_D = 0
\end{aligned}
$$

Thanks to the constraint, it would be easy to express a_S function of a_D. However, writing down the dual problem will be usefull later. We start by computing the Lagrangian:

$$
\mathcal{L}(a_S, a_D, \lambda) = \frac{1}{2} a_D^2 V_D P_D + a_D V_D P_{D1} - \frac{1}{2} a_S^2 V_S P_S - a_S V_S P_{S1} + \lambda (V_{S1} + a_S V_S - V_{D1} - a_D V_D)
$$

The dual function $D(\lambda) = \min_{a_D, a_S} L(a_S, a_D, \lambda)$ is found when:

$$\begin{aligned}
&\leftrightarrow \nabla_{(a_S, a_D)} L = 0 \\
&\qquad \leftrightarrow \begin{cases} \frac{\partial \mathcal{L}}{\partial a_S} = 0 \\ \frac{\partial \mathcal{L}}{\partial a_D} = 0 \end{cases} \\
&\qquad \leftrightarrow \begin{cases} -a_S V_S P_S - V_S P_{S1} + \lambda V_S = 0 \\ a_D V_D P_D + V_D P_{D1} - \lambda V_D = 0 \end{cases} \\
&\qquad \leftrightarrow \begin{cases} a_S = \frac{\lambda - P_{S1}}{P_S} \\ a_D = \frac{\lambda - P_{D1}}{P_D} \end{cases}
\end{aligned}$$

We observe that $\lambda = P_{S1} + a_S P_S = P_{D1} + a_D P_D$. λ is the Day-Ahead price! Let's check it by injecting those expressions in the Lagrangian:

$$\begin{aligned}
\mathcal{L}(a_S, a_D, \lambda) &= \frac{1}{2} a_D^2 V_D P_D + a_D V_D P_{D1} - \frac{1}{2} a_S^2 V_S P_S - a_S V_S P_{S1} + \lambda (V_{S1} + a_S V_S - V_{D1} - a_D V_D) \\
\mathcal{D}(\lambda) &= \frac{V_D P_D}{2} \left(\frac{\lambda^2}{P_D^2} - \frac{2\lambda P_{D1}}{P_D^2} + \frac{P_{D1}^2}{P_D^2} \right) + \frac{\lambda V_D P_{D1}}{P_D} - \frac{V_D}{P_D} P_{D1}^2 \\
&\quad - \frac{V_S P_S}{2} \left(\frac{\lambda^2}{P_S^2} - \frac{2\lambda P_{S1}}{P_S^2} + \frac{P_{S1}^2}{P_S^2} \right) - \frac{\lambda V_S P_{S1}}{P_S} + \frac{V_S}{P_S} P_{S1}^2 \\
&\quad - \lambda V_{D1} + \lambda V_{S1} - \lambda \left(\frac{\lambda V_D - V_D P_{D1}}{P_D} \right) + \lambda \left(\frac{\lambda V_S - V_S P_{S1}}{P_S} \right) \\
&= \lambda^2 \left[\frac{1}{2} \frac{V_D}{P_D} - \frac{1}{2} \frac{V_S}{P_S} + \frac{V_S}{P_S} - \frac{V_D}{P_D} \right] \\
&\quad + \lambda \left[- \frac{V_D P_{D1}}{P_D} + \frac{V_D P_{D1}}{P_D} + \frac{V_S P_{S1}}{P_S} - \frac{V_S P_{S1}}{P_S} - V_{D1} + V_{S1} + \frac{V_D P_{D1}}{P_D} - \frac{V_S P_{S1}}{P_S} \right] \\
&\quad + \frac{1}{2} \frac{V_D P_{D1}^2}{P_D} - \frac{V_D}{P_D} P_{D1}^2 - \frac{1}{2} \frac{V_S P_{S1}^2}{P_S} + \frac{V_S}{P_S} P_{S1}^2 \\
\mathcal{D}(\lambda) &= \frac{\lambda^2}{2} \left[\frac{V_S}{P_S} - \frac{V_D}{P_D} \right] + \lambda \left[V_{S1} - V_{D1} + \frac{V_D P_{D1}}{P_D} - \frac{V_S P_{S1}}{P_S} \right] + \frac{1}{2} \frac{V_S P_{S1}^2}{P_S} - \frac{1}{2} \frac{V_D P_{D1}^2}{P_D}
\end{aligned}$$

with the constant K independant of λ. We are then able to express the dual problem as a quadratic, unconstrained problem:

$$\min_{\lambda \in \mathbb{R}} \ \frac{\lambda^2}{2} \left[\frac{V_S}{P_S} - \frac{V_D}{P_D} \right] + \lambda \left[V_{S1} - V_{D1} + \frac{V_D P_{D1}}{P_D} - \frac{V_S P_{S1}}{P_S} \right] + \frac{1}{2} \frac{V_S P_{S1}^2}{P_S} - \frac{1}{2} \frac{V_D P_{D1}^2}{P_D}$$

The dual problem is solved when the function $D(\lambda)$ is minimized:

$$\begin{aligned}
&\nabla_\lambda \mathcal{D} = 0 \\
&\leftrightarrow 0 = \lambda\left(\frac{V_S}{P_S} - \frac{V_D}{P_D}\right) + V_{S1} - V_{D1} + \frac{V_D P_{D1}}{P_D} - \frac{V_S P_{S1}}{P_S} \\
&\quad = \lambda\frac{V_S P_D - V_D P_S}{P_S P_D} + \frac{P_S P_D V_{S1} - P_S P_D V_{D1} + V_D P_S P_{D1} - V_S P_D P_{S1}}{P_S P_D} \\
&\quad \lambda = \frac{V_S P_D P_{S1} - V_D P_S P_{D1} + P_S P_D (V_{D1} - V_{S1})}{V_S P_D - V_D P_S} \\
&\quad \lambda = P^*
\end{aligned}$$

The optimal variable of the dual problem is the day-ahead price. Let's check if the solution of the primal problem aligns with our previous findings:

$$\begin{aligned}
&\begin{cases} a_D = \frac{\lambda - P_{D1}}{P_D} \\ a_S = \frac{\lambda - P_{S1}}{P_S} \end{cases} \\
\leftrightarrow &\begin{cases} a_D P_D = \frac{V_S P_D P_{S1} - V_D P_S P_{D1} + P_D P_S (V_{D1} - V_{S1})}{V_S P_D - V_D P_S} - \frac{V_S P_D P_{D1} - V_D P_S P_{D1}}{V_S P_D - V_D P_S} \\ a_S P_S = \frac{V_S P_D P_{S1} - V_D P_S P_{D1} + P_D P_S (V_{D1} - V_{S1})}{V_S P_D - V_D P_S} - \frac{V_S P_D P_{S1} - V_D P_S P_{S1}}{V_S P_D - V_D P_S} \end{cases} \\
\leftrightarrow &\begin{cases} a_D P_D = \frac{V_S P_D (P_{S1} - P_{D1}) + P_D P_S (V_{D1} - V_{S1})}{V_S P_D - V_D P_S} \\ a_S P_S = \frac{V_D P_S (P_{S1} - P_{D1}) + P_D P_S (V_{D1} - V_{S1})}{V_S P_D - V_D P_S} \end{cases} \\
\leftrightarrow &\begin{cases} a_D = \frac{V_S (P_{S1} - P_{D1}) + P_S (V_{D1} - V_{S1})}{V_S P_D - V_D P_S} \\ a_S = \frac{V_D (P_{S1} - P_{D1}) + P_D (V_{D1} - V_{S1})}{V_S P_D - V_D P_S} \end{cases}
\end{aligned}$$

and we can then find $V*$:

$$\begin{aligned}
V^* &= V_{D1} + a_D V_D \\
&= \frac{V_S P_D V_{D1} - V_D P_S V_{D1} + V_S V_D (P_{S1} - P_{D1}) + V_D P_S (V_{D1} - V_{S1})}{V_S P_D - V_D P_S} \\
V^* &= \frac{V_D V_S (P_{S1} - P_{D1}) + V_S P_D V_{D1} - V_D P_S V_{S1}}{V_S P_D - V_D P_S}
\end{aligned}$$

we check with the supply side:

$$\begin{aligned}
V^* &= V_{S1} + a_S V_S \\
&= \frac{V_S P_D V_{S1} - V_D P_S V_{S1} + V_S V_D (P_{S1} - P_{D1}) + V_S P_D (V_{D1} - V_{S1})}{V_S P_D - V_D P_S} \\
V^* &= \frac{V_S V_D (P_{S1} - P_{D1}) + V_S P_D V_{D1} - V_D P_S V_{S1}}{V_S P_D - V_D P_S}
\end{aligned}$$

which is the same expression than the graphical resolution.

E.2 Expressing $\mathcal{D}(\lambda)$

The Lagrangian of the problem can be written as:

$$\mathcal{L}(A,\lambda,M,K) = \sum_{i\in\, OB} \mathcal{L}_i(A,\lambda,M,K) \text{ with}$$

$$\mathcal{L}_i(A,\lambda,M,K) = -\frac{1}{2}A_i^2 V_i P_i - A_i V_i Po_i + \lambda A_i V_i + A_i(M_i - K_i) + K_i$$

Then $\forall i = 1:N$, the STATIONARITY condition yields:

$$\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial A_i} = 0 \\
\leftrightarrow\ &\frac{\partial \sum_j \mathcal{L}_j}{\partial A_i} = 0 \\
\leftrightarrow\ &\frac{\partial \mathcal{L}_i}{\partial A_i} = 0 \ \text{ because } \ \frac{\partial \mathcal{L}_j}{\partial A_i} = 0 \ \ \forall i \neq j \\
\leftrightarrow\ &0 = -A_i V_i P_i - V_i Po_i + \lambda V_i + M_i - K_i \\
\leftrightarrow\ &A_i = \frac{\lambda - Po_i}{P_i} + \frac{M_i - K_i}{V_i P_i}
\end{aligned}$$

Rather than re-injecting in the Lagrangian, we use the COMPLEMENTARITY and PRIMAL ADMISSIBILITY conditions $\forall i \in [1,N]$:

$$\begin{cases}
-A_i & \leqslant 0 \\
A_i - 1 & \leqslant 0 \\
-M_i A_i & = 0 \\
K_i(A_i - 1) & = 0 \\
A_i & = \frac{\lambda - Po_i}{P_i} + \frac{M_i - K_i}{V_i P_i}
\end{cases}$$

That leads to only 3 possible scenarios:

Case	$\mathbf{K_i}$	$\mathbf{M_i}$	$\mathbf{A_i}$
(1)	0	$V_i\,(Po_i - \lambda)$	0
(2)	$V_i\,(\lambda - P_i - Po_i)$	0	1
(3)	0	0	$\frac{\lambda - Po_i}{P_i}$

Since the DUAL ADMISSIBILITY condition yields that $M_i \geqslant 0$ and $K_i \geqslant 0\ \forall i \in [1,N]$, case (1) occurs if and only if $V_i\,(Po_i - \lambda) \geqslant 0$ and case (2) if $V_i\,(\lambda - P_i - Po_i) \geqslant 0$. We can then compute the value of $\mathcal{L}_i$ for the three possible scenarios using:

$$\mathcal{L}_i(A,\lambda,M,K) = -\frac{1}{2}A_i^2 V_i P_i - A_i V_i Po_i + \lambda A_i V_i + A_i(M_i - K_i) + K_i$$

$$\begin{cases}
(1)\ \mathcal{L}_i = 0 \\
(2)\ \mathcal{L}_i = V_i(\lambda - \frac{P_i}{2} - Po_i) \\
(3)\ \mathcal{L}_i = \frac{V_i}{2P_i}(\lambda - Po_i)^2
\end{cases}$$

Lastly, we can express the dual function as a sum of piecewise functions:

$$\mathcal{D}(\lambda) = \sum_{i \in OB} \mathcal{D}_i(\lambda)$$

$$\mathcal{D}_i(\lambda) = \begin{cases} (1)\ 0 \text{ if } V_i(Po_i - \lambda) > 0 \\ (2)\ V_i(\lambda - \frac{P_i}{2} - Po_i) \text{ if } V_i(\lambda - P_i - Po_i)) > 0 \\ (3)\ \frac{V_i}{2P_i}(\lambda - Po_i)^2 \text{ if } \lambda \in [Po_i, Po_i + P_i] \end{cases}$$

And the dual problem is:

$$\min_{\lambda} \sum_i D_i(\lambda)$$

E.3 Computing $\mathcal{D}'(\lambda)$

To find the minimum of $\mathcal{D}(\lambda)$, we have to derive it:

$$\frac{\partial \mathcal{D}}{\partial \lambda} = \frac{\partial \sum_i \mathcal{D}_i(\lambda)}{\partial \lambda} = \sum_i \frac{\partial \mathcal{D}_i(\lambda)}{\partial \lambda}$$

Each segment of $\mathcal{D}_i$ is differentiable or equals 0. We have to compute the differential at the critical points, which are Po_i and $Po_i + P_i$. These are the points where the function can switch segment while slightly moving along λ. To this aim, we compute:

$$\lim_{h \mapsto 0^-} \frac{\mathcal{D}_i(\lambda + h) - \mathcal{D}_i(\lambda)}{h} \quad \text{and} \quad \lim_{h \mapsto 0^+} \frac{\mathcal{D}_i(\lambda + h) - \mathcal{D}_i(\lambda)}{h}$$

Let's start by considering only supply orders. For $(\lambda = Po_i, h \mapsto 0^+)$ and $(\lambda = Po_i + P_i, h \mapsto 0^-)$, the function stays in the first case while moving $h : \lambda + h \in [Po_i, Po_i + P_i]$. We compute:

$$\begin{aligned} \mathcal{D}_i(\lambda + h) &= \frac{V_i}{2P_i}(\lambda + h - Po_i)^2 \\ &= \lambda^2 \frac{V_i}{2P_i} + h^2 \frac{V_i}{2P_i} + Po_i^2 \frac{V_i}{2P_i} + \frac{\lambda h V_i}{P_i} - \frac{\lambda V_i Po_i}{P_i} - \frac{h V_i Po_i}{P_i} \\ &= \mathcal{D}_i(\lambda) + h^2 \frac{V_i}{2P_i} + \frac{\lambda h V_i}{P_i} - \frac{h V_i Po_i}{P_i} \\ \frac{\mathcal{D}_i(\lambda + h) - \mathcal{D}_i(\lambda)}{h} &= h \frac{V_i}{2P_i} + \frac{\lambda V_i}{P_i} - \frac{V_i Po_i}{P_i} \\ &= \frac{V_i}{P_i}(\frac{h}{2} + \lambda - Po_i) \end{aligned}$$

For the point $\lambda = Po_i$, we have $\mathcal{D}_i(Po_i) = 0$.

$$\begin{aligned} \lim_{h \mapsto 0^+} \frac{\mathcal{D}_i(\lambda + h) - \mathcal{D}_i(\lambda)}{h} &= \lim_{h \mapsto 0^+} \frac{2V_i}{P_i}(\frac{h}{2} + \lambda - Po_i) &= 0 \\ \lim_{h \mapsto 0^-} \frac{\mathcal{D}_i(\lambda + h) - \mathcal{D}_i(\lambda)}{h} &= \lim_{h \mapsto 0^-} \frac{0 - \mathcal{D}_i(Po_i)}{h} &= 0 \end{aligned}$$

So $\mathcal{D}_i(\lambda)$ has a derivative at $\lambda = Po_i$ and $\mathcal{D}'_i(Po_i) = 0$. Now for $\lambda = Po_i + P_i$:

$$\begin{aligned}
\mathcal{D}_i(Po_i + P_i) &= \frac{V_i}{2P_i}(Po_i + P_i - Po_i)^2 \\
&= \frac{V_i P_i}{2} \\
\lim_{h \mapsto 0^-} \frac{\mathcal{D}_i(\lambda + h) - \mathcal{D}_i(\lambda)}{h} &= \lim_{h \mapsto 0^+} \frac{V_i}{P_i}(\frac{h}{2} + \lambda - Po_i) && = V_i \\
\lim_{h \mapsto 0^+} \frac{\mathcal{D}_i(\lambda + h) - \mathcal{D}_i(\lambda)}{h} &= \frac{V_i(\lambda + h - \frac{P_i}{2} - Po_i) - \mathcal{D}_i(Po_i + P_i)}{h} \\
&= \frac{V_i(\lambda + h - \frac{P_i}{2} - Po_i) - \frac{V_i P_i}{2}}{h} && = V_i
\end{aligned}$$

So $\mathcal{D}_i(\lambda)$ has a derivative at $\lambda = Po_i + P_i$ and $\mathcal{D}'_i(Po_i + P_i) = V_i$. We apply the same reasonning for the demand orders. Computations are exactly the same by inverting Po_i and $Po_i + P_i$. We obtain that the derivative of $\mathcal{D}_i(\lambda)$ $\forall i \in [1, N]$:

$$\mathcal{D}'_i(\lambda) = \begin{cases} (1)\ 0 & \text{if } V_i(Po_i - \lambda) > 0 \\ (2)\ V_i & \text{if } V_i(\lambda - P_i - Po_i) > 0 \\ (3)\ \frac{V_i}{P_i}(\lambda - Po_i) & \text{if } \lambda \in [Po_i, Po_i + P_i] \end{cases}$$

We put the 2 points Po_i and $Po_i + P_i$ into the first case because the expression $\frac{V_i}{P_i}(\lambda - Po_i)$ matches our results.

E.4 Writing $\mathcal{D}'(\lambda)$ using Heaviside

We propose to rewrite $\mathcal{D}'(\lambda)$ as a sum of heaviside functions. Case (1) adds nothing to the sum, and case (2) is easy to write:

$$\begin{aligned}
&(2)\ V_i \ \text{ if } \ V_i(\lambda - P_i - Po_i) > 0 \\
&\quad \leftrightarrow V_i H(y_i) \ \text{ with } \ y_i = V_i(\lambda - Po_i - P_i)
\end{aligned}$$

For case (1), we have to use 2 heaviside functions:

$$\begin{aligned}
&(3)\ \frac{V_i}{P_i}(\lambda - Po_i) \text{ if } \lambda \in [Po_i, Po_i + P_i] \\
&\qquad \leftrightarrow \frac{x_i}{P_i}\left[H(x_i) - H(y_i)\right] \ \text{ with } \ x_i = V_i(\lambda - Po_i)
\end{aligned}$$

Indeed, $\frac{x_i}{P_i}H(x_i)$ yields the right value on the interval $[Po_i, Po_i + P_i]$ but also when $V_i(\lambda - Po_i - P_i) > 0$! We thus have to remove the same quantity on the inverval $V_i(\lambda - Po_i - P_i) > 0$. The dual function becomes:

$$\mathcal{D}'(x, y) = \sum \frac{x_i H(x_i) - y_i H(y_i)}{P_i}$$

This notation is notation will help us taking gradients of D' for the rest of the computations.

E.5 Proof that $\mathcal{D}'(\lambda)$ is strictly increasing

We start by proving that $\mathcal{D}'(\lambda)$ is strictly increasing, noting that the derivative of the Heaviside function is the Dirac function :

$$\delta(x) = \begin{cases} +\infty & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases}$$

$$\begin{aligned} \frac{\partial \mathcal{D}'(x(\lambda), y(\lambda))}{\partial \lambda} &= \frac{\partial \mathcal{D}'(x,y)}{\partial x}\frac{\partial x(\lambda)}{\partial \lambda} + \frac{\partial \mathcal{D}'(x,y)}{\partial y}\frac{\partial y(\lambda)}{\partial \lambda} \\ &= \sum \left(\frac{H(x_i)}{P_i} + \frac{x_i}{P_i} H'(x_i) \right) V_i - \sum \left(\frac{H(y_i)}{P_i} + \frac{y_i}{P_i} H'(y_i) \right) V_i \\ &= \sum \frac{V_i}{P_i} \left[H(x_i) - H(y_i) \right] + \frac{x_i \delta(x_i) - y_i \delta(y_i)}{P_i} \end{aligned}$$

Now, $H(x_i) - H(y_i)$ is the boxcar function that is always positive or nul. The fraction $\frac{V_i}{P_i}$ is always positive because V_i and P_i always have the same sign. Lastly, we note that $x\delta(x) = 0 \; \forall \; x$ because either $x = 0$ and $\delta(x) > 0$ or $x \neq 0$ but $\delta(x) = 0$. Hence, $\frac{\partial \mathcal{D}'(x(\lambda), y(\lambda))}{\partial \lambda} \geqslant 0$ and $\mathcal{D}'(\lambda)$ is always increasing.

E.6 Subsituing IF-statements by Heaviside in the dichotomy search

To keep the solution do $\mathcal{D}'(\lambda) = 0$ differentiable, we replace the IF-statements of the dichotomy search by the Heaviside function that derives as the Direct function.

$$(a) \text{ IF } \mathcal{D}'(M) < 0 \; lb \Leftarrow M$$
$$(b) \text{ IF } \mathcal{D}'(M) > 0 \; ub \Leftarrow M$$

Statement (b) becomes :

$$ub \Leftarrow ub - H\left[\mathcal{D}'(M)\right](ub - M)$$

because ub either takes the value ub if $\mathcal{D}'(M) < 0$ or $ub - (ub + M)$ if $\mathcal{D}'(M) > 0$. Analogously, statement (a) becomes :

$$\begin{aligned} lb &\Leftarrow lb + H\left[-\mathcal{D}'(M)\right](M - lb) \\ &= lb + (1 - H\left[\mathcal{D}'(M)\right])(M - lb) \\ &= M - H\left[\mathcal{D}'(M)\right] * (M - lb) \end{aligned}$$

Bibliography

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11), 2018.

Paris Agreement. Paris agreement. In *report of the conference of the parties to the United Nations framework convention on climate change (21st session, 2015: Paris). Retrived December*, volume 4, page 2017. HeinOnline, 2015.

Nima Amjady and Meisam Hemmati. Day-ahead price forecasting of electricity markets by a hybrid intelligent system. *European Transactions on Electrical Power*, 19(1):89–102, 2009.

Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145. PMLR, 2017.

S Anbazhagan and Narayanan Kumarappan. Day-ahead deregulated electricity market price forecasting using recurrent neural network. *IEEE Systems Journal*, 7(4):866–872, 2012.

Arash Andalib and Farid Atry. Multi-step ahead forecasts for electricity prices using narx: a new approach, a critical analysis of one-step ahead forecasts. *Energy Conversion and Management*, 50(3):739–747, 2009.

K Arya and KRM Vijaya Chandrakala. Machine learning based prediction and forecasting of electricity price during covid-19. In *2021 IEEE International Power and Renewable Energy Conference (IPRECON)*, pages 1–6. IEEE, 2021.

Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.

Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, 31:606–660, 2017.

Ross Baldick, Ryan Grant, and Edward Kahn. Theory and application of linear supply function equilibrium in electricity markets. *Journal of regulatory economics*, 25:143–167, 2004.

Martin T Barlow. A diffusion model for electricity prices. *Mathematical finance*, 12(4):287–298, 2002.

C Batlle. A model for electricity generation risk analysis. *Industrial Organization*, pages 1–223, 2002.

Carlos Batlle and Julián Barquín. A strategic production costing model for electricity market price analysis. *IEEE Transactions on Power Systems*, 20(1):67–74, 2005.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

Alexander Boogert and Dominique Dupont. When supply meets demand: the case of hourly spot electricity prices. *IEEE Transactions on Power Systems*, 23(2):389–398, 2008.

Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*, 2017.

Leo Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

Derek W Bunn. Forecasting electric loads with multiple predictors. *Energy*, 10(6):727–732, 1985.

Alvaro Cartea and Marcelo G Figueroa. Pricing in electricity markets: a mean reverting jump diffusion model with seasonality. *Applied Mathematical Finance*, 12(4):313–335, 2005.

JinXing Che and JianZhou Wang. Short-term load forecasting using a kernel-based support vector regression combination model. *Applied energy*, 132:602–609, 2014.

Xia Chen, Zhao Yang Dong, Ke Meng, Yan Xu, Kit Po Wong, and HW Ngan. Electricity price forecasting with extreme learning machine and bootstrapping. *IEEE Transactions on Power Systems*, 27(4):2055–2062, 2012.

Yuntian Chen and Dongxiao Zhang. Theory-guided deep-learning for electrical load forecasting (tgdlf) via ensemble long short-term memory. *Advances in Applied Energy*, 1:100004, 2021.

Hsu-Yung Cheng, Ping-Huan Kuo, Yamin Shen, and Chiou-Jye Huang. Deep convolutional neural network model for short-term electricity price forecasting, 2020.

NEMO Committee et al. Euphemia public description. *NEMO Committee*, 2020.

Antonio J Conejo, Miguel A Plazas, Rosa Espinola, and Ana B Molina. Day-ahead electricity price forecasting using the wavelet transform and ARIMA models. *IEEE transactions on power systems*, 20(2):1035–1042, 2005.

Javier Contreras, Rosario Espinola, Francisco J Nogales, and Antonio J Conejo. Arima models to predict next-day electricity prices. *IEEE transactions on power systems*, 18(3):1014–1020, 2003.

Alberto Cruz, Antonio Muñoz, Juan Luis Zamora, and Rosa Espínola. The effect of wind generation and weekday on spanish electricity spot price forecasting. *Electric Power Systems Research*, 81(10):1924–1935, 2011.

Jesús Crespo Cuaresma, Jaroslava Hlouskova, Stephan Kossmeier, and Michael Obersteiner. Forecasting electricity spot-prices using linear univariate time-series models. *Applied Energy*, 77(1):87–106, 2004.

Sumeyra Demir, Krystof Mincev, Koen Kok, and Nikolaos G Paterakis. Data augmentation for time series regression: Applying transformations, autoencoders and adversarial networks to electricity price forecasting. *Applied Energy*, 304:117695, 2021.

Emir Demirovic, Peter J Stuckey, Tias Guns, James Bailey, Christopher Leckie, Kotagiri Ramamohanarao, Jeffrey Chan, et al. Dynamic programming for predict+ optimise. In *AAAI*, pages 1444–1451, 2020.

Julia Díaz, Álvaro Romero, and José Ramón Dorronsoro. Day-ahead price forecasting for the spanish electricity market. *International Journal of Interactive Multimedia and Artificial Intelligence*, 5(4):42–50, 03/2019 2019. ISSN 1989-1660. doi: 10.9781/ijimai.2018.04.008. URL `http://www.ijimai.org/journal/sites/default/files/files/2018/04/ijimai_5_4_5_pdf_14997.pdf`.

Francis X Diebold. Comparing predictive accuracy, twenty years later: A personal perspective on the use and abuse of diebold–mariano tests. *Journal of Business & Economic Statistics*, 33(1):1–1, 2015.

Francis X Diebold and Robert S Mariano. Comparing predictive accuracy. *Journal of Business & economic statistics*, 20(1):134–144, 2002.

Abdou Kâ Diongue, Dominique Guegan, and Bertrand Vignal. Forecasting electricity spot market prices with a k-factor gigarch process. *Applied energy*, 86(4):505–510, 2009.

Priya Donti, Brandon Amos, and J Zico Kolter. Task-based end-to-end model learning in stochastic optimization. *Advances in neural information processing systems*, 30, 2017.

Othman El Balghiti, Adam N Elmachtoub, Paul Grigas, and Ambuj Tewari. Generalization bounds in the predict-then-optimize framework. *Advances in neural information processing systems*, 32, 2019.

Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1):9–26, 2022.

Thomas Epelbaum, Fabrice Gamboa, Jean-Michel Loubes, and Jessica Martin. Deep learning applied to road traffic speed forecasting. *arXiv preprint arXiv:1710.08266*, 2017.

Shahrazad Far and Richard Youngs. Energy union and eu global strategy. *SIEPS Report N°*, 5, 2015.

Aaron Ferber, Bryan Wilder, Bistra Dilkina, and Milind Tambe. Mipaal: Mixed integer program as a layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1504–1511, 2020.

Antonio García Alcalde, Mariano Ventosa Rodríguez, Michel Luis Rivier Abbad, Andrés Ramos Galán, and Gregorio Relaño Cobián. Fitting electricity market models. a conjectural variations approach. In *14th Power Systems Computation Conference*. Sin editorial (Sevilla, España), 2002.

Carolina Garcia-Ascanio and Carlos Maté. Electric power demand forecasting using interval time series: A comparison between var and imlp. *Energy Policy*, 38(2):715–725, 2010.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

Virginia Gonzalez, Javier Contreras, and Derek W Bunn. Forecasting power prices using a hybrid fundamental-econometric model. *IEEE Transactions on Power Systems*, 27(1):363–372, 2011.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Tao Hong. Crystal ball lessons in predictive analytics. *EnergyBiz Mag*, 12(2):35–37, 2015.

Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *International conference on machine learning*, pages 754–762. PMLR, 2014.

IEA. Competition in electricity markets. `https://www.iea.org/reports/competition-in-electricity-markets`, 2001.

Joanna Janczura, Stefan Trück, Rafał Weron, and Rodney C Wolff. Identifying spikes and seasonal components in electricity spot price data: A guide to robust modeling. *Energy Economics*, 38:96–110, 2013.

Tim Janke and Florian Steinke. Forecasting the price distribution of continuous intraday electricity trading. *Energies*, 12(22):4262, 2019.

Tor Arnt Johnsen. Demand, generation and price in the norwegian market for electric power. *Energy Economics*, 23(3):227–251, 2001.

Argyro Kampouraki, George Manis, and Christophoros Nikou. Heartbeat time series classification with support vector machines. *IEEE transactions on information technology in biomedicine*, 13(4):512–518, 2008.

Takashi Kanamura and Kazuhiko Ōhashi. A structural model for electricity prices with spikes: Measurement of spike risk and optimal policies for hydropower plant operation. *Energy economics*, 29(5):1010–1032, 2007.

Nektaria V Karakatsani and Derek W Bunn. Forecasting electricity prices: The impact of fundamentals and time-varying coefficients. *International Journal of Forecasting*, 24(4): 764–785, 2008.

Dogan Keles, Jonathan Scelle, Florentina Paraschiv, and Wolf Fichtner. Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks. *Applied energy*, 162:218–230, 2016.

Zahoor Ali Khan, Sahiba Fareed, Mubbashra Anwar, Afrah Naeem, Hira Gul, Arooj Arif, and Nadeem Javaid. Short term electricity price forecasting through convolutional neural network (cnn). In *Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 34th International Conference on Advanced Information Networking and Applications (WAINA-2020)*, pages 1181–1188. Springer, 2020.

Christopher R Knittel and Michael R Roberts. An empirical examination of restructured electricity prices. *Energy Economics*, 27(5):791–817, 2005.

Tarjei Kristiansen. Forecasting nord pool day-ahead prices with an autoregressive model. *Energy Policy*, 49:328–332, 2012.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Sergei Kulakov. X-model: further development and possible modifications, 2019.

Jesus Lago, Fjo De Ridder, and Bart De Schutter. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Applied Energy*, 221:386–405, 2018a.

Jesus Lago, Fjo De Ridder, Peter Vrancx, and Bart De Schutter. Forecasting day-ahead electricity prices in europe: The importance of considering market integration. *Applied energy*, 211:890–903, 2018b.

Jesus Lago, Grzegorz Marcjasz, Bart De Schutter, and Rafał 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.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, 2018.

F Lira, C Muñoz, F Nuñez, and A Cipriano. Short-term forecasting of electricity prices in the colombian electricity market. *IET generation, transmission & distribution*, 3(11):980–986, 2009.

Nicole Ludwig, Stefan Feuerriegel, and Dirk Neumann. Putting big data analytics to work: Feature selection for forecasting electricity prices using the lasso and random forests. *Journal of Decision Systems*, 24(1):19–36, 2015.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.

Katarzyna Maciejowska, Bartosz Uniejewski, and Rafal Weron. Forecasting electricity prices, 2022.

Jayanta Mandi and Tias Guns. Interior point solving for lp-based prediction+ optimisation. *Advances in Neural Information Processing Systems*, 33:7272–7282, 2020.

Jayanta Mandi, Peter J Stuckey, Tias Guns, et al. Smart predict-and-optimize for hard combinatorial optimization problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1603–1610, 2020.

Jayanta Mandi, Vıctor Bucarey, Maxime Mulamba Ke Tchomba, and Tias Guns. Decision-focused learning: through the lens of learning to rank. In *International Conference on Machine Learning*, pages 14935–14947. PMLR, 2022.

Grzegorz Marcjasz. Forecasting electricity prices using deep neural networks: A robust hyper-parameter selection scheme. *Energies*, 13(18):4605, 2020.

Grzegorz Marcjasz, Tomasz Serafin, and Rafał Weron. Selection of calibration windows for day-ahead electricity price forecasting. *Energies*, 11(9):2364, 2018.

Jie Mei, Dawei He, Ronald Harley, Thomas Habetler, and Guannan Qu. A random forest method for real-time price forecasting in new york electricity market. In *2014 IEEE PES General Meeting| Conference & Exposition*, pages 1–5. IEEE, 2014.

Adam Misiorek, Stefan Trueck, and Rafal Weron. Point and interval forecasting of spot electricity prices: Linear vs. non-linear time series models. *Studies in Nonlinear Dynamics & Econometrics*, 10(3), 2006.

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

Hossam Mosbah and Mohamed El-Hawary. Hourly electricity price forecasting for the next month using multilayer neural network. *Canadian Journal of Electrical and Computer Engineering*, 39(4):283–291, 2016.

Michał Narajewski and Florian Ziel. Econometric modelling and forecasting of intraday electricity prices. *Journal of Commodity Markets*, 19:100107, 2020a.

Michał Narajewski and Florian Ziel. Changes in electricity demand pattern in europe due to covid-19 shutdowns, 2020b.

Dongxiao Niu, Da Liu, and Desheng Dash Wu. A soft computing system for day-ahead electricity price forecasting. *Applied Soft Computing*, 10(3):868–875, 2010.

Francisco J Nogales and Antonio J Conejo. Electricity price forecasting through transfer function models. *Journal of the Operational Research Society*, 57(4):350–356, 2006.

Francisco Javier Nogales, Javier Contreras, Antonio J Conejo, and Rosario Espínola. Forecasting next-day electricity prices by time series models. *IEEE Transactions on power systems*, 17(2):342–348, 2002.

Kin G Olivares, Cristian Challu, Grzegorz Marcjasz, Rafał Weron, and Artur Dubrawski. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with nbeatsx. *International Journal of Forecasting*, 39(2):884–900, 2023.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

James Orlin. Lecture notes in optimization methods in management science, 2023.

Carlotta Orsenigo and Carlo Vercellis. Combining discrete svm and fixed cardinality warping distances for multivariate time series classification. *Pattern Recognition*, 43(11):3787–3794, 2010.

Kadir Özen and Dilem Yıldırım. Application of bagging in day-ahead electricity price forecasting and factor augmentation. *Energy Economics*, 103:105573, 2021.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12: 2825–2830, 2011.

NM Pindoriya, SN Singh, and SK Singh. An adaptive wavelet neural network-based energy price forecasting in electricity markets. *IEEE Transactions On power systems*, 23(3):1423–1432, 2008.

Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

Claudio M Ruibal and Mainak Mazumdar. Forecasting the mean and the variance of electricity prices in deregulated markets. *IEEE Transactions on Power Systems*, 23(1):25–32, 2008.

Alejandro Pasos Ruiz, Michael Flynn, James Large, Matthew Middlehurst, and Anthony Bagnall. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 35(2):401–449, 2021.

Damien C Sansom, Tom Downs, and Tapan K Saha. Evaluation of support vector machine based forecasting tool in electricity price forecasting for australian national electricity market participants. *Journal of Electrical & Electronics Engineering, Australia*, 22(3):227–233, 2003.

Simon Schnürch and Andreas Wagner. Machine learning on epex order books: Insights and forecasts. *arXiv preprint arXiv:1906.06248*, 2019.

Mohammad Shahidehpour, Hatim Yamin, and Zuyi Li. *Market operations in electric power systems: forecasting, scheduling, and risk management*. John Wiley & Sons, 2003.

Lloyd S Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker, editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, Princeton, 1953.

V Sharma and Dipti Srinivasan. A hybrid intelligent model based on recurrent neural networks and excitable dynamics for price prediction in deregulated electricity market. *Engineering Applications of Artificial Intelligence*, 26(5-6):1562–1574, 2013.

David Sheppard. Gas shortages: What is driving europe's energy crisis?'. *Financial Times*, 11, 2021.

Ali Shiri, Mohammad Afshar, Ashkan Rahimi-Kian, and Behrouz Maham. Electricity price forecasting using support vector machines by considering oil and natural gas price impacts. In *2015 IEEE international conference on smart energy grid engineering (SEGE)*, pages 1–5. IEEE, 2015.

Manu Suvarna, Apoorva Katragadda, Ziying Sun, Yun Bin Choh, Qianyu Chen, PS Pravin, and Xiaonan Wang. A machine learning framework to quantify and assess the impact of covid-19 on the power sector: An indian context. *Advances in Applied Energy*, 5:100078, 2022.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Hamdy A Taha. *Integer programming: theory, applications, and computations*. Academic Press, 2014.

Stefano Teso, Laurens Bliek, Andrea Borghesi, Michele Lombardi, Neil Yorke-Smith, Tias Guns, and Andrea Passerini. Machine learning for combinatorial optimisation of partially-specified problems: Regret minimisation as a unifying lens. *arXiv preprint arXiv:2205.10157*, 2022.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Léonard Tschora, Erwan Pierre, Marc Plantevit, and Céline Robardet. Electricity price forecasting on the day-ahead market using machine learning. *Applied Energy*, 313:118752, 2022.

Léonard Tschora, Erwan Pierre, Marc Plantevit, and Céline Robardet. Forecasting electricity prices: An optimize then predict-based approach. In *International Symposium on Intelligent Data Analysis*, pages 446–458. Springer, 2023a.

Léonard Tschora, Tias Guns, Erwan Pierre, Marc Plantevit, and Céline Robardet. Electricity price forecasting based on order books: a differentiable optimization approach. In *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10, 2023b. doi: 10.1109/DSAA60987.2023.10302542.

Bartosz Uniejewski and Rafał Weron. Efficient forecasting of electricity spot prices with expert and lasso models. *Energies*, 11(8):2039, 2018.

Bartosz Uniejewski, Jakub Nowotarski, and Rafał Weron. Automated variable selection and shrinkage for day-ahead electricity price forecasting. *Energies*, 9(8):621, 2016.

Bartosz Uniejewski, Rafał Weron, and Florian Ziel. Variance stabilizing transformations for electricity spot price forecasting. *IEEE Transactions on Power Systems*, 33(2):2219–2229, 2017.

Ties Van der Heijden, Jesus Lago, Peter Palensky, and Edo Abraham. Electricity price forecasting in european day ahead markets: a greedy consideration of market integration. *IEEE Access*, 9:119954–119966, 2021.

Iivo Vehviläinen and Tuomas Pyykkönen. Stochastic factor model for electricity spot price—the case of the nordic market. *Energy Economics*, 27(2):351–367, 2005.

Xavier Vives. *Oligopoly pricing: old ideas and new tools*. MIT press, 1999.

Rafał Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International journal of forecasting*, 30(4):1030–1081, 2014.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1658–1665, 2019.

Allen J Wood, Bruce F Wollenberg, and Gerald B Sheblé. *Power generation, operation, and control*. John Wiley & Sons, 2013.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Hamidreza Zareipour, Claudio A Cañizares, Kankar Bhattacharya, and John Thomson. Application of public-domain market information to forecast ontario's wholesale electricity prices. *IEEE Transactions on Power Systems*, 21(4):1707–1717, 2006.

G Peter Zhang and Min Qi. Neural network forecasting for seasonal and trend time series. *European journal of operational research*, 160(2):501–514, 2005.

Jun Hua Zhao, Zhao Yang Dong, Zhao Xu, and Kit Po Wong. A statistical approach for interval forecasting of the electricity price. *IEEE Transactions on Power Systems*, 23(2): 267–276, 2008.

Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J Leon Zhao. Time series classification using multi-channels deep convolutional neural networks. In *International conference on web-age information management*, pages 298–310. Springer, 2014.

Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J Leon Zhao. Exploiting multi-channels deep convolutional neural networks for multivariate time series classification. *Frontiers of Computer Science*, 10:96–112, 2016.

Florian Ziel. Forecasting electricity spot prices using lasso: On capturing the autoregressive intraday structure. *IEEE Transactions on Power Systems*, 31(6):4977–4987, 2016.

Florian Ziel and Rick Steinert. Electricity price forecasting using sale and purchase curves: The x-model. *Energy Economics*, 59:435–454, 2016.

Florian Ziel and Rafał Weron. Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks. *Energy Economics*, 70:396–420, 2018.

Florian Ziel, Rick Steinert, and Sven Husmann. Efficient modeling and forecasting of electricity spot prices. *Energy Economics*, 47:98–111, 2015a.

Florian Ziel, Rick Steinert, and Sven Husmann. Forecasting day ahead electricity spot prices: The impact of the exaa to other european electricity markets. *Energy Economics*, 51:430–444, 2015b.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

FOLIO ADMINISTRATIF

THESE DE L'INSA LYON, MEMBRE DE L'UNIVERSITE DE LYON

NOM : TSCHORA
(avec précision du nom de jeune fille, le cas échéant)

DATE de SOUTENANCE : 17/01/2024

Prénoms : Léonard

TITRE : Machine Learning Techniques for Electricity Price Forecasting

Numéro d'ordre : 2024ISAL0007

NATURE : Doctorat

Ecole doctorale : Infomaths

Spécialité :Informatique

RESUME : Electricity is essential for the energetic transition due to the diversity of greenhouse-gas free means of production and its potential to replace fossil fuels in transportation, heating and industries. However, it requires a constant balance between generation and consumption to maintain intensity in the network, and it can't be stored efficiently. It is then necessary to use Price Fixing Algorithm (PFA) for developing competitive markets. Daily, Euphemia determines the prices for the next day in Europe, called the Day-Ahead prices, that maximize the Social Welfare, while maintaining energy balance. Unlike other purely speculative markets, the Day-Ahead price is algorithmically computed and that renders its forecasts paramount for many business applications.

This introduces the problem of Electricity Price Forecasting (EPF) at the European scale, that consists in predicting the 24 hourly prices for each market before their fixation at 12am. The literature highlights two approaches: Expert Models aim at replicating the PFA and computing the prices based on estimates of the inputs of Euphemia, while Data-Driven methods directly estimate prices using exogenous variables and past prices. Both approaches are incomplete : Expert Models approaches are theoretically appealing but fail to produce accurate forecasts in practice. Conversely, Data Driven approaches lack transparency, lowering the forecasts reliability. Also, the true relationship between variables and prices is only captured by Euphemia, implicitly limiting the performances of Data Driven approaches.

This thesis addresses those limitations. The first challenge is to produce accurate and explainable models for a given market. We achieve the former by extending methodologies from the literature, while we use Shap Values, a model-agnostic explanability tool, for the latter. Then, we build a multi-market forecasting model by representing the European network as a Graph where each market is a node labeled with its prices. Graph edges are connection lines between markets, and we estimate the cross-market flows using an optimization problem prior to training. Lastly, we combine the Euphemia algorithm with in a Neural Network (NN) that forecasts its inputs. To consider the price forecasting error in the NN's training, we compute the gradient of Euphemia's output with respect to its input, by vanishing the derivative of the dual function using a dichotomic search.

We hope this thesis will be beneficial for the EPF practitioners and will contribute toward bridging the gap between Expert Models and Data Driven approaches. We also believe that our work on mixing optimization problems with machine learning models will benefit the broader scientific community.

MOTS-CLÉS : Machine Learning, Electricity Markets, Differential Optimization

Laboratoire (s) de recherche : LIRIS – Laboratoire d'Informatique en Image et Systèmes d'Information

Directeur de thèse: Céline Robardet

Président de jury :

Composition du jury :
Rapporteurs : Massih-Reza AMINI, Siegfried NIJSSEN
Examinateurs : Stéphane CANU, Elisa FROMONT, Charlotte LACLAU, Erwan PIERRE, Marc PLANTEVIT, Céline ROBARDET