Towards the Prescriptive Analytics Paradigm for Energy Forecasting and Power System Optimization

Akylas Stratigakos

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Mines Paris-PSL

Vers le Paradigme de l'Analyse Prescriptive pour la Prévision Énergétique et l'Optimisation des Systèmes Électriques

Towards the Prescriptive Analytics Paradigm for Energy Forecasting and Power System Optimization

Soutenue par

Akylas Stratigakos

Le 10/07/2023

École doctorale n°621

Ingénierie des Systèmes, Matériaux, Mécanique, Energétique

Spécialité

Energétique et Génie des Procédés

Composition du jury :

Jean-Michel POGGI Professor, Université Paris Cité	*Président*
Pierre PINSON Professor, Imperial College London	*Rapporteur*
François VALLÉE Professor, University of Mons	*Rapporteur*
Louis WEHENKEL Professor, University of Liège	*Rapporteur*
Juan Miguel MORALES Associate Professor, University of Málaga	*Examinateur*
Carlos RUIZ MORA Associate Professor, UC3M	*Examinateur*
Sonja WOGRIN Professor, TU Graz	*Examinatrice*
Georges KARINIOTAKIS Director of Research, Mines Paris-PSL	*Directeur de thèse*

MINES PARIS | PSL

The road to hell is paved with works-in-progress.

Philip Roth

Abstract

To mitigate the adverse effects of climate change, the power sector is rapidly transitioning towards decarbonization through the integration of renewable energy sources, such as wind and solar. In this context, advanced data-driven methods, leveraging tools from machine learning and operations research, hold significant potential as key enablers to deal with the uncertainty and variability of weather-dependent renewable energy sources. This thesis takes a holistic approach by examining the model chain that goes from data to uncertainty modeling and then to decisions and developing data-driven methods that enable improved and resilient decision-making in modern power systems, focusing on a short-term operational time frame.

First, we develop methods to enable improved decisions from data. Particularly, we examine the interaction between forecasting and optimization, which comprise two integral parts of data-driven decision-making processes. To maximize forecast value and simplify complex model chains, we propose a method that integrates forecasting and optimization and directly learns decisions conditioned on some contextual information, such as weather and market conditions. To speed up traditional works and foster the adoption of advanced data-driven methods, we further develop an interpretable learning method to forecast the solutions to constrained optimization problems, thus bypassing the need for an optimization solver.

Next, we examine methods that address challenges associated with the deployment of data-driven methods in real-world applications. To enable the resilience of data-driven decision-making processes, we propose a principled approach to handle missing data in an operational setting using the task of day-ahead forecasting as a guiding example. To address the potential scarcity of training data, we further develop an optimization-based approach to pool data across a number of contextually-dependent problems, thereby improving the overall performance and robustness of decisions.

The proposed methods are validated in comprehensive numerical experiments related to power system operations and participation of renewable energy sources in competitive electricity markets. Overall, the methods and tools developed in this thesis contribute to the transition toward a decarbonized and sustainable electricity grid.

Acknowledgements

My journey through this Ph.D. has been deeply enriched by the support and influence of numerous individuals, both directly and indirectly. Teachers, professors, colleagues, mentors, and friends, have all played pivotal roles in shaping my academic pursuits and personal growth. While thanking each of them individually would be a herculean task, I wish to make a heartfelt attempt.

Foremost, I thank my supervisors, Georges and Andrea, for entrusting me with the opportunity to carry out this work and for their willingness to allow me to explore my own research interests within the context of this thesis. I am especially grateful to Georges, not only for his continuous guidance and mentorship, which propelled me beyond the expectations I had set for myself at the outset of this Ph.D., but also for his friendship. I am also grateful to the members of my thesis jury: Prof. Pierre Pinson, Prof. François Vallée, Prof. Louis Wehenkel, Prof. Juan Miguel Morales, Prof. Jean-Michel Poggi, Prof. Carlos Ruiz Mora, and Prof. Sonja Wogrin. Their willingness to take the time out of their busy schedule to review this work and provide constructive feedback is deeply appreciated. The comments they shared and the engaging discussions that ensued have been invaluable in enhancing the quality of my work.

During this Ph.D., I was fortunate to visit the OASYS research group at the University of Málaga on two occasions. I am grateful to Juanmi and Salva for their warm hospitality. The three-way meetings with them were both stimulating and challenging, fostering my growth. I also thank the rest of the members of the OASYS research group: Conchi, Asu, Antonio, Ricardo, Álvaro, Jesús, Jóse, Miguel Ángel, Manuel, and Lisa, for making my stay in Málaga productive and enjoyable. I reserve special thanks to Adrián for his pertinent paper suggestions that significantly shaped the latter part of this thesis, and for sharing his vast knowledge on topics spanning from distributionally robust optimization to Greek pop singers.

My time at Center PERSEE was greatly enriched by my interactions with an amazing group of researchers from diverse backgrounds. I greatly benefited from my collaboration with Panagiotis, whom I thank for his guidance and support. From reviewing my earlier work to intensive collaboration in the later stage of my Ph.D., I gained a great appreciation for his meticulous and diligent approach to research and technical writing. I thank Simon for all his guidance during the early stages of my Ph.D. and our fruitful collaborations.

I also thank Dennis for all the stimulating discussions and for sharing his expertise in forecasting. My appreciation further extends to my office mates, Kostas, Yun, Shengfei, Biswarup, Matias, and Sylvain; to the older members of the ERSEI group, Valentin, Kevin, Stefano, Alberto, and Anaëlle; to the newest group members, Owen, Luca, Lukas, Sergio, and Paul; and to the rest of my colleagues and friends, both from the ERSEI and the MATPRO group, who collectively made this journey more fulfilling. Although the COVID-19 pandemic significantly limited the amount of time we spent together, I am grateful for the engaging moments we shared over the last few years.

I would be remiss not to mention my formative experience at IPTO which was pivotal in my decision to pursue a Ph.D. degree. I thank particularly George Papaioannou for his mentorship and for instilling in me a passion for research, as well as my former colleagues from the Department of Research, Technology, and Development with whom we successfully collaborated on several research projects.

On a personal level, I would like to thank my friends and family. To the amazing community of fellow Ph.D. students I found here in France during the stressful lockdown period, I am truly thankful. To my close friends from Greece, your emotional support and enduring friendship are deeply appreciated. Lastly, I extend my deepest gratitude to both my close and extended family for all their unconditional love, support, and encouragement over the years. In particular, I thank my parents, to whom I dedicate this work.

Financial Support

I gratefully acknowledge the financial support provided by the Smart4RES Project (Grant No 864337) funded under the Horizon 2020 Framework Program and the European project REgions (Grant No 646039), supported by ADEME's 'Investissement d'Avenir' program and the ERA-Net SES RegSys project. Part of this work was carried out while visiting the OASYS research group at the University of Málaga, Málaga, Spain, which was supported in part by an Erasmus+ grant.

Contents

List of Figures

List of Tables

List of Acronyms

AGC Automatic Generation Control

CART Classification and Regression Trees

DC-OPF DC Optimal Power Flow

ECMWF European Centre for Medium-Range Weather Forecasts

FDRR Feature-Deletion Robust Regression

LAD Least Absolute Deviations

LP Linear Programming

LS Least Squares

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

MAR Missing at Random

MCAR Missing Completely at Random

MCI Mean Cost Increase

MDI Mean Decrease Impurity

MNAR Missing Not at Random

MSE Mean Squared Error

NN Neural Network

NWP Numerical Weather Prediction

OOB Out-of-Bag

OPF Optimal Power Flow

OT Optimal Transport

PACF Partial Autocorrelation Function

PTDF Power Transfer Distribution Factors

PV Photovoltaic

QP Quadratic Programming

QR Quantile Regression

QRF Quantile Regression Forests

SAA Sample Average Approximation

SPO Smart Predict-then-Optimize

SVM Support Vector Machine

WPP Wind Power Plant

Chapter 1

Introduction

Résumé en Français

Augmenter la part des sources d'énergie renouvelables dans le mix énergétique est crucial pour atténuer les risques liés au changement climatique. Cependant, la nature intermittente et variable des sources d'énergie renouvelables dépendantes des conditions météorologiques, telles que l'énergie éolienne et solaire, pose des défis importants dans le fonctionnement des systèmes électriques modernes. Parallèlement, la numérisation en cours des systèmes électriques, combinée à la libéralisation des marchés de l'électricité, a conduit à une disponibilité accrue des données. Les méthodes avancées basées sur les données, combinant des outils d'apprentissage automatique, de recherche opérationnelle et de science des données, offrent un potentiel important pour soutenir la prise de décision en exploitant les données disponibles et permettre la transition vers un réseau électrique décarboné, exploité de manière rentable et fiable. Dans ce chapitre, nous présentons d'abord la chaîne de modèles génériques qui passe des données à la modélisation de l'incertitude, puis aux décisions, qui sous-tend la plupart des applications de systèmes électriques et de gestion de l'énergie. L'accent est mis principalement sur le calendrier opérationnel, allant de quelques minutes à plusieurs jours à l'avance. Ensuite, nous identifions plusieurs défis associés au déploiement et au développement de méthodes avancées basées sur les données dans les systèmes électriques modernes. Ces défis incluent, entre autres, l'écart potentiel entre les méthodes de modélisation de l'incertitude et le problème d'optimisation en aval, la nécessité d'accélérer les flux de travail et les processus traditionnels pour faire face à une incertitude et une variabilité accrue, gérer la nature de boîte noire des algorithmes basés sur les données et gérer les risques liés aux données, tels que les données manquantes ou rares. En adoptant une approche holistique qui examine conjointement la chaîne de modèles allant des données aux décisions, cette thèse vise à relever ces défis grâce à des méthodes avancées basées sur les données pour les opérations des systèmes électriques qui permettent de meilleures décisions, améliorent l'interprétabilité, simplifient les chaînes de modèles complexes et sont résilientes aux risques liés aux données. Par la suite, les contributions techniques de la thèse par rapport aux défis mentionnés ci-dessus sont résumées. Enfin, ce chapitre se termine par un aperçu de la structure de ce document et une liste de publications pertinentes.

1.1 Context

Climate change poses significant dangers to human well-being, necessitating the implementation of mitigation strategies aimed at reducing greenhouse gas emissions. In this context, the power and energy sector plays a critical role, first as a contributor of approximately a quarter of global emissions and second as an enabler through the electrification of other sectors, such as transportation. Consequently, the transition towards a decarbonized electricity grid is a pressing issue. Achieving this goal requires diversifying electricity generation sources and reducing dependence on fossil fuels, such as natural gas and coal, and renewable energy sources, such as wind and solar, offer an effective solution. The established targets for emission reduction in the power sector are highly ambitious. For instance, the European Union raised the overall target for the integration of renewable energy sources for 2030 to approximately 40% [1]. Overall, the share of renewable energy sources in the generation mix is rapidly increasing and they are expected to become the largest source of global electricity generation by 2027 [2].

While the integration of renewable energy sources is a crucial step in the direction of mitigating climate change risks, their intermittent and variable nature raises some important challenges. Indeed, the integration of weather-dependent production from wind and solar results in significant uncertainty and variability in the power supply. For many years, the main sources of uncertainty in power systems were potential equipment failures, such as generator or line outages, and uncertain demand, while variability was primarily driven by demand fluctuations and did not affect the supply side. The integration of large shares of renewable production presents a shift in the traditional mode of operation, necessitating the development of advanced tools to mitigate uncertainty and operating power systems at a higher speed and scale, which challenges traditional workflows.

In recent years, data-driven decision-making methods have been making advancements and transforming various industries, such as manufacturing, finance, and healthcare, leveraging tools from optimization, machine learning, and statistics [4]. At the same time, electricity grids and power systems are becoming increasingly data-centric [5], through advanced monitoring, control, and communications capabilities. This leads to an increase in data availability, coming from several sources such as sensors, smart meters, and market information. The convergence of large data sets, improved computational resources, and the development of data-driven methods, such as machine learning, has resulted in the growth of *energy analytics*, i.e., a specialized field of data analytics tools geared towards the energy and power sector. Energy analytics tools are designed to utilize available data and operate within the function layer of modern power systems [3]— see Fig. 1.1 for an illustration— and hold significant potential as key enablers towards a decarbonized and sustainable electricity grid [6].

Data analytics provide value to stakeholders by delivering insights and interpretations of historical data, offering informed predictions of future events, and suggesting an appropriate

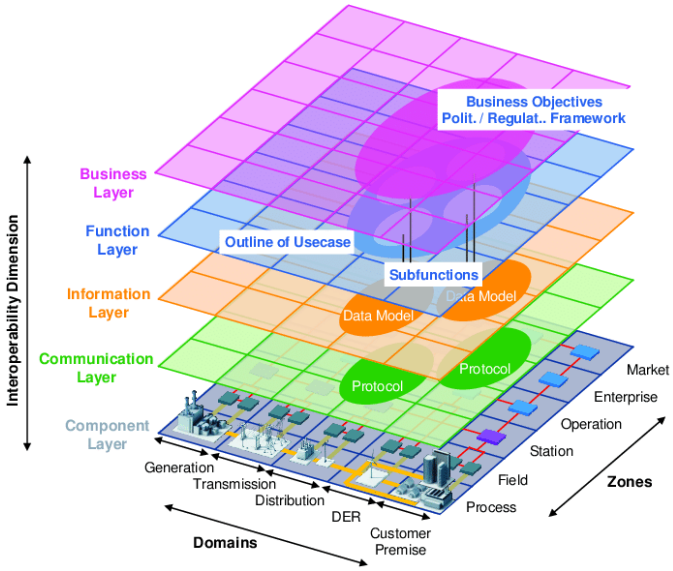


Figure 1.1: The Smart Grid Architecture Model (SGAM) [3].

set of actions to optimize outcomes [7]. In the context of the power sector, energy analytics provide forecasting, optimization, and control tools to support power systems with a high share of renewable production [8]. For instance, accurate forecasts of future production from renewable energy sources are important inputs in the operational management of modern power systems, enabling system operators to make informed decisions on balancing supply and demand, and reducing the risk of power outages. Additionally, accurate forecasts enable renewable energy sources to become financially competitive in deregulated electricity markets, by informing trading decisions and managing financial risks. By leveraging interdisciplinary methods from machine learning, operations research, and data science, energy analytics offer a comprehensive approach to mitigate the uncertainty and variability associated with the integration of renewable energy sources, having the potential to improve the efficiency and reliability of modern power systems.

A wide range of analytics tools are applied in modern power systems. In this thesis, we focus primarily on the operational and medium-term planning time frame, ranging from several minutes to several days ahead. In this context, predictive analytics and mathematical optimization are the main tools used within decision-making processes. Predictive analytics mostly concern forecasting applications in power systems. These include, among others, load forecasting, electricity price forecasting, wind production, and solar production forecasting, which, throughout this thesis, will be referred to as *energy forecasting* [9]. The goal is to leverage available contextual information, such as weather or historical production data, to provide an estimation of an uncertain parameter in a future time interval. Machine learning methods are becoming increasingly popular and are considered a relatively mature technology that can readily be used today for energy forecasting applications.

Conversely, mathematical optimization (hereafter, optimization) [10] aims to identify a set of actions that optimize a particular cost criterion while satisfying a set of physical

constraints. Common power system applications that leverage optimization tools in an operational time frame include, among others, determining the optimal dispatch schedule, estimating the appropriate system reserves, participating in competitive electricity markets, and managing controllable assets, e.g., storage devices [11]. Typically, the majority of optimization tools used in practice adopt a deterministic formulation. However, as the reliance on weather-dependent renewable energy sources grows, advanced optimization tools that incorporate uncertainty [12], such as stochastic, robust, and chance-constrained optimization, are becoming more widespread [13].

Predictive analytics and optimization comprise two integral parts of the sequential, two-step process that spans the model chain from data to decisions, which is illustrated in Fig. 1.2. In the first step, predictive analytics tools provide accurate estimations, i.e., forecasts, of future uncertain quantities, such as renewable production or market quantities. In the next step, these forecasts are used as inputs in an optimization problem to find a set of actions that optimize an objective function, subject to a number of physical constraints.

Naturally, the model chain presented in Fig. 1.2 may vary depending on the stakeholders and business cases. For instance, a forecast provider focuses on developing predictive analytics methods to determine the statistics of uncertain variables, while a trader uses forecasts to participate in electricity markets and hedge financial risks. Conversely, a grid operator assessing system security may have to solve an optimization problem repeatedly for a large number of uncertainty scenarios, while an aggregator operating a large portfolio of assets may have to deal with several independent model chains that define similar problems in parallel. Analytics tools and data-driven methods offer a variety of avenues to improve these processes, including improving forecasting, streamlining traditional workflows, and enabling improved and more resilient decisions.

The next step towards increasing the maturity of analytics tools in power systems is to take a holistic approach and examine the entire model chain of Fig. 1.2 under a *prescriptive analytics* framework. Prescriptive analytics methods transform the available contextual information, such as weather forecasts, into implementable decisions, enhancing decision quality and maximizing stakeholder value. Prescriptive analytics, as defined in the context of this thesis, explicitly depend on data provided by some external sources and integrate both uncertainty modeling (i.e., forecasting) and optimization components. In fact, both forecasting and optimization tasks can be considered special cases of prescriptive analytics.

1.2 Challenges, Gaps, and Contributions

There are several challenges associated with the effective development and deployment of prescriptive analytics tools and, in general, data-driven methods for power systems. These challenges relate to the interactions between the different components in Fig. 1.2, practical considerations about the deployment of complex model chains, the black-box nature of data-driven tools, and risks associated with external factors, such as data-related challenges.

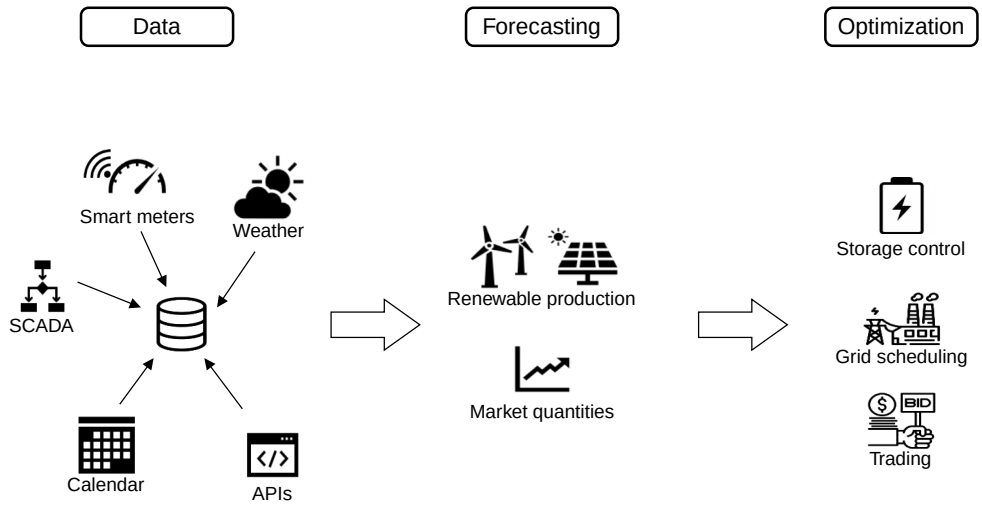


Figure 1.2: A generic model chain going from data to decisions.

The first challenge concerns the interaction between forecasting and optimization. While optimization tools typically offer strong guarantees and provably optimal decisions, they may rely on a deterministic setting that ignores uncertainty. In practice, the decision quality depends heavily on the forecast quality. However, forecasting models are typically trained to maximize statistical accuracy, independent from the downstream optimization problem. As a result, forecasting models do not account for the impact of forecasting errors on the decision costs, i.e., *forecast value*. For instance, [14] examines the economic impact of electricity price forecasting errors and finds that increased accuracy does not always translate into economic benefits; in fact, in some cases, improving forecast accuracy might be counterproductive [15]. Consequently, the transition from prediction to prescription (equivalently, decision) is not always straightforward. To maximize forecast value, it is necessary to develop novel approaches that are cognizant of the downstream optimization problem and embed its cost function and physical constraints within the learning process.

From a practical standpoint, the model chain presented in Fig. 1.2 can involve significant modeling effort, as it requires forecasting multiple uncertain quantities separately. For instance, an aggregator managing a portfolio of renewable power plants participating in short-term electricity markets must forecast a large number of unknown parameters in order to effectively hedge against financial losses [16]. Developing and maintaining a large number of forecasting models can be a labor-intensive task, highlighting the need for innovative methods that simplify complex model chains in real-world applications.

Moreover, deploying multiple analytics tools within a decision-making process can introduce additional complexity and obscure the impact of data on decisions. Furthermore, data-driven methods, especially those based on machine learning, are often characterized by their black-box nature, which can hinder their adoption in industrial applications. This is

especially true for critical infrastructure industries such as power systems, where stakeholders are highly risk-averse. To facilitate the adoption of advanced data-driven methods, it is crucial to provide stakeholders with explainable and interpretable decisions and provide performance guarantees [17].

In addition, the effective deployment of analytics tools for power systems is also subject to external risks associated with data quality and availability. Models deployed in industrial settings must address several data-management challenges that can emerge only after the system is online [18]. For instance, network latency, equipment failures, or cyberattacks may render input data unavailable and compromise model performance. Moreover, future power systems will integrate a multitude of heterogeneous assets, such as small-scale renewable energy sources and flexible loads. While the aggregate volume of data is large, decision-makers may encounter data scarcity on an individual asset level, which can negatively affect the performance of machine learning-based models. To ensure the reliability and consistency of data-driven decision-making processes, it is crucial to develop novel approaches that instill resilience against data-related risks and effectively leverage available data from various sources.

In this thesis, we take an interdisciplinary approach that leverages tools from machine learning, operations research, and data science, to address challenges associated with the development and deployment of prescriptive analytics tools in modern power systems, largely focusing on the operational time frame. Specifically, our overarching goal is:

To develop data analytics tools for power systems operations that enhance decision-making processes by improving techno-economic benefits, simplifying complex model chains, increasing transparency and explainability, and enabling resilience against data-related risks.

Our contributions are summarized as follows:

1. To maximize forecast value and reduce modeling effort, we propose an approach that integrates forecasting and optimization and provides a generic framework to evaluate the impact of data on decisions, thus also improving explainability. The proposed method is validated in several real-world case studies related to electricity market participation.

2. To foster the adoption of advanced data-driven methods and speed up traditional workflows, we develop an interpretable learning approach to directly forecast the decisions of a constrained optimization problem, thus bypassing the need for an optimization solver. To ensure interpretability, we employ a two-step approach that incorporates domain knowledge into model development. We demonstrate the effectiveness of our approach on a critical operations task for power systems and electricity markets.

3. To improve the resilience of data-driven decision-making processes, we propose a principled approach to handle missing data in an operational setting. Unlike ad hoc

solutions commonly deployed in practice, our method leads to consistent performance and effective hedging against worst-case scenarios while maintaining practicality. We demonstrate the efficacy of our approach in several prevalent energy forecasting applications and subsequently apply it in the context of integrated forecasting and optimization.

4. To deal with the potential data scarcity, we propose an optimization-based method to pool data across a number of independent problems, thereby improving the overall performance and robustness of energy analytics tools.

1.3 Structure of the Thesis

Each chapter of this thesis contributes to a different aspect of the model chain described in Fig. 1.2 and is intended to be comprehensible when read separately. A detailed description of the chapters is as follows.

In Chapter 2, we examine the complete model chain presented in Fig. 1.2, that is, we examine decision-making problems under uncertainty, where the uncertainty is associated with some contextual information. To maximize forecast value and enable improved decisions, we propose an integrated forecasting-optimization method and further establish a generic framework to evaluate the impact of data on decision performance. Specifically, we formulate a tree-based algorithm trained to minimize decision costs and adapt feature importance metrics in a prescriptive context. For validation, we examine various problems related to the participation of renewable energy sources in competitive electricity markets. A series of numerical experiments with real-world data illustrate that the proposed approach outperforms the standard modeling approach, while also reducing the associated modeling effort.

In Chapter 3, we consider a setting that involves solving an optimization problem repeatedly for different realizations of uncertainty, which can be considered as a special case of the problem examined in Chapter 2 assuming a one-to-one mapping between contextual information and uncertainty. Rather than looking for improved decisions, our goal in this chapter is to examine methods that speed up traditional workflows. As a guiding example, we use the DC Optimal Power Flow (DC-OPF) problem, which is pivotal in the operation of power systems and electricity markets. We develop an interpretable method to forecast the solutions of a constrained optimization problem with feasibility guarantees, extending the method developed in Chapter 2. Particularly, we propose a tree-based algorithm that learns a piecewise affine mapping from data to decisions, thus eliminating the need for an optimization solver at test time, using robust optimization to ensure that decisions are feasible. To enhance both model performance and interpretability, we encode domain knowledge during model development. We provide extensive empirical validation, under different types of uncertainty and operating conditions, with our results demonstrating that interpretable trees

perform comparably to state-of-the-art methods that do not offer performance guarantees.

In Chapter 4, we examine the issue of missing data in an operational setting, i.e., after a model has been deployed in production. We present a robust optimization approach to enable model resilience, using the task of energy forecasting in a day-ahead horizon as a guiding example. Specifically, we formulate a robust regression model that is optimally resilient against missing data at test time, considering both point and probabilistic forecasting, and develop three solution methods, with varying degrees of tractability and conservativeness. We provide an extensive empirical validation of the proposed methods in prevalent forecasting applications in power systems, against well-established benchmarks and methods of dealing with missing features. Next, we apply the proposed approach in an integrated forecasting and optimization framework, whereby we directly forecast the decisions of a renewable producer participating in a day-ahead market. The results show that the proposed approach enables model resilience, while also maintaining practicality.

Chapter 5 further examines data-related issues, specifically dealing with scarce training data. We consider dealing with multiple stochastic optimization problems, each associated with some contextual information, as in Chapter 2, and investigate data pooling methods to address data scarcity on an individual problem level. We propose two methods to leverage data across a number of problems and further develop an optimization-based data pooling algorithm that determines when and how much data to pool, effectively interpolating between a local and a pooled distribution. We validate our approach in two pivotal applications related to the integration of renewable energy sources, namely power production forecasting and trading in a day-ahead electricity market. Our empirical results show that data pooling mitigates the solution instability when data are scarce, thereby leading to improved predictive and prescriptive performance.

Finally, in Chapter 6, we summarize the work presented and offer perspectives on future developments.

1.4 List of Publications

The following publications were prepared in the context of my Ph.D.:

Journal Publications

[**J1**] **A. Stratigakos**, P. Andrianesis, A. Michiorri and G. Kariniotakis,"Towards Resilient Energy Forecasting Against Missing Features: a Robust Optimization Approach," in *IEEE Transactions on Smart Grid*, pp. 1-1, May 2023. Preprint available at: `https://hal.science/hal-03792191/`.

[**J2**] **A. Stratigakos**, S. Camal, A. Michiorri and G. Kariniotakis, "Prescriptive Trees for Integrated Forecasting and Optimization Applied in Trading of Renewable Energy,"

in *IEEE Transactions on Power Systems*, vol. 37, no. 6, pp. 4696-4708, Nov. 2022. Preprint available at: `https://hal.science/hal-03330017v3`.

Working Papers/Under Review

[**J3**] **A. Stratigakos**, S. Pineda, J. M. Morales and G. Kariniotakis, "Interpretable Machine Learning for DC Optimal Power Flow with Feasibility Guarantees," in *IEEE Transactions on Power Systems (3rd round of review)*. Preprint available at: `https://hal.science/hal-04038380`.

[**J4**] **A. Stratigakos**, S. Pineda, J. M. Morales and G. Kariniotakis, "Optimization-based Data Pooling for Contextual Stochastic Optimization," in preparation for submission in *European Journal of Operational Research*.

Conference Publications (Peer Reviewed)

[**C1**] M. Kühnau, **A. Stratigakos**, S. Camal, S. Chevalier and G. Kariniotakis, "Resilient Feature-driven Trading of Renewable Energy with Missing Data," 2023 *IEEE Innovative Smart Grid Technologies - Europe*, Grenoble, France, 2023.

[**C2**] **A. Stratigakos**, D. van der Meer, S. Camal and G. Kariniotakis, "End-to-end Learning for Hierarchical Forecasting of Renewable Energy Production with Missing Values," 2022 *17th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, Manchester, United Kingdom, 2022, pp. 1-6. Preprint available at: `https://hal.science/hal-03527644`.

[**C3**] **A. Stratigakos**, A. Michiorri and G. Kariniotakis, "A Value-Oriented Price Forecasting Approach to Optimize Trading of Renewable Generation," 2021 *IEEE Madrid PowerTech*, Madrid, Spain, 2021, pp. 1-6. Preprint available at: `https://hal.science/hal-03208575v1`.

Conference Presentations

- **A. Stratigakos**, P. Andrianesis, A. Michiorri and G. Kariniotakis, "Making Energy Forecasting Resilient to Missing Features: a Robust Optimization Approach," *42nd International Symposium on Forecasting*, 2022 (Best Student Presentation & Travel Grant Award).
- **A. Stratigakos**, S. Camal, A. Michiorri and G. Kariniotakis, "An Integrated Forecasting and Optimization Approach Applied in Trading Renewable Energy," *41st International Symposium on Forecasting*, June 27-30, 2021.
- **A. Stratigakos**, S. Camal, T. Blondel and G. Kariniotakis, "Short-term Trading of Wind Energy Production Using Data-driven Prescriptive Optimization," *Wind Energy Science Conference*, May 2021, Hannover, Germany.

The main contributions of the thesis appear in the journal publications [**J1**]-[**J4**], while the conference publications ([**C1**]-[**C3**]) do not explicitly appear in the remainder of the thesis.

The following publications are the result of my participation in the "IEEE-CIS Technical Challenge on Predict+Optimize for Renewable Energy Scheduling," where some of the methods developed in my Ph.D. were tested, and do not explicitly appear in the remainder of the thesis:

- C. Bergmeir, F. de Nijs, A. Sriramulu, M. Abolghasemi, R. Bean, J. Betts, Q. Bui, N. T. Dinh, N. Einecke, R. Esmaeilbeigi, S. Ferraro, P. Galketiya, E. Genov, R. Glasgow, R. Godahewa, Y. Kang, S. Limmer, L. Magdalena, P. Montero-Manso, D. Peralta, Y. P. S. Kumar, A. Rosales-Pérez, J. Ruddick, **A. Stratigakos**, P. Stuckey, G. Tack, I. Triguero and R. Yuan, "Comparison and Evaluation of Methods for a Predict+Optimize Problem in Renewable Energy," in *IEEE Transactions on Neural Networks and Learning Systems (under review)*. Preprint available at :`https://arxiv.org/abs/2212.10723`.

- **A. Stratigakos**, "A Robust Fix-and-Optimize Matheuristic for Timetabling Problems with Uncertain Renewable Energy Production," *IEEE Symposium Series on Computational Intelligence (invited)* 2021, IEEE, Dec 2021, Orlando, United States. Preprint available at: `https://hal.science/hal-03449920v1`.

During my Ph.D. I also co-authored the following publications which are outside of the scope of the thesis and do not appear in the remainder:

- **A. Stratigakos**, A. Bachoumis, V. Vita and E. Zafiropoulos, "Short-Term Net Load Forecasting with Singular Spectrum Analysis and LSTM Neural Networks," in *Energies*, 14(14), 4107, 2021.

- K. Krommydas, **A. Stratigakos**, C. Dikaiakos, G. Papaioannou, M. Jones and G. McLoughlin, "A Novel Modular Mobile Power Flow Controller for Real-Time Congestion Management Tested on a 150kV Transmission System," in *IEEE Access*, vol. 10, pp. 96414-96426, 2022.

- K. Krommydas, C. Dikaiakos, G. Papaioannou and **A. Stratigakos**, "Flexibility Study of the Greek Power System Using a Stochastic Programming Approach for Estimating Reserve Requirements," in *Electric Power Systems Research*, 213, p.108620, 2022.

1.5 Notation

Throughout the thesis, boldfaced lowercase letters, e.g., $\mathbf{x}$, denote vectors, and boldfaced uppercase letters, e.g., $\mathbf{X}$, denote matrices. Sets are denoted with calligraphic font, e.g., $\mathcal{S}$,

and scalars with ordinary letters, either lowercase or uppercase, e.g., n or N. The notation $[n]$ is used as a shorthand for $\{1, \ldots, n\}$ and $|\mathcal{S}|$ denotes the cardinality (i.e., number of elements) of a set $\mathcal{S}$.

Chapter 2

Integrating Forecasting and Optimization to Improve Decision Performance

Résumé en Français

Déduire des décisions à partir de données implique généralement un processus séquentiel en deux étapes avec deux composants. Dans la première étape, un modèle de prévision est déployé pour prédire les paramètres incertains du problème. Dans la deuxième étape, ces prévisions sont utilisées comme données d'entrée dans un problème d'optimisation qui en déduit un ensemble d'actions appropriées. Les modèles de prévision apprennent généralement en minimisant une fonction de perte qui se présente comme une approximation des coûts spécifiques à une tâche (par exemple, le commerce, la planification) sans tenir compte du problème d'optimisation en aval. En pratique, cela crée un goulot d'étranglement des performances et masque l'impact des données sur les décisions. Pour relever ces défis, nous proposons un module unique basé sur les données qui exploite la structure du composant d'optimisation et apprend directement une politique conditionnée par des informations contextuelles. Nous développons un algorithme pour former des ensembles d'arbres de décision en minimisant directement les coûts spécifiques à la tâche, et prescrivons des décisions via une approximation pondérée de la moyenne d'échantillon du problème d'origine. Pour évaluer l'impact des informations contextuelles sur la performance décisionnelle, nous adaptons davantage les métriques d'importance des fonctionnalités dans un contexte normatif. La méthode proposée est validée dans diverses études de cas liées à la commercialisation de la production d'énergie renouvelable et à la participation aux marchés de l'électricité. Nous considérons le problème de l'arbitrage des prix avec un dispositif de stockage, suivi du problème de l'échange de production renouvelable sur un marché journalier sous différents mécanismes d'équilibrage, et nous proposons des stratégies qui équilibrent les décisions de trading optimales et la précision des previsions. Enfin, nous considérons une agrégation de centrales renouvelables et de stockage, et optimisons à la fois la stratégie de trading day-ahead et la politique de contrôle opérationnel du stockage, sur la base d'une approximation traitable utilisant l'approche de la règle de décision linéaire. Les résultats empiriques démontrent que le cadre de modélisation prescriptif proposé surpasse constamment le cadre de modélisation standard.

The work in this chapter extends the work previously published in [**J2**].

2.1 Introduction

Data play an increasingly important role in decision-making processes in modern power systems. Moving from data to decisions usually involves a two-step, sequential process. The first step involves modeling uncertainty stemming from multiple sources, such as stochastic renewable production and unknown market quantities. To this end, forecasting models are typically deployed to predict uncertain parameters at a future time interval conditioned on some associated contextual information, such as weather or market conditions. In the second step, the output of the forecasting models is used as input in an optimization problem, which finds the set of actions (equivalently, decisions or prescriptions) that minimize a cost function while considering a set of physical constraints.

Forecasting models, usually based on machine learning or statistical methods, are trained by minimizing a loss function over a training data set. This loss function optimizes a statistical criterion, such as accuracy, and is agnostic to the downstream optimization problem, thus, it serves only as a proxy for the true decision cost. However, increased forecast accuracy does not always lead to better decisions. For instance, [14] examines the economic impact of electricity price forecasting errors and shows that increased accuracy does not always translate into increased economic value. Recently, there has been a growing trend of moving beyond the simple statistical evaluation of forecasting errors to assessing the incurred decision cost associated with these errors. For instance, [19] proposes a multivariate probabilistic forecasting model and considers the economic benefits for an electricity retailer as a means of assessing its benefits. Indeed, assessing the impact of forecasts on decision costs, i.e., *forecast value*, is considered to be one of the key challenges in energy forecasting in the coming years [9]. Further, directly optimizing towards forecast value rather than accuracy is identified as a high-leverage objective to employ machine learning as means of enabling the decarbonization of power systems [6]. To maximize forecast value, therefore, we need to embed knowledge about the downstream task in the learning process of the forecasting model.

2.1.1 Aim and Contribution

In this chapter, we jointly examine forecasting and optimization for decision-making in power systems and electricity markets. Inspired by the framework established in [20], we integrate forecasting and optimization by formulating ensembles of decision trees trained to directly learn decisions from data and maximize forecast value. The proposed integrated approach allows for directly considering multiple sources of uncertainty, thus reducing the associated modeling effort, and provides decisions that satisfy possible physical constraints. To evaluate the impact of data on decisions and enhance model explainability, we further adapt well-known metrics from the machine learning literature in a prescriptive context. We validate the proposed approach on several real-world case studies related to the integration

of renewable energy sources in competitive electricity markets and demonstrate improved decision performance compared to the standard modeling approach.

Our contributions are summarized as follows:

- We propose and validate an integrated forecasting-optimization modeling approach for power system applications that leverages contextual information to directly learn decisions from data. The proposed approach *(i)* improves prescriptive performance, *(ii)* reduces the modeling effort, *(iii)* handles multiple sources of uncertainty, and *(iv)* guarantees the feasibility of decisions.

- Methodologically, we propose tree-based ensembles trained to minimize decision costs and adapt well-known feature importance metrics from the machine learning literature to a prescriptive context.

- We illustrate the efficacy of the proposed approach in various case studies of increasing complexity related to participation in electricity markets. First, we examine the problem of price arbitrage with a storage device. Then, we examine trading renewable production in a day-ahead market under different pricing mechanisms and propose strategies that balance trading cost and predictive accuracy. Finally, we consider a combination of renewable plants and a storage system and jointly optimize the day-ahead offering strategy and operational control policy; for the latter, we employ the linear decision rule approach [21] to provide a tractable approximation.

2.1.2 Chapter Outline

The rest of the chapter is organized as follows. Section 2.2 presents the mathematical background and reviews related work. Section 2.3 develops the proposed methodology. Section 2.4 formulates relevant power system applications to apply the proposed methodology. Section 2.5 presents the numerical experiments. Section 2.6 provides a summary and conclusions.

2.2 Mathematical Background and Related Work

This section presents the mathematical framework and related work (in Subsection 2.2.1) and reviews related applications in power systems (in Subsection 2.2.2).

2.2.1 Mathematical Framework and Related Work

We examine decision-making problems under uncertainty where $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^{d_y}$ denotes some unknown problem parameters, such as renewable production or market prices, and $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$ denotes associated contextual information (also known as *features*), such as weather or market conditions. The uncertain problem parameters and the associated

contextual information follow a joint probability distribution $(\mathbf{x}, \mathbf{y}) \sim \mathbb{Q}$. Our goal is to solve the following contextual stochastic optimization (or prescriptive analytics) problem

$$v = \min_{\mathbf{z} \in \mathcal{Z}} \mathbb{E}_{\mathbb{Q}}[c(\mathbf{z}; \mathbf{y}) | \mathbf{x} = \mathbf{x}_0] = \min_{\mathbf{z} \in \mathcal{Z}} \mathbb{E}_{\mathbf{y} \sim \mathbb{Q}_{\mathbf{x}_0}}[c(\mathbf{z}; \mathbf{y})], \tag{2.1}$$

where v is the objective value, $\mathbf{z} \in \mathbb{R}^{d_z}$ is the decision vector, $\mathcal{Z}$ is a convex set of feasible solutions, $c(\cdot)$ is a convex cost function, $\mathbf{x}_0$ is a realization of $\mathbf{x}$, and $\mathbb{Q}_{\mathbf{x}_0}$ is the marginal distribution of $\mathbf{y}$ conditioned on $\mathbf{x} = \mathbf{x}_0$. In words, our goal is to solve a stochastic optimization problem conditioned on an out-of-sample realization of some features that are associated with the target uncertainty.

Classical stochastic optimization [12] examines problems with uncertain parameters assuming known distributions of uncertainty. In practice, however, the true distributions are unknown. Instead, we assume to have access to a training data set $\{(\mathbf{y}_i, \mathbf{x}_i)\}_{i=1}^n$ of n observations, which we can use to approximate (2.1).

The fundamental method of approximating (2.1) given a set of observations $\{\mathbf{y}_i\}_{i=1}^n$ (either empirical or sampled from an estimated distribution) is with Sample Average Approximation (SAA) [22]

$$\min_{\mathbf{z} \in \mathcal{Z}} \frac{1}{n} \sum_{i=1}^{n} c(\mathbf{z}; \mathbf{y}_i). \tag{2.2}$$

Although SAA enjoys several nice theoretical properties, such as consistency and asymptotic optimality, (2.2) does not leverage the available contextual information $\{\mathbf{x}_i\}_{i=1}^n$.

The standard modeling approach to leverage the available contextual information is to first employ a forecasting model $f \in \mathcal{F} : \mathcal{X} \to \mathcal{Y}$ that maps observations of $\mathbf{x}$ to $\mathbf{y}$, where $\mathcal{F}$ is a hypothesis space, and then solve a deterministic optimization problem. We term this two-step approach *forecast-then-optimize*. Typically, f belongs in the class of machine learning or statistical models and is trained by minimizing a surrogate loss $l(\mathbf{y}_i, f(\mathbf{x}_i))$ over the training data set $\{(\mathbf{y}_i, \mathbf{x}_i)\}_{i=1}^n$, such as the Mean Squared Error (MSE),

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} l(\mathbf{y}_i, f(\mathbf{x}_i)) = \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{y}_i - f(\mathbf{x}_i)\|_2^2. \tag{2.3}$$

Thus, f approximates $\mathbb{E}[\mathbf{y} | \mathbf{x} = \mathbf{x}_0]$, i.e., the conditional expectation of $\mathbf{y}$ given an observation of $\mathbf{x}$. The original problem (2.1) is then approximated by

$$\min_{\mathbf{z} \in \mathcal{Z}} c(z; \mathbb{E}[\mathbf{y} | \mathbf{x} = \mathbf{x}_0]) \approx \min_{\mathbf{z} \in \mathcal{Z}} c(z; f(\mathbf{x}_0)), \tag{2.4}$$

which is a deterministic problem and thus easier to solve. However, replacing the uncertainty with its conditional expectation is not equivalent to solving (2.1). Furthermore, (2.4) ignores the uncertainty due to potential forecast errors, which, in turn, may lead to significant out-of-sample disappointment.

In the forecast-then-optimize modeling framework, the process of training f and the subsequent optimization problem are treated separately. Recently, there has been significant effort in tackling the prescriptive analytics problem described in (2.1) in a holistic way [23].

Broadly, relevant research can be classified in three directions: *(i)* forecast-then-optimize under an alternative loss function, *(ii)* directly forecasting the solutions of the optimization problem, and *(iii)* approximating the conditional distribution $\mathbb{Q}_{\mathbf{x}_0}$.

The first approach proposes learning under alternative loss functions to derive forecasts that are cognizant of the downstream problem and explicitly minimize the decision cost. Let $\mathbf{z}^*(\mathbf{y}) \in \arg\min_{\mathbf{z}\in\mathcal{Z}} c(\mathbf{z};\mathbf{y})$ be an optimal solution of (2.1) for a realization of uncertainty $\mathbf{y}$. Instead of the true uncertainty $\mathbf{y}$, we use an estimation $\widehat{\mathbf{y}}$ derived from a forecasting model f. The goal is to develop a training model that minimizes the decision cost induced by erroneous predictions, which can be formulated as

$$\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} c(\mathbf{z}^*(f(\mathbf{x}_i));\mathbf{y}_i) - c(\mathbf{z}^*(\mathbf{y}_i);\mathbf{y}_i). \tag{2.5}$$

Training f by minimizing (2.5) might lead to predictions that differ significantly from those derived under the MSE loss function (2.3), e.g., they could be biased. We term this approach *value-oriented* forecasting. The challenge here is to embed the optimization problem within the learning process. Gradient-based methods usually assume a smooth objective function, as in an earlier work [24] that employs a specialized financial criterion as the loss function. An important milestone in this area is the introduction of differentiable optimization layers [25] that compute exact gradients for backpropagation by differentiating the optimality conditions of a Quadratic Programming (QP) problem; differentiable optimization layers are subsequently leveraged in [26] to develop a task-based learning approach with applications in energy storage arbitrage and grid scheduling. Concurrently, [27] investigates Linear Programming (LP) problems with unknown cost vectors and proposes the Smart Predict-then-Optimize (SPO) loss function that minimizes the true decision cost; a convex and differentiable surrogate of the SPO loss is further derived. Conversely, [28] directly trains decision trees to minimize the SPO loss. An alternative approach based on bilevel programming is presented in [29], where the lower problem computes the best decision given a forecast and the upper problem estimates the linear coefficients of a forecasting model that lead to minimum costs. In any case, training a value-oriented forecasting model might be challenging, as the loss function could be non-convex and discontinuous. Further, it is unclear how this approach would perform when $\mathbf{y}$ comprises uncertainty from different sources, which may be associated with a different set of features.

The second approach proposes forecasting models that directly predict the solutions of a (constrained) optimization problem. Formally, we consider a forecasting model $f : \mathcal{X} \to \mathcal{Z}$ that maps contextual information $\mathbf{x}$ to decisions $\mathbf{z}$, using the cost function as loss, given by

$$\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} c(f(\mathbf{x}_i);\mathbf{y}_i), \tag{2.6}$$

where $\mathbf{z}$ is replaced by a decision rule $f(\mathbf{x})$. For an out-of-sample observation $\mathbf{x}_0$ the optimal solution is computed directly from $\mathbf{z}_0 = f(\mathbf{x}_0)$, which is highly efficient and reduces inference time, as it bypasses the optimization solver. For instance, [30] proposes linear decision rules

to solve the newsvendor problem with contextual information, which effectively results in high-dimensional quantile regression. A significant drawback of this approach, however, is the lack of guarantees for the feasibility of decisions for out-of-sample observations.

The third approach follows a non-parametric approach based on SAA [22]. In this case, we first estimate a conditional probability distribution $\widehat{\mathbb{Q}}_{\mathbf{x}_0}$ which approximates $\mathbb{Q}_{\mathbf{x}_0}$. Then, we sample a number of scenarios from $\widehat{\mathbb{Q}}_{\mathbf{x}_0}$, and apply SAA. Here, contextual information is leveraged during the estimation step, which is commonly referred to as probabilistic forecasting in the energy forecasting literature. Along this line of work, [20] introduced the framework of *predictive prescriptions* that leverages a function that weights training observations and then solves a weighted SAA, given by

$$\min_{\mathbf{z}\in\mathcal{Z}} \sum_{i=1}^{n} \omega_{n,i}(\mathbf{x}_0) c(\mathbf{z}; \mathbf{y}_i), \tag{2.7}$$

where $\omega_{n,i}(\mathbf{x}_0)$ denotes weights, such that $\sum_{i=1}^{n} \omega_{n,i}(\mathbf{x}_0) = 1$ and $\omega_{n,i}(\mathbf{x}_0) \geq 0$, derived from local-learning, non-parametric algorithms. This class of algorithms includes, among others, nearest neighbors, decision trees, and kernel-based methods. The framework of [20] has found several extensions, e.g., adding robustness to deal with small data sets [31], dealing with multi-stage problems [32], and considering multi-stage problems with adjustable robust optimization [33]. In [34] the residuals induced by the SAA solution are used to infer decision uncertainty. Conversely, [35] directly works with observations of the joint distribution $\mathbb{Q}$ to derive an ambiguity set conditioned on contextual information, thereby bypassing the need for a learning model altogether.

In the framework of [20], weights $\omega_{n,i}(\mathbf{x}_0)$ are derived by training local learning methods in a standard way that minimizes prediction error, thus still ignoring the downstream problem. Subsequent work investigates integrating forecasting and optimization directly within this framework. Specifically, [36] and [37] leverage tree-based methods to combine the framework of [20] with learning under an alternative loss function, effectively using trees to learn a policy from data to decisions. Conversely, [38] takes an intermediate approach and proposes a validation method to select model hyperparameters that minimize the downstream decision cost. Nevertheless, such approaches still do not offer insight regarding the importance of each feature on the decision quality, thus largely remain a black box.

2.2.2 Power System Applications and Related Work

Jointly examining forecasting and optimization has also become a popular research area in power systems. Several works have considered the day-ahead unit commitment problem. An earlier work develops an asymmetric loss function to improve the value of day-ahead load forecasts [39]. In [40], a closed-loop forecast-and-optimize module is described for the same problem, employing the loss function introduced in [27], while [41] presents a bilevel model to jointly tune the forecasting model and solve the unit commitment and economic dispatch problem. Conversely, [42] examines the stochastic unit commitment problem with contextual

information and further proposes a task-based approach to tune model hyperparameters. A task-based load forecasting model that combines deep learning with stochastic economic dispatch is proposed in [43], following the work of [26]. In [44], an electricity price forecasting model is optimized to directly maximize economic benefits for a storage system performing arbitrage in electricity markets. In [45], a contextually-dependent distributionally robust formulation of the Optimal Power Flow (OPF) problem is developed, leveraging the fact that wind production forecasting errors depend on the magnitude of the forecast. Additionally, [46] considers wind forecasting for short-term trading applications and [47] examines load forecasting for dispatch scheduling, both relying on two-step approaches that involve first inferring a convex loss from data, then training the forecasting model. Conversely, [48] integrates the DC-OPF problem within a neural network model to derive adversarial load scenarios that are statistically credible and improve system resilience. In [49], a decision rule approach is presented for value-oriented demand forecasts to clear a day-ahead market, by considering the downstream balancing costs during learning. In [50], a value-oriented model that forecasts electricity market quantities is developed by employing a risk-averse trading strategy as an alternative loss function, with the subsequent forecasts leading to improved trading profit. However, this approach does not reduce the modeling effort and cannot handle multiple uncertainties, e.g., when both renewable production and market prices are uncertain. In [51], the framework put forward in [30] is extended by proposing linear decision rules to improve both the forecasting and trading performance of a Wind Power Plant (WPP) participating in a day-ahead market. In [52], the linear decision rule approach for trading wind production is further extended in an online learning setting. For a similar case study with Photovoltaic (PV) plants, [53] utilizes neural networks to directly forecast trading decisions. Nonetheless, these works deal with variations of the newsvendor problem and the proposed solutions cannot guarantee feasibility for problems with complex physical constraints. This issue can be circumvented by considering a discrete set of actions that approximate continuous decisions. For instance, [54] examines the control of a storage device formulated as a multi-label classification problem.

2.3 Methodology

This section formulates the problem of integrating forecasting and optimization (in Subsection 2.3.1), presents the proposed prescriptive trees method (in Subsection 2.3.2), adapts feature importance metrics in a prescriptive analytics context (in Subsection 2.3.3), and provides an illustrative example (in Subsection 2.3.4).

2.3.1 Embedding the Decision Cost in Learning

In this work, we focus on methods that estimate the distribution of uncertainty conditioned on some contextual information. Specifically, we focus on the framework established in

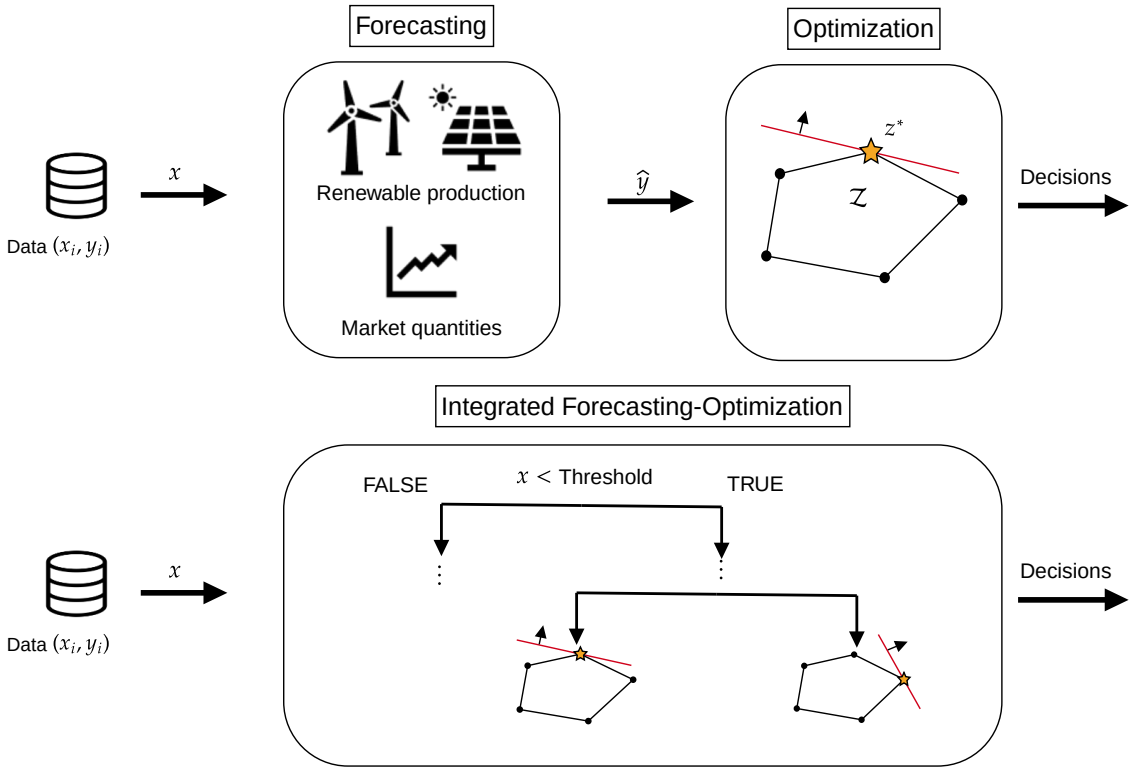


Figure 2.1: The standard two-step approach (top) and the proposed integrated approach (bottom).

[20], which proposes using weights $\omega_{n,i}(\mathbf{x}_0)$ derived by training a local learning method in a standard way, i.e., by minimizing a surrogate loss. In many relevant power system applications, we deal with uncertainty stemming from different sources, such as renewable production and market quantities. In turn, each uncertain parameter may be associated with a different set of features. A local learning algorithm would be agnostic to the impact of each source of uncertainty on the downstream decision cost and, thus, the standard training process would inevitably lead to suboptimal decision performance. To this end, we propose an optimization-aware training method that assesses the relative impact of each uncertain parameter and associated contextual information on the downstream costs during learning, while also exploiting possible cross-dependencies across variables.

Formally, we define the problem of searching over functions $f : \mathcal{X} \to \mathcal{Y}$ that improve prescriptive performance in the context of a weighted SAA of the form (2.7), given by

$$\min_{f\in\mathcal{F},\mathbf{z}(f,\mathbf{x}_i)\in\mathcal{Z}} \quad \sum_{i=1}^{n} c(\mathbf{z}(f,\mathbf{x}_i);\mathbf{y}_i), \tag{2.8a}$$

$$\text{s.t.} \quad \mathbf{z}(f,\mathbf{x}_i) = \arg\min_{\mathbf{z}\in\mathcal{Z}} \sum_{j=1}^{n} \omega_{n,j}^{f}(\mathbf{x}_i) c(\mathbf{z};\mathbf{y}_j), \qquad i = 1,\ldots,n, \tag{2.8b}$$

where $\mathbf{z}(f,\mathbf{x}_i)$ is the decision implemented for $\mathbf{x}_i$ under model f. In words, (2.8) finds a

forecasting model f that directly optimizes the decision cost for a contextual stochastic optimization problem approximated using a weighted SAA. Several non-parametric approaches have been adapted to approximate (2.8), e.g., k-nearest neighbors, kernel regression, and decision trees — see [37] for an overview. In the following, we focus exclusively on tree-based ensemble methods. The reason for this choice is twofold. First, tree-based ensemble methods perform exceptionally well in predictive tasks and have found success in several energy forecasting applications. Second, contrary to other local learning approaches, tree-based ensembles are fairly robust to noisy inputs of large dimensions. A conceptual overview of the different modeling approaches using decision trees is presented in Fig. 2.1.

2.3.2 Prescriptive Trees

Decision tree learning [55] is a popular machine learning algorithm, employed both for classification and regression tasks. Let $\tau : \mathbb{R}^{d_x} \to \{1, \dots, L\}$ be a map that corresponds to a disjoint partition of $\mathbb{R}^{d_x}$ into L leaves, so that $\tau(\mathbf{x})$ is the identity of the leaf that $\mathbf{x}$ falls into. In this work, we consider partitions created by following the popular Classification and Regression Trees (CART) [55] method, that recursively applies greedy binary splits to separate a region $\mathcal{R} \subseteq \mathbb{R}^{d_x}$ at feature $j \in [d_x]$ and point s into two disjoint partitions $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$, such that $\mathcal{R}_1 = \{i \in [n] \,|\, x_{ij} < s\}$ and $\mathcal{R}_2 = \{i \in [n] \,|\, x_{ij} \geq s\}$, where scalar x_{ij} denotes the i-th observation of the j-th feature. Each partition defines a tree node and observations that satisfy $x_{ij} < s$ fall to the left of the node, while the rest fall to the right[1].

To train decision trees in an optimization-aware way, at each node that we aim to split, we are searching for the pair (j, s) that minimizes

$$\min_{(j,s)} \Big(\min_{\mathbf{z}_1 \in \mathcal{Z}} \sum_{i \in \mathcal{R}_1} c(\mathbf{z}_1; \mathbf{y}_i) + \min_{\mathbf{z}_2 \in \mathcal{Z}} \sum_{i \in \mathcal{R}_2} c(\mathbf{z}_2; \mathbf{y}_i) \Big), \tag{2.9}$$

where the inner minimization problems correspond to the SAA solution of each partition, with $\mathbf{z}_1, \mathbf{z}_2$ being the locally constant decisions of the left and right child nodes. Thus, we search for a split that minimizes the decision cost function $c(\cdot)$, rather than the prediction error. We refer to a single tree trained by minimizing the split criterion in (2.9) as a *presrciptive tree*.

Note that problem (2.9) is of discrete nature and must be solved once per each candidate split for each node. The standard approach, following the CART method, is to order all observations per selected feature j, evaluate each candidate split point, and select the one that leads to the greatest error reduction. This approach benefits from the existence of an analytical solution to the internal minimization problems. In the regression setting, for instance, the SAA solutions in (2.9) equal the within leaf average, which can be updated recursively for all candidate splits. Furthermore, in the special case where $c(\mathbf{z}; \mathbf{y}) = \mathbf{y}^\intercal \mathbf{z}$, i.e., we have a linear cost function with uncertain cost coefficients, then (2.9) can be equivalently

[1] For ease of exposition, we focus exclusively on quantitative features. Note, however, that also including categorical features is straightforward.

written as

$$\min_{(j,s)} \Big(\sum_{i \in \mathcal{R}_1} c(\mathbf{z}(\overline{\mathbf{y}}_1); \mathbf{y}_i) + \sum_{i \in \mathcal{R}_2} c(\mathbf{z}(\overline{\mathbf{y}}_2); \mathbf{y}_i) \Big),$$

where $\overline{\mathbf{y}}_1, \overline{\mathbf{y}}_2$ denote the average of observations $\mathbf{y}_i$ that fall into leaves $\mathcal{R}_1, \mathcal{R}_2$, respectively, and $\mathbf{z}(\mathbf{y}_0) \in \arg\min_{\mathbf{z} \in \mathcal{Z}} c(\mathbf{z}; \mathbf{y}_0)$. That is, we estimate the average cost coefficients per leaf, solve a simpler, deterministic problem, and evaluate the cost function over all the observations in the leaf. This result follows from the linearity of the expectation operator and significantly simplifies the training process. For the general case, we need to call an optimization solver for each of the two SAA problems per each candidate split, and, depending on the structure of the underlying problem, this process might lead to a significant increase in computation time.

Overall, decision trees are prone to overfitting, i.e., they suffer from high variance, which significantly hinders their predictive capacity. Randomization-based ensemble methods address the overfitting issue and lead to impressive predictive performance. Popular methods include bootstrap aggregation (bagging), Random Forests [56], and Extremely Randomized Trees (ExtraTrees) [57]. Evidently, we can leverage these popular methods to train ensembles of prescriptive trees, which we refer to as *prescriptive forests*. However, if training a single prescriptive tree is computationally costly, training an ensemble is even costlier. To this end, we propose training ensembles that employ a randomized split criterion, following the paradigm of the ExtraTrees algorithm [57], which significantly decreases the number of candidate splits evaluated per node.

For a single prescriptive tree, we start from the top with a full training data set and recursively partition the feature space until no further improvements are possible or a stopping criterion is met. Typical stopping criteria include the maximum tree depth δ^{max} and the minimum number of observations n_{min} that fall at each leaf. At each node of each tree, we randomly select a subset of K features from $\mathbf{x}$ and for each feature randomly select a candidate split point within its range. Next, we estimate the aggregated cost of (2.9) for each candidate split and compare it with the cost at its root node, updating the tree structure accordingly. The process is repeated recursively until no further improvement is possible—see Algorithm 2.1 for a detailed description.

To derive prescriptions from a single tree, we can first estimate the corresponding weights $\omega_{n,i}(\mathbf{x}_0)$ for a new query $\mathbf{x}_0$ from

$$\omega_{n,i}(\mathbf{x}_0) = \frac{\mathbb{I}\left[\tau(\mathbf{x}_i) = \tau(\mathbf{x}_0)\right]}{\sum_{i'=1}^{n} \mathbb{I}\left[\tau(\mathbf{x}_{i'}) = \tau(\mathbf{x}_0)\right]}, \tag{2.10}$$

where $\tau(\mathbf{x}_0)$ returns the identity of the leaf that $\mathbf{x}_0$ falls into, and $\mathbb{I}[\cdot]$ is an indicator function. Then, we can use the estimated weights to solve (2.7). Nonetheless, the constant prescriptions for each leaf are already estimated when evaluating candidate splits — see (2.9). Therefore, a single prescriptive tree is fully compiled and provides a direct, piecewise constant mapping from features to decisions, while also ensuring feasibility.

Algorithm 2.1 `PrescriptiveTree`

Input: data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, current node $\mathcal{R}$, current depth δ, hyperparameters $\{n_{min}, K, \delta^{max}\}$

Output: prescriptive tree τ

determine cost at current node $v_0 = \min_{\mathbf{z} \in \mathcal{Z}} \sum_{i \in \mathcal{R}} c(\mathbf{z}; \mathbf{y}_i)$
set $v^* \longleftarrow v_0$, `split` $\longleftarrow$ `False`, $j^* \longleftarrow$ empty, $s^* \longleftarrow$ empty
if $\delta < \delta^{max}$ and $n \geq 2n_{min}$ **then**
  for $\kappa = 1, \ldots, K$ **do**
    sample a feature $j \in [d_x]$ without replacement
    sample a split point s from the range of feature x_j
    left child node: $\mathcal{R}_1 = \{i \in [n] \,|\, x_{ij} < s\}$
    right child node: $\mathcal{R}_2 = \{i \in [n] \,|\, x_{ij} \geq s\}$
    if $|\mathcal{R}_1| \geq n^{min}$ and $|\mathcal{R}_2| \geq n^{min}$ **then**
      $v = \Big(\min_{\mathbf{z}_1 \in \mathcal{Z}} \sum_{i \in \mathcal{R}_1} c(\mathbf{z}_1; \mathbf{y}_i) + \min_{\mathbf{z}_2 \in \mathcal{Z}} \sum_{i \in \mathcal{R}_2} c(\mathbf{z}_2; \mathbf{y}_i) \Big)$
      if $v < v^*$ **then**
        update $v^* \longleftarrow v$, `split` $\longleftarrow$ `True`, $j^* \longleftarrow j$, $s^* \longleftarrow s$
      end if
    end if
  end for
  if `split == True` **then**
    $\mathcal{D}_1 = \{(\mathbf{x}_i, \mathbf{y}_i) \,|\, i \in \mathcal{R}_1\}$
    $\mathcal{D}_2 = \{(\mathbf{x}_i, \mathbf{y}_i) \,|\, i \in \mathcal{R}_2\}$
    $\tau_1 =$ `PrescriptiveTree`$(\mathcal{D}_1, \mathcal{R}_1, \delta + 1)$
    $\tau_2 =$ `PrescriptiveTree`$(\mathcal{D}_2, \mathcal{R}_2, \delta + 1)$
    update tree structure τ
  end if
end if
return τ

For an ensemble $\{\tau_1, \ldots, \tau_B\}$ of B trees, we first estimate the weights from

$$\omega_{n,i}(\mathbf{x}_0) = \frac{1}{B}\sum_{b=1}^{B}\frac{\mathbb{I}\left[\tau_b(\mathbf{x}_i) = \tau_b(\mathbf{x}_0)\right]}{\sum_{i'=1}^{n}\mathbb{I}\left[\tau_b(\mathbf{x}_{i'}) = \tau_b(\mathbf{x}_0)\right]}, \tag{2.11}$$

which effectively is the average weight of all the trees in the ensemble. Next, the estimated weights $\omega_{n,i}(\mathbf{x}_0)$ are used to solve (2.7).

For the special case $c(\mathbf{z};\mathbf{y}) = \mathbf{y}^\intercal\mathbf{z}$, instead of (2.7), we can replace $\mathbf{y}$ with its point forecast

$$\widehat{\mathbf{y}}_0 = \sum_{i=1}^{n}\omega_{n,i}(\mathbf{x}_0)\mathbf{y}_i, \tag{2.12}$$

and solve a simpler, deterministic problem. As the weights $\omega_{n,i}(\mathbf{x}_0)$ are derived by minimizing the optimization cost, (2.12) effectively determines a value-oriented forecast of $\mathbf{y}$, which may differ considerably from the one derived from a standard tree-based method. Therefore, our proposed framework bridges two research directions on prescriptive analytics, namely value-oriented forecasting and directly learning decisions from data.

We further elaborate on our motivation behind selecting the random split criterion when training an ensemble of prescriptive trees. As discussed, the main computational cost of Algorithm 2.1 occurs during the evaluation of candidate splits. The motivating factor behind selecting the random split criterion lies in the expected reduction in computation time, as only a small number of splits are evaluated at each node. Computational experiments between the ExtraTrees and the Random Forest algorithm [57] suggest an average reduction in training time by a factor of 3 for $K = \sqrt{d_x}$, which can rise up to a factor of 10 for wider data sets (larger d_x). Regarding the ensemble size B, the generalization error is expected to monotonically decrease as B increases, thus the computation time is the main consideration for its selection. Note that the task of training an ensemble is trivially parallelizable. Similarly, the rest of the hyperparameters K, n_{min} represent an inherent trade-off between model capacity and computational costs (single trees are maximally grown, thus δ^{max} is set at infinity). The number of candidate splits K controls how strong individual splits are (for $K = 1$ splits are completely random), while larger values of n_{min} result in shallower trees (and reduced computations), with higher bias and lower variance.

2.3.3 Measuring the Impact of Data on Decisions

Explainability is pivotal to disseminating the results to industry stakeholders and enabling large-scale adoption of analytics tools in real-world applications. Here, our goal is to evaluate the impact of the various features on the efficacy of decisions, which is termed *prescriptiveness*. This evaluation is especially important in cases where obtaining contextual information incurs in and of itself additional costs, e.g., acquiring weather forecasts for multiple locations.

To this end, we adapt the well-known Mean Decrease Impurity (MDI) metric in a prescriptive analytics context. Provided a scoring rule that decides whether a node is split,

MDI measures the total decrease in node impurity (dissimilarity) weighted by the probability of reaching a specific node, averaged over the ensemble [58]. Considering a prescriptive tree node $\mathcal{R}_0$ partitioned at (j, s) into $\mathcal{R}_1, \mathcal{R}_2$, the decrease in aggregated cost is given by

$$\Delta v(j, s) = v(\mathcal{R}_0) - v(\mathcal{R}_1) - v(\mathcal{R}_2). \tag{2.13}$$

For an ensemble of B trees, the importance of feature j in terms of prescriptiveness, $\text{Imp}(j)$, is measured as the aggregated cost decrease over all the nodes that j defines the split variable, over all trees B in the ensemble:

$$\text{Imp}(j) = \frac{1}{B} \sum_{b=1}^{B} \sum_{\ell \in \mathcal{R}_{1:L} \,|\, j_\ell = j} p(b) \Delta v(j_\ell, s), \tag{2.14}$$

with $p(b) = \frac{|\mathcal{R}_\ell^b|}{n}$ being the proportion of observations reaching node $\mathcal{R}_\ell$ in tree b and j_ℓ the feature used for splitting that node. The MDI metric is estimated internally during training, therefore it can be obtained without additional computational cost.

We also consider measuring prescriptiveness by adapting the permutation importance technique proposed in [56]. First, we estimate aggregated costs with respect to the selected objective function over a hold-out set, which determines a base score. Next, we iterate over all the features, permutate (re-shuffle) each one, and derive new prescriptions, repeating the process a number of times. The permutation importance is then defined as the expected cost increase compared to the base score. In some cases, this approach may lead to a significant increase in computational costs, as prescriptions need to be re-optimized at each query. Therefore, we omit it from our experimental setup but note that it presents an attractive alternative if our model consists of a single prescriptive tree.

2.3.4 An Illustrative Example

To illustrate the proposed method, we examine a toy newsvendor problem [30]. Consider an uncertain demand y generated from

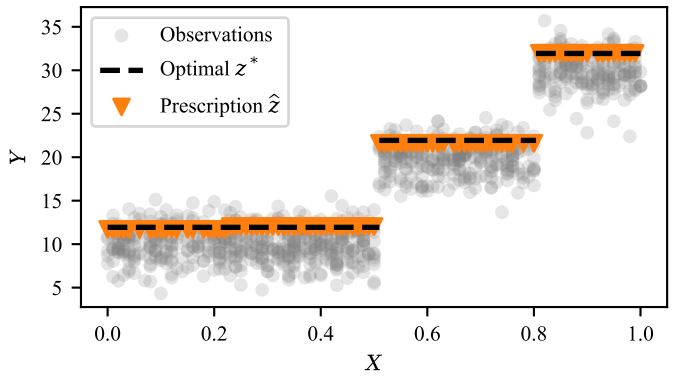
$$y = 10 + 10\mathbb{I}[x > 0.5] + 10\mathbb{I}[x > 0.8] + \epsilon,$$

where x is a single feature that is uniformly distributed in the interval $[0, 1]$ and ϵ is a random noise component that follows a normal distribution $N(0, 2)$. Further, assume that the cost function is given by

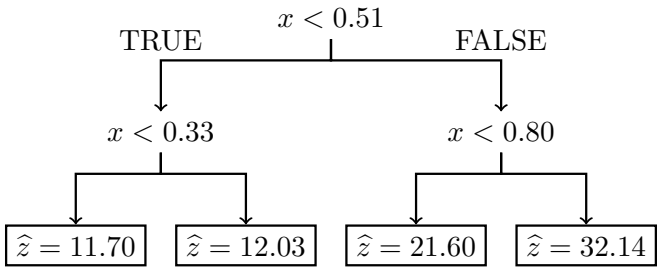
$$c(z; y) = 2(y - z)^+ + 10(z - y)^+,$$

where $(\cdot)^+ = \max(0, \cdot)$. For a realization x_0 of x, the optimal solution is given by the analytical formula $z^* = F_{x_0}^{-1}(\frac{10}{10+2})$, where $F_{x_0}^{-1}$ is the inverse cumulative distribution function of y given $x = x_0$.

We sample 1 000 observations and train a single prescriptive tree with $\delta^{max} = 2$, considering splitting a node at 99 equally spaced quantiles of the empirical distribution of



(a) In-sample fit and derived decisions.



(b) Prescriptive tree structure.

Figure 2.2: Illustrative example.

x. Fig. 2.2a presents a scatterplot with the in-sample fit, and Fig. 2.2b presents the tree structure. Indeed, both plots illustrate that the prescriptive tree learns a policy that is a piecewise constant function of x, with the tree nodes being split at the threshold where the indicator functions are activated. In turn, this leads to a learned policy that is a very good approximation of the true optimal decision, as highlighted in Fig. 2.2a.

2.4 Motivating Power System Applications

In this section, we describe a series of motivating power system applications that serve to validate the proposed methodology, primarily focused on participation in competitive electricity markets. First, we examine a storage arbitrage task with price uncertainty (in Subsection 2.4.1). Next, we consider the problem of deriving offers for a renewable producer participating in a day-ahead market with uncertain production and market quantities (in Subsection 2.4.2). Finally, we consider a more complex scenario that involves the aggregation of renewable plants and storage, where we jointly optimize the trading strategy and storage operation (in Subsection 2.4.3).

2.4.1 Price Arbitrage with Storage

We first examine the problem of scheduling a generic battery storage device to perform price arbitrage in a day-ahead market, inspired by [26]. The operator of a grid-scale storage

device decides the charging, p_t^{ch}, and discharging, p_t^{dis}, actions for each period t of the day-ahead horizon $T = 24$. The goal is to maximize profits while also accounting for battery degradation costs and penalizing excessive deviations from a reference state of charge. Both degradation costs and excessive deviations are modeled as quadratic regularization terms, controlled by design parameters γ and ϵ. The problem is given by

$$\min_{p_t^{\text{ch}}, p_t^{\text{dis}}, p_t^{\text{soc}}} \quad \mathbb{E}\left[\sum_{t=1}^{T} \pi_t^{\text{da}}(p_t^{\text{ch}} - p_t^{\text{dis}}) + \gamma \left\|p_t^{\text{soc}} - p_0\right\|_2^2 + \epsilon \left\|p_t^{\text{ch}}\right\|_2^2 + \epsilon \left\|p_t^{\text{dis}}\right\|_2^2\right], \tag{2.15a}$$

$$\text{s.t.} \quad p_{t+1}^{\text{soc}} = p_t^{\text{soc}} + \eta^{\text{ch}} p_t^{\text{ch}} - \frac{1}{\eta^{\text{dis}}} p_t^{\text{dis}}, \qquad t \in [T-1], \tag{2.15b}$$

$$0 \le p_t^{\text{ch}} \le c^{\text{ch}}, \qquad t \in [T], \tag{2.15c}$$

$$0 \le p_t^{\text{dis}} \le c^{\text{dis}}, \qquad t \in [T], \tag{2.15d}$$

$$0 \le p_t^{\text{soc}} \le B^{\max}, \qquad t \in [T], \tag{2.15e}$$

$$p_1^{\text{soc}} = p_T^{\text{soc}} = \frac{B^{\max}}{2}, \tag{2.15f}$$

where the expectation is taken with respect to the stochastic market prices π_t^{da}, p_t^{soc} denotes the induced state of charge, $\eta^{\text{dis/ch}}$ denotes the discharging/charging efficiency, $c^{\text{dis/ch}}$ denotes the discharging/charging limits, and $B^{\max}$ denotes the storage capacity. The problem constraints include the transition function for the induced state of charge (2.15b), technical limits on charging (2.15c), discharging (2.15d), and state of charge (2.15e), and constraints on the initial and final state of charge (2.15f). To approximate (2.15a), the storage operator uses a training data set $\{(\boldsymbol{\pi}_i^{da}, \mathbf{x}_i^{\text{market}})\}_{i=1}^n$ of n observations, where $\boldsymbol{\pi}_i^{da} \in \mathbb{R}^T$ denotes a sample path observation of length T and $\mathbf{x}_i^{\text{market}}$ denotes associated features. The standard modeling approach dictates first training a forecasting model, deriving point predictions for the day-ahead prices, and then optimizing the storage actions. In our proposed framework, we directly embed (2.15) in a tree-based ensemble, which is trained considering the impact of forecasts on decision cost.

2.4.2 Trading Renewable Production

In this section, we consider the problem of deriving optimal energy offers for an aggregation of renewable plants, namely wind and solar power plants, participating in a day-ahead market, and examine different market designs. This problem is more complex than the previous one in the sense that it involves two sources of uncertainty, namely the stochastic renewable production and unknown market quantities, and has been studied extensively over recent years. Earlier works consider deriving optimal offers based on probabilistic production forecasts [59–61]. Participation in adjustment markets and developing risk-averse strategies are examined in [62]. Jointly participating in energy and reserve capacity markets is studied in [16], while hedging against uncertainty by strategic reserve purchases is discussed in [63].

Trading using probabilistic forecasts of both renewable production and market quantities is investigated in [64]. The problem of trading in markets that feature a single-price balancing mechanism is studied in [65]. Finally, [66] examines coordinated trading with a generic energy system storage and renewable production plants.

Problem Description

We consider a renewable producer participating in a day-ahead market as a price-taker under different balancing mechanisms. Prior to market closure, the producer submits an energy offer p_t^{offer} for each clearing period t of the day-ahead market. During real-time operation, the system operator activates balancing reserves to maintain the demand-supply equilibrium and stabilize the system frequency. The system assumes two states, namely *short*, i.e., demand exceeds supply and upward regulation is required, and *long*, i.e., supply exceeds demand and downward regulation is required. Based on real-time production, the producer buys back (sells) the amount of energy shortage (surplus) in order to balance its individual position. In the following, we two present problem formulations that pertain to different balancing market designs. For simplicity, as temporal constraints do not apply, subscript t is dropped from the formulation.

Let p^{E} denote the renewable production, π^{da} the clearing price of the day-ahead market, and $\pi^{\uparrow/\downarrow}$ the marginal cost of activating upward/downward regulation services. Evidently, both the renewable production and the market quantities are unknown to the producer at the time of submitting offers in the market. Assuming market participants behave rationally, a shortage of supply leads to increased real-time marginal costs. In other words, we assume that if the system is short, it holds that $\pi^{\uparrow} \geq \pi^{\text{da}}$ and $\pi^{\downarrow} = \pi^{\text{da}}$; while if the system is long, then $\pi^{\downarrow} \leq \pi^{\text{da}}$ and $\pi^{\uparrow} = \pi^{\text{da}}$. Let us further define $\lambda^{\uparrow} = \max(0, \pi^{\uparrow} - \pi^{\text{da}})$ and $\lambda^{\downarrow} = \max(0, \pi^{\text{da}} - \pi^{\downarrow})$ as the respective upward and downward *unit regulation costs*. Evidently, it holds that $\lambda^{\uparrow} \cdot \lambda^{\downarrow} = 0$, i.e., only one of them (at most) assumes a value greater than zero for a given settlement period.

Single-price Balancing Mechanism

If the electricity market operates under a *single*-price balancing mechanism, then the uncertain trading profit, ρ^{single}, for each settlement period is given by

$$\begin{aligned}\rho^{\text{single}} &= \pi^{\text{da}} p^{\text{offer}} + \pi^{\uparrow}(p^{\text{E}} - p^{\text{offer}}) + \pi^{\downarrow}(p^{\text{E}} - p^{\text{offer}}) \\ &= \pi^{\text{da}} p^{\text{E}} - \underbrace{\Big[-\lambda^{\uparrow}(p^{\text{E}} - p^{\text{offer}}) + \lambda^{\downarrow}(p^{\text{E}} - p^{\text{offer}})\Big]}_{\text{imbalance cost}},\end{aligned} \tag{2.16}$$

which decomposes into the revenue from the day-ahead market and the imbalance cost. Note that the first term, i.e., the revenue from participating in the day-ahead market, does not depend on the producer's actions; thus, maximizing the trading profit ρ^{single} is equivalent to minimizing the imbalance cost term. The problem of minimizing the imbalance cost is

given by

$$\min_{p^{\text{offer}}} \quad \mathbb{E}\left[-\lambda^{\uparrow}(p^{\text{E}} - p^{\text{offer}}) + \lambda^{\downarrow}(p^{\text{E}} - p^{\text{offer}})\right], \tag{2.17a}$$

$$\text{s.t.} \quad p^{\min} \leq p^{\text{offer}} \leq p^{\max}, \tag{2.17b}$$

where the expectation is taken with respect to the joint distribution of uncertainty $\mathbf{y} = (p^{\text{E}}, \lambda^{\uparrow}, \lambda^{\downarrow})$, following the generic notation of Section 2.2.1. Since (2.17) is affine with respect to the decision variable p^{offer}, the optimal energy offer is derived analytically from

$$p^{\text{offer}*} = \begin{cases} p^{\min}, & \text{if } -\widehat{\lambda}^{\uparrow} + \widehat{\lambda}^{\downarrow} \leq 0, \\ p^{\max}, & \text{if } -\widehat{\lambda}^{\uparrow} + \widehat{\lambda}^{\downarrow} > 0, \end{cases} \tag{2.18}$$

where $\widehat{\cdot}$ denotes expected (forecast) values—see [60, Section II] for a proof. We interpret (2.18) as follows: the optimal offer equals zero if the system is expected to be short (note that typically $p^{\min} = 0$) and the nominal capacity if the system is expected to be long. Note that if the unit regulation costs are zero, then any energy offer is optimal; thus, without loss of generality, this case is merged with the case of the system being short. Therefore, to participate in the day-ahead market, the producer leverages point forecasts of the unit regulation costs, while renewable production does not affect the trading offer. However, following this trading strategy incurs great risks and could constitute market abuse; this motivates the design of a trading strategy that does not lead to excessive imbalances.

Dual-price Balancing Mechanism

Conversely, if the balancing market operates under a *dual*-price balancing mechanism, the profit equation (2.16) is modified to impose a non-arbitrage condition between the day-ahead and the balancing market. The single-period profit for a dual-price balancing mechanism is given by

$$\rho^{\text{dual}} = \pi^{\text{da}} p^{\text{E}} - \underbrace{\left[-\lambda^{\uparrow}(p^{\text{E}} - p^{\text{offer}})^{-} + \lambda^{\downarrow}(p^{\text{E}} - p^{\text{offer}})^{+}\right]}_{\text{imbalance cost}}, \tag{2.19}$$

where $(\cdot)^{-} = \min(\cdot, 0)$. Similarly to the case of a single-price balancing mechanism, the trading profit ρ^{dual} decomposes into the revenue from the day-ahead market, which does not depend on the producer's actions, and the imbalance cost. The key difference from the single-price case is that the imbalance cost term in (2.19) is always non-negative, which, in turn, means that no additional profit can be attained in the balancing market (i.e., no arbitrage). In contrast, under a single-price market design, deviations that help restore the system frequency result in negative imbalance costs, i.e., additional profit. The problem of minimizing the imbalance cost under a dual-price balancing mechanism is given by

$$\min_{p^{\text{offer}}} \quad \mathbb{E}\left[-\lambda^{\uparrow}(p^{\text{E}} - p^{\text{offer}})^{-} + \lambda^{\downarrow}(p^{\text{E}} - p^{\text{offer}})^{+}\right], \tag{2.20a}$$

$$\text{s.t.} \quad p^{\min} \leq p^{\text{offer}} \leq p^{\max}, \tag{2.20b}$$

where, again, the expectation is taken with respect to $\mathbf{y} = (p^{\mathrm{E}}, \lambda^{\uparrow}, \lambda^{\downarrow})$. Problem (2.20) is an instance of the well-known *newsvendor* problem [30], where the objective costs are also unknown. If the conditional probability distribution of p^{E} is known, or approximated via a probabilistic forecasting model, the optimal offer is derived analytically from

$$p^{\mathrm{offer}*} = \widehat{F}^{-1}\left(\frac{\widehat{\lambda}^{\downarrow}}{\widehat{\lambda}^{\downarrow} + \widehat{\lambda}^{\uparrow}}\right), \tag{2.21}$$

where $\widehat{F}^{-1}$ is the predicted inverse cumulative distribution function of p^{E}, and $\widehat{\lambda}^{\downarrow}, \widehat{\lambda}^{\uparrow}$ are the point forecasts of the downward and upward unit regulation cost, respectively. Note that (2.21) holds without assuming independence between energy production and unit regulation costs— see [60, Section III] for a proof. Thus, a producer trading in a market with a dual-price balancing mechanism leverages probabilistic forecasts of renewable production and point forecasts of unit regulation costs.

Balancing between Prescriptive and Predictive Performance

For both market design paradigms, the optimal offering strategy might incur a significant risk of excessive losses. Thus, producers may be willing to reduce their expected profit in order to hedge the financial risk [62]. To this end, we propose a hybrid trading strategy that balances the prescriptive cost and and the MSE, i.e., balances trading profit maximization and renewable production forecast accuracy. The proposed hybrid trading strategy is given by

$$\min_{p^{\mathrm{offer}}} \quad \mathbb{E}\left[(1-k)(-\rho^{\mathrm{single/dual}}) + k\left\|p^{\mathrm{E}} - p^{\mathrm{offer}}\right\|_2^2\right], \tag{2.22a}$$

$$\text{s.t.} \quad p^{\min} \leq p^{\mathrm{offer}} \leq p^{\max}, \tag{2.22b}$$

where the objective function (2.22a) minimizes a convex combination of (normalized) trading cost, which depends on the market design, and prediction error. In our numerical experiments, we directly embed (2.22) within the proposed tree algorithm, using a training data set $\{(p_i^{\mathrm{E}}, \lambda_i^{\downarrow}, \lambda_i^{\uparrow}, \mathbf{x}_i^{\mathrm{E}}, \mathbf{x}_i^{\mathrm{market}})\}_{i=1}^{n}$ of n observations, where $\mathbf{x}^{\mathrm{E}}, \mathbf{x}^{\mathrm{market}}$ denote features associated with the renewable production and the unit regulation costs, respectively. This trading strategy is interpreted as adding a regularization term that penalizes excessive deviations from the expected energy production, which we believe provides an intuitive trade-off, unlike other risk-averse formulations. This trade-off is controlled by design parameter k; specifically, for $k = 0$ we retrieve a purely prescriptive task (maximize trading profit), while for $k = 1$ we obtain a purely predictive task (minimize forecast error) with the standard MSE loss function.

2.4.3 Trading and Operating a Renewable-Storage System

Battery storage systems present a promising avenue to support the participation of renewable power plants in electricity markets and enhance their profitability [66], offering

functions such as arbitraging in day-ahead markets and compensating for deviations from the submitted schedule during real-time operation. Here, we consider an extension of the previous case studies by appending a generic storage device to the aggregation of renewable plants and jointly optimize the day-ahead offers, considering a closed system, and the operational control policy of the storage. We maintain a similar setting as before, i.e., the aggregation participates in a day-ahead market subject to imbalance penalties, considering a dual-price balancing mechanism. While participating in additional markets, such as intraday or offering balancing services, is not examined, the extension is straightforward. To optimize the operational control policy of the storage, we allow recourse (i.e., corrective) actions based on the realization of uncertainty. However, this leads to a multi-stage dynamic optimization problem; a tractable reformulation is provided by applying the linear decision rule approach [21], modeling real-time control actions as an affine function of uncertainty, in this case the renewable production forecasting error. For the rest of this section, index t is used to define a specific time period (scalar), while the absence of t defines a vector over the day-ahead horizon of length $T = 24$.

Let $\boldsymbol{\xi} \in \Xi \subseteq \mathbb{R}^T$ define the renewable production forecasting error for the day-ahead horizon, i.e., a sample path of length T, taking values in the uncertainty set Ξ. The uncertain renewable production is defined as $\mathbf{p}^{\mathrm{E}} = \widehat{\mathbf{p}}^{\mathrm{E}} + \boldsymbol{\xi}$, i.e., the expected value (forecast) $\widehat{\mathbf{p}}^{\mathrm{E}} \in \mathbb{R}^T$ plus the error term $\boldsymbol{\xi}$. The storage recourse actions are defined as an affine function of uncertainty. For instance, the decision vector for the charging actions is given by

$$\mathbf{p}^{\mathrm{ch}}(\boldsymbol{\xi}) = \widehat{\mathbf{p}}^{\mathrm{ch}} + \mathbf{D}^{\mathrm{ch}}\boldsymbol{\xi},$$

where $\widehat{\mathbf{p}}^{\mathrm{ch}} \in \mathbb{R}^T$ denotes the scheduled day-ahead charging decisions for the whole horizon T and $\mathbf{D}^{\mathrm{ch}} \in \mathbb{R}^{T \times T}$ is a linear coefficient matrix that maps realizations of uncertainty $\boldsymbol{\xi}$ to recourse actions and, thus, determines the operational policy of the storage. Note that $\mathbf{D}^{\mathrm{ch}}$ considers the whole history of errors over the period; to retain non-anticipativity we require $\mathbf{D}^{\mathrm{ch}}$ to be lower-triangular. Similarly, the decision vector for the discharging actions is given by

$$\mathbf{p}^{\mathrm{dis}}(\boldsymbol{\xi}) = \widehat{\mathbf{p}}^{\mathrm{dis}} + \mathbf{D}^{\mathrm{dis}}\boldsymbol{\xi},$$

with $\mathbf{D}^{\mathrm{dis}} \in \mathbb{R}^{T \times T}$ also being lower-triangular.

We consider a modified version of [66] and design a control policy that aims at minimizing the imbalance volume. For simplicity, we do not consider the balancing mechanism in the objective function. Nonetheless, the results presented in Section 2.5 will show that this presents a realistic application for a dual-price balancing mechanism. Following our previous formulation, we minimize a convex combination of trading performance and deviations from

the day-ahead offer. The problem is given by

$$\min_{\mathcal{P}} \quad \mathbb{E}\left[\sum_{t=1}^{T} -(1-k)\pi_t^{\mathrm{da}} p_t^{\mathrm{offer}} + k \left\| p_t^{\mathrm{out}} - p_t^{\mathrm{offer}} \right\|_2^2\right], \tag{2.23a}$$

$$\text{s.t.} \quad p^{\min} \leq p_t^{\mathrm{offer}} \leq p^{\max}, \qquad t \in [T], \tag{2.23b}$$

$$\widehat{p}_{t+1}^{\mathrm{soc}} = \widehat{p}_t^{\mathrm{soc}} + \eta^{\mathrm{ch}} \widehat{p}_t^{\mathrm{ch}} - \frac{1}{\eta^{\mathrm{dis}}} \widehat{p}_t^{\mathrm{dis}}, \qquad t \in [T-1], \tag{2.23c}$$

$$p_1^{\mathrm{soc}} = p_T^{\mathrm{soc}} = p_0, \tag{2.23d}$$

$$p_t^{\mathrm{out}} = p_t^{\mathrm{E}} + \xi_t + p_t^{\mathrm{dis}}(\boldsymbol{\xi}) - p_t^{\mathrm{ch}}(\boldsymbol{\xi}), \qquad t \in [T], \tag{2.23e}$$

$$0 \leq \mathbf{p}^{\mathrm{ch}}(\boldsymbol{\xi}) \leq c^{\mathrm{ch}}, \qquad \forall \boldsymbol{\xi} \in \Xi, \tag{2.23f}$$

$$0 \leq \mathbf{p}^{\mathrm{dis}}(\boldsymbol{\xi}) \leq c^{\mathrm{dis}}, \qquad \forall \boldsymbol{\xi} \in \Xi, \tag{2.23g}$$

$$0 \leq \mathbf{p}^{\mathrm{soc}}(\boldsymbol{\xi}) \leq B^{\max}, \qquad \forall \boldsymbol{\xi} \in \Xi, \tag{2.23h}$$

$$D_{ij}^{\mathrm{ch}} = 0, \qquad i \in [T], j \in [i+1, T], \tag{2.23i}$$

$$D_{ij}^{\mathrm{dis}} = 0, \qquad i \in [T], j \in [i+1, T], \tag{2.23j}$$

where $\mathcal{P} = \{\mathbf{p}^{\mathrm{offer}}, \widehat{\mathbf{p}}^{\mathrm{ch}}, \widehat{\mathbf{p}}^{\mathrm{dis}}, \mathbf{D}^{\mathrm{ch}}, \mathbf{D}^{\mathrm{dis}}\}$ is the set of decision variables, and $\mathbf{p}^{\mathrm{soc}}, \mathbf{p}^{\mathrm{out}}$ are auxiliary variables for the induced state of charge in the storage and the actual output of the plant-storage system. The expectation is taken with respect to $\mathbf{y} = (\boldsymbol{\pi}^{\mathrm{da}}, \mathbf{p}^{\mathrm{E}})$, i.e., the joint distribution of day-ahead prices and renewable production over the day-ahead horizon. The storage parameters are defined as in Section 2.4.1. The objective (2.23a) minimizes a convex combination of trading profit from the day-ahead market and deviations between actual output and the contracted energy. The trade-off is controlled with parameter k. For $k = 0$ the primary function of the storage is to arbitrage in the day-ahead market, while for $k = 1$ the focus is placed on compensating deviations from the schedule during real-time operation. The problem constraints include the limits for contracted energy (2.23b), the state transition equation of the storage (2.23c), initial and terminal conditions for the state of charge (2.23d), the definition of the total output of the system (2.23e), technical limits of the storage (2.23f)-(2.23h), and the non-anticipativity constraints (2.23i)-(2.23j).

To approximate (2.23), we use a training data set $\{(\boldsymbol{\pi}_i^{\mathrm{da}}, \mathbf{p}_i^{\mathrm{E}})\}_{i=1}^n$, alongside associated features; recall that the i-th observation denotes a sample path of length T. After subtracting the expected production $\widehat{\mathbf{p}}^{\mathrm{E}}$ from $\{\mathbf{p}_i^{\mathrm{E}}\}_{i=1}^n$, we can express the uncertainty with respect to renewable production in terms of forecasting error $\boldsymbol{\xi} \in \Xi = \{\boldsymbol{\xi}_i\}_{i=1}^n$. Note that (2.23f)-(2.23h) are robust constraints; to reformulate them, we employ duality theory and techniques from robust optimization [67].

For illustration, constraint (2.23f) is reformulated as follows. First, we define a polyhedral uncertainty set $\Xi' = \{\boldsymbol{\xi} \,|\, \mathbf{H}\boldsymbol{\xi} \leq \mathbf{h}\}$, where $\mathbf{H} = [\mathbf{I}, -\mathbf{I}]^\intercal \in \mathbb{R}^{2T \times T}$, with $\mathbf{I}$ being an identity matrix of appropriate size, and $\mathbf{h} \in \mathbb{R}^{2T}$ being a vector that contains the largest and smallest observed forecasting error for each period t. Constraint (2.23f) is equivalently written as

$$\max_{\boldsymbol{\xi}} \left\{ \widehat{\mathbf{p}}^{\mathrm{ch}} + \mathbf{D}^{\mathrm{ch}} \boldsymbol{\xi} \,|\, \mathbf{H}\boldsymbol{\xi} \leq \mathbf{h} \right\} \leq c^{\mathrm{ch}},$$

which is linear in $\boldsymbol{\xi}$. From duality, we equivalently write

$$\min_{\boldsymbol{\mu}} \left\{ \mathbf{h}^{\intercal}\boldsymbol{\mu} \,|\, \mathbf{H}^{\intercal}\boldsymbol{\mu} = \mathbf{D}^{\text{ch}}, \boldsymbol{\mu} \geq \mathbf{0} \right\} \leq c^{\text{ch}} - \widehat{\mathbf{p}}^{\text{ch}},$$

where $\boldsymbol{\mu}$ is a vector of dual variables of appropriate size. Evidently, the min operator becomes redundant, which finally leads to

$$\exists \boldsymbol{\mu}, \text{ with } \mathbf{h}^{\intercal}\boldsymbol{\mu} \leq c^{\text{ch}} - \widehat{\mathbf{p}}^{\text{ch}}, \; \mathbf{H}^{\intercal}\boldsymbol{\mu} = \mathbf{D}^{\text{ch}}, \boldsymbol{\mu} \geq \mathbf{0}, \tag{2.24}$$

which replaces the original constraint. The rest of the constraints are reformulated in a similar fashion. Under the standard modeling approach, the producer first generates temporally correlated scenarios for renewable production over the whole day-ahead horizon and expected values for day-ahead prices [68], then solves (2.23), after constraint reformulation. Conversely, in our proposed approach, we directly embed (2.23), after constraint reformulation, in a tree-based ensemble.

Note that if the uncertainty set Ξ' is too wide, no control will take place during real-time operation, while if it is too tight, it is possible to get infeasible actions. The robust formulation ensures control actions are feasible only for uncertainty realizations within the data-driven uncertainty set Ξ'. Evidently, during real-time operation, it is possible that a realization of uncertainty falls outside of Ξ', which may lead to infeasible recourse actions. To ensure that recourse actions are feasible in out-of-sample scenarios, we incorporate an additional saturation block. Specifically, the maximum charge is set as $\min(c^{\text{in}}, \frac{B^{\max} - \mathbf{p}^{\text{soc}}}{\eta^{\text{ch}}})$, while the maximum discharge is $\min(c^{\text{out}}, \mathbf{p}^{\text{soc}}\eta^{\text{dis}})$. Also, note that Ξ' varies on an hourly basis, based on the underlying samples $\boldsymbol{\xi}_i$; we illustrate this effect in the next section.

2.5 Numerical Experiments

This section presents our numerical experiments. First, we present our experimental setup (in Subsection 2.5.1) and discuss hyperparameter tuning (in Subsection 2.5.2). Next, we present the results for the price arbitrage problem (in Subsection 2.5.3), followed by the results for the problem of trading renewable production (in Subsection 2.5.4), and for the problem of jointly optimizing the trading of renewable production and the storage operation (in Subsection 2.5.5).

2.5.1 Experimental Setup, Input Data, and Forecasting Models

Experimental Setup

We first describe a common experimental setup that applies in all case studies. For all the problems considered, the following methods are compared:

- `FO`: The standard forecast-then-optimize sequential modeling approach. This involves training a separate forecasting model for each uncertain parameter and solving a stochastic optimization problem given appropriate forecasts.

- `PF`: The weighted SAA method (2.7) using a prescriptive forest with random splits, trained to minimize decision costs.
- `SAA`: The naive SAA solution (2.2) that ignores contextual information.
- `Oracle`: The perfect-foresight solution.

The specific forecasting models required for `FO` depend on the particular problem [68]. The problem of price arbitrage with storage (2.15) requires point forecasts of day-ahead prices; the trading problem (2.22) requires probabilistic forecasts of renewable production and point forecasts of unit regulation costs; the problem of jointly optimizing trading decisions and the operational control of storage the (2.23) requires trajectory (scenario) forecasts of renewable production over the whole day-ahead horizon and point forecasts of day-ahead prices.

In contrast, `PF` always uses a single model that takes as input the concatenation of available features from the individual forecasting models. The specific details of the forecasting models implemented are discussed in the following subsection.

During our experiments, we vary the different design parameters that appear in the objective functions, resulting in different optimization problems. For `PF`, we train a separate model for each value of the design parameters considered. Conversely, the forecasting models incorporated in `FO` are independent of the downstream problem, therefore they are trained once for all values of design parameters.

Moreover, we use `SAA` and `Oracle` to estimate a unitless metric that measures the relative prescriptive performance. Specifically, for each method i in $\{\texttt{FO}, \texttt{PF}\}$, we estimate the coefficient of prescriptiveness P [20] given by:

$$P_i = 1 - \frac{\widehat{v}^i - \widehat{v}^*}{\widehat{v}^{\texttt{SAA}} - \widehat{v}^*}, \tag{2.25}$$

where $\widehat{v}^i, \widehat{v}^{\texttt{SAA}}, \widehat{v}^*$ are the aggregated cost over the test set under methods i, `SAA`, and `Oracle`, respectively. The coefficient of prescriptiveness P is bounded above by one, while negative values indicate a failure to outperform `SAA`. Additional evaluation metrics are introduced in the respective results sections.

Input Data

For the numerical experiments, we consider data from the French electricity market, downloaded from [69], and production data from an aggregation of renewable plants consisting of 3 WPPs and 1 PV plant, with a total capacity of 49 MW (16% PV share), respectively located in northern and southern France. Both data sets span the same period, from January 2019 to April 2020. We use data from 2019 for training and validation, while data from 2020 is used for testing, assuming a half-hour settlement for the balancing market.

For the two case studies that include storage, we use a typical set of parameters, presented in Table 2.1.

Table 2.1: Storage device parameters, normalized by the nominal capacity of the renewable plants.

Parameter	Value
$B^{\max}$	0.5
c^{ch}	$0.5B^{\max}$
c^{dis}	$0.2B^{\max}$
η^{ch}	0.8
η^{dis}	0.9

Forecasting Models

To address the forecasting requirements of the different optimization problems of `FO`, we construct two sets of features: one related to renewable production and one related to market quantities. For renewable production, we construct a feature vector $\mathbf{x}^{\text{E}}$ that includes weather forecasts from a Numerical Weather Prediction (NWP) model, namely wind speed, wind direction, temperature, cloud coverage, and solar radiation forecasts for each plant location, resulting in a total of 10 features. The NWP model forecasts are issued at 00:00 on the day $D-1$ spanning a horizon of 24 to 48 hours ahead. We also check whether to include historical production lags as features by examining the Partial Autocorrelation Function (PACF). Since the PACF does not reveal any important lags, we do not include any in $\mathbf{x}^{\text{E}}$; this result is standard when the forecast horizon is larger than a couple of hours ahead.

For market-related quantities, we construct feature vector $\mathbf{x}^{\text{market}}$, which includes historical lags for the day-ahead electricity prices indicated by the PACF (one day and one week prior), historical lags for system imbalance volumes (two days prior), and day-ahead forecasts for available thermal generation, electricity demand, and renewable generation at the transmission level. The system-wide forecasts issued by the system operator are processed to determine a net load series, by subtracting the expected renewable production from the expected electricity demand, and a system margin series, defined as the ratio of net load to available thermal generation. Additionally, we include categorical variables to model the calendar effect, namely the day of the week and the hour of the day, resulting in a total of 7 features.

In all experiments, we consider a forecast horizon of 12 to 36 hours ahead as is standard in market-related applications. For `FO`, we always train univariate forecasting models, i.e., each model outputs a prediction for a single time period t of the day-ahead horizon. Conversely, if the optimization problem involves the full day-ahead horizon, we reshape data accordingly in sample paths of length T and feed it into the `PF` model. In the following, we discuss the forecasting models implemented for each case study.

Price arbitrage with storage (2.15) For `FO`, point forecasts of day-ahead prices π_t^{da} are required for each period t of the forecast horizon. To this end, we use the ExtraTrees [57] method and features $\mathbf{x}^{\mathrm{market}}$. For `PF`, we train a model with vector output of length T, reshaping features $\mathbf{x}^{\mathrm{market}}$ accordingly.

Trading renewable production (2.22) For `FO`, probabilistic forecasts of renewable production and point forecasts of the unit regulation costs are required for each period t of the forecast horizon. To generate probabilistic forecasts of renewable production, we use features $\mathbf{x}^{\mathrm{E}}$ and train a Quantile Regression Forests (QRF) [70] model, which is an extension of the Random Forest [56] method that achieves state-of-the-art performance [71] in probabilistic forecasting.

To forecast the unit regulation costs, the standard practice is to partition the problem into three forecasting tasks, namely forecasting the magnitude of the upward unit regulation cost, forecasting the magnitude of the downward unit regulation cost, and forecasting the probability of the system being short or long. The individual forecasts are then combined accordingly to the requirements of the specific market design. Formally, the three forecasts are given by

$$\widehat{\phi} = \mathbb{P}(\lambda^{\uparrow} > 0), \tag{2.26a}$$

$$\widehat{\lambda}^{\uparrow} = \widehat{\phi}\mathbb{E}\left[\lambda^{\uparrow} | \lambda^{\uparrow} > 0\right], \tag{2.26b}$$

$$\widehat{\lambda}^{\downarrow} = (1-\widehat{\phi})\mathbb{E}\left[\lambda^{\downarrow} | \lambda^{\downarrow} > 0\right], \tag{2.26c}$$

where $\widehat{\phi}$ denotes the estimated probability of the system being short. Therefore, the prediction for the upward unit regulation cost $\lambda^{\uparrow}$ equals the expectation of a regression model trained conditionally on the system being short, weighted by probability $\widehat{\phi}$. Following [72], we apply exponential smoothing to model each of the individual components. For `PF`, we concatenate feature vectors $\mathbf{x}^{\mathrm{market}}$ and $\mathbf{x}^{\mathrm{E}}$ and train a single prescriptive forest.

Trading and operating a renewable-storage system (2.22) For `FO`, multivariate probabilistic forecasts of renewable production over the day-ahead horizon, i.e., trajectory or scenario forecasts, and point forecasts of day-ahead prices are required. To generate scenario forecasts, we implement a two-step process. First, we use the QRF model to derive marginal predictive densities for each period t of the day-ahead horizon. Then, we estimate the in-sample correlation across periods and employ a Gaussian copula function to generate correlated scenarios of length T, following the procedure detailed in [73]. For `PF`, we concatenate feature vectors $\mathbf{x}^{\mathrm{market}}$ and $\mathbf{x}^{\mathrm{E}}$ and reshape them accordingly to create sample paths of length T.

For illustration, Fig. 2.3 plots the different types of renewable production forecasts utilized in the experiments, namely point forecasts, probabilistic forecasts (in the form of prediction intervals), and temporally correlated scenario forecasts. Fig. 2.3 also illustrates

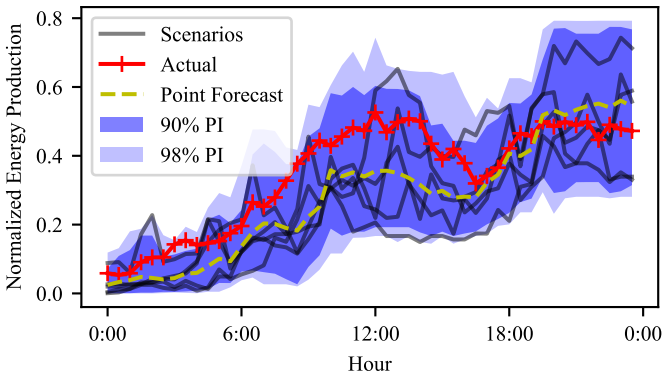


Figure 2.3: Example of day-ahead renewable production forecasts: point forecasts, probabilistic forecasts (prediction intervals or PI), and scenarios.

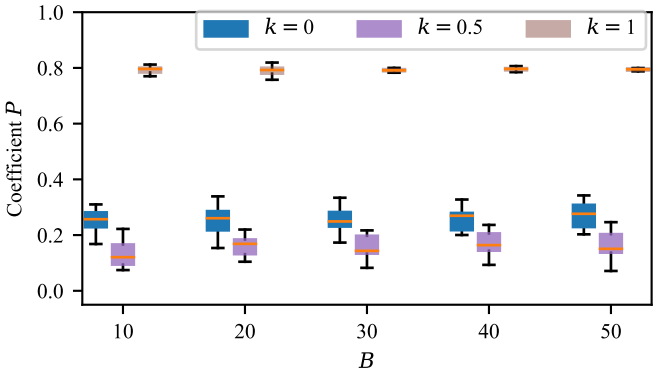
how the uncertainty set Ξ' varies for the combined renewable-storage case study, based on the underlying scenario forecasts. At 00:00, the scenarios exhibit small dispersion, which results in tighter upper and lower bounds. In contrast, at 12:00, the derived bounds are wider due to the larger dispersion of the underlying scenarios.

For all tree-based forecasting models, we train a large number of trees (300) and use default hyperparameter settings. For `PF`, we discuss the impact of hyperparameters selection in detail in the next section.

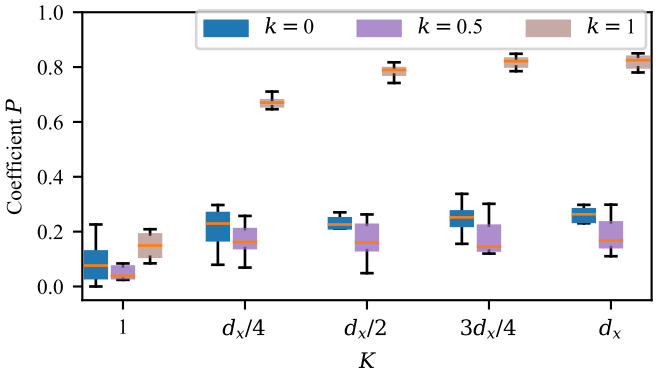
2.5.2 Effect of Hyperparameters and Split Algorithm

Before presenting the results for each case study, we first examine the performance of the proposed tree-based method with respect to hyperparameters $\{B, K, n_{\min}\}$ in a controlled setting. Specifically, we consider the problem of trading in a day-ahead market under a single-price balancing mechanism (2.22) as a test bed and examine prescriptive performance for values of $k = \{0, 0.5, 1\}$ by randomly sampling 1 000 training and test observations and estimating the coefficient of prescriptiveness P for each value of k. The process is repeated 10 times.

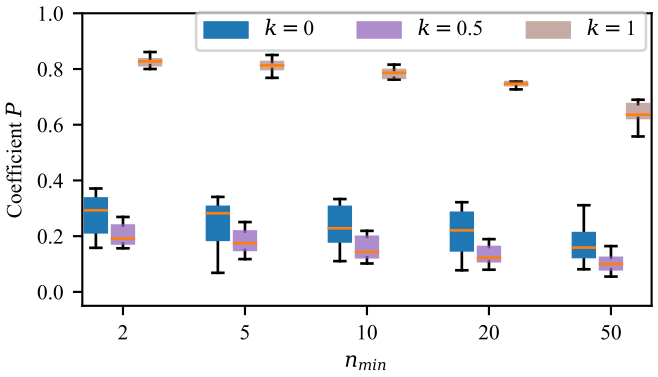
Fig. 2.4 presents the prescriptive performance as a function of the model hyperparameters. Specifically, Fig. 2.4a (top) plots the performance versus the ensemble size B for the different values of k, indicating that the decision performance is rather insensitive to the number of trees within the ensemble, as results are similar across the different tasks. A large discrepancy across the values P for the different values of k is observed, which is attributed to the relative difficulty of each underlying problem. For $k = 1$, i.e., a standard regression task, P is over typically over 0.80, while for $k = 0$, i.e., focusing solely on trading cost, P is less than 0.30. In other words, the regression task is relatively "easier", as it achieves performance closer to the one derived from the perfect foresight solution. Next, we examine the effect of the number of splits evaluated per node K, which controls the model's capacity. For $K = 1$ node splits are completely random (requiring minimum computations), while for



(a) Ensemble size B ($K = d_x/2, n_{min} = 5$).



(b) Number of splits K ($B = 25, n_{min} = 5$).



(c) Leaf size n_{min} ($B = 25, K = d_x/2$).

Figure 2.4: Effect of hyperparameters $B, K,$ and n_{min}.

$K = d_x$ all features are considered when splitting a node. From Fig. 2.4b (middle), it is evident that the impact of K on model performance is significant. The impact of K is more pronounced for the predictive task ($k = 1$), with the coefficient P ranging from below 0.20 (for $K = 1$) to over 0.80 (for $K = d_x$). The effect is similar, although less pronounced, for the rest of the tasks, with higher values of K leading to consistently improved prescriptive performance. Next, we examine the impact of the minimum leaf size $n_{\min}$. Generally, smaller values of $n_{\min}$ result in lower bias, while larger values provide a smoothing effect. Fig. 2.4c (bottom) indicates a decrease in performance for values of leaf size greater than 10, with the effect being more pronounced for the predictive task ($k = 1$). For the rest of the experiments, all results are obtained with hyperparameters $K = 3d_x/4, B = 50$, and $n_{\min} = 10$.

The selection of the tree-learning algorithm can also be viewed as a hyperparameter. To examine its effect on model performance and computational cost, we repeat the above experiment for $k = 0.5$ and examine three methods. Namely, we consider ordering observations and evaluating all candidate splits as in Random Forests (`RF`), evaluating candidate splits on 10 equally spaced quantiles of the empirical distribution of each feature (`RF-Q`), and random splits as in ExtraTrees (`ET`). Note that the effect of the hyperparameters $\{B, K, n_{\min}\}$ may vary for the different algorithms. Hence, we are not primarily interested in an exhaustive comparison in terms of prescriptive performance but rather want to highlight the effect of the selected algorithm on computational costs for a specific set of hyperparameters.

Table 2.2 presents results in terms of prescriptive performance and average CPU time to train a single tree over 10 iterations using a standard machine with an Intel Core i7 CPU with a 2.3GHz clock rate and 32GB of RAM. We observe that the random split criterion shows a significant reduction in computation time, both against `RF` and `RF-Q`, without compromising prescriptive performance. Evidently, the computational cost is associated with the underlying optimization problem. In this experiment, the problem is relatively simple; for larger problems (e.g., including storage) `RF` might become intractable.

2.5.3 Results for Price Arbitrage with Storage

This section presents results on the problem of price arbitrage with storage (2.15). Table 2.3 presents the results obtained for different values of the design parameters γ, ϵ that control the regularization penalties. We compare `FO`, `PF` in terms of prescriptive performance, measured by the coefficient of prescriptiveness P, and predictive performance for electricity price forecasting, measured in terms of Mean Absolute Error (MAE). Recall that the forecasting model incorporated in `FO` does not depend on the downstream optimization problem, hence the constant MAE values in Table 2.3.

From Table 2.3, we observe that improved predictive performance does not translate to improved decisions with respect to the decision cost defined in the objective function (2.15a). Indeed, while `FO` leads to an approximately 43% lower MAE on average, `PF` significantly

Table 2.2: Average performance (±one standard deviation) for sample size $n = 1\,000$.

	RF	RF-Q	ET
Coefficient P	0.16 ±0.08	0.18 ±0.05	0.16 ±0.04
Single tree CPU time (sec)	650.58 ±103.84	26.43 ±1.80	2.15 ±0.24

Table 2.3: Results for storage arbitrage.

	$\gamma, \epsilon = 0.01$		$\gamma, \epsilon = 0.1$		$\gamma, \epsilon = 1$	
	FO	PF	FO	PF	FO	PF
MAE (Eur/MWh)	8.07	11.58	8.07	11.77	8.07	11.31
Coefficient P	0.25	0.25	0.23	0.34	0.37	0.49

outperforms FO in the true task as indicated by the coefficient of prescriptiveness P, with the effect being more pronounced for higher values of γ and ϵ. Therefore, the forecasts derived from PF are oriented towards maximizing forecast value, leading to an expected reduction in the decision cost of approximately 10%.

2.5.4 Results for Trading Renewable Production

This section presents our results on the problem of trading renewable production in a day-ahead market (2.22), considering both a single-price and dual-price balancing mechanism. To evaluate trading performance, in addition to the coefficient of prescriptiveness P, we further estimate the aggregated trading profit and trading risk. For the latter, we use the conditional value at risk at 5% level ($\text{CVaR}_{5\%}$) as a proxy, defined as the expected profit over the 5% worst-case returns.

Single-price Balancing Mechanism

First, we examine results for a single-price balancing market. Regarding the effect of the hybrid trading strategy, we observe that larger values of k lead to more conservative offers and thus to a higher $\text{CVaR}_{5\%}$. This result is expected, as the minimization of the imbalance volume is weighted more heavily in the objective function as k increases. Fig. 2.5 illustrates this effect, with trading offers showing larger deviations from actual production as k decreases. Table 2.4 presents aggregated trading results for $k = \{0, 0.25, 0.5, 0.75, 1\}$, with PF leading to an expected profit increase of 3.82% across all values of k, with a maximum of profit increase of 7.44% for $k = 0.25$. Fig. 2.6 further highlights the improved risk-reward trade-off of PF compared to FO, as it sets the efficient frontier, i.e., leads to higher revenue for a given level of risk and vice versa.

These results are further validated by examining the coefficient of prescriptiveness P, which compares PF and FO to a benchmark without features (SAA) and the perfect-foresight

Table 2.4: Results for renewable trading, single-price market.

	$k = 0$		$k = 0.25$		$k = 0.50$		$k = 0.75$		$k = 1$	
	FO	PF	FO	PF	FO	PF	FO	PF	FO	PF
Total Profit (10^3 EUR)	1 191	1 250	1 170	1 257	1 170	1 225	1 182	1 212	1 184	1 178
$\text{CVaR}_{5\%}$ (EUR)	-442.44	-353.51	-403.50	-281.84	-243.29	-228.68	-119.56	-132.32	-92.45	-105.12
Coefficient P	0.06	0.15	-0.01	0.13	-0.01	0.08	0.11	0.17	0.85	0.85

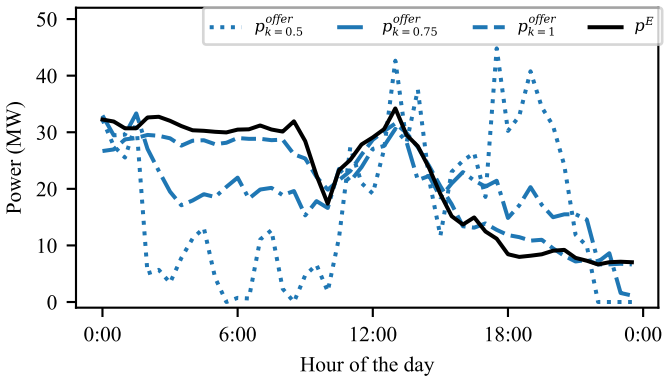


Figure 2.5: Illustration of actual production and different day-ahead offers for a single day.

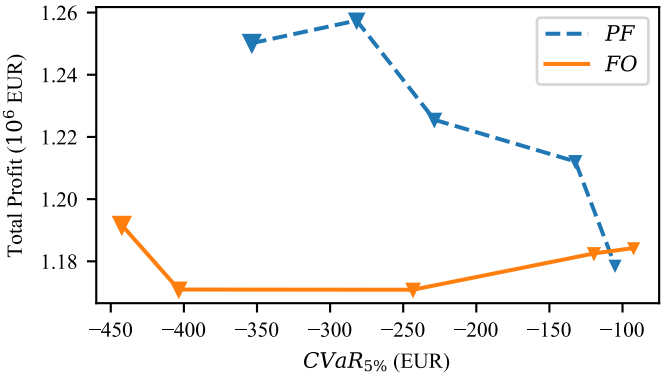


Figure 2.6: Risk versus reward for trading in a single-price market. Marker size is analogous to k. Values towards the top and right are preferred.

solution (`Oracle`). Overall, `PF` outperforms `SAA` consistently, as P is larger than 0 for all values of k. In contrast, `FO` fails to outperform `SAA` for lower values of k, with the respective P being close to 0 or even negative. Both `PF` and `FO` converge to similar performance for $k = 1$; this result is expected, as the prescriptive forest algorithm converges to a standard tree-based method for a regression task. Regarding relative performance against `Oracle`, we observe that P is significantly lower than 1 for all tasks except the standard regression. This result highlights that trading in the day-ahead market under a single-price balancing mechanism is a relatively more demanding task than standard renewable production forecasting, as the relative distance from `Oracle` is larger. We attribute this result to the fact that maximizing trading profit requires forecasting the unit regulation costs in a 12 to 36 hours ahead horizon,

which, in practice, is known to be extremely difficult. Nonetheless, our results manage to quantify this empirical knowledge, which we believe to be of use to both researchers and other stakeholders.

Dual-price Balancing Mechanism

Next, we examine trading performance under a dual-price balancing mechanism, with Table 2.5 presenting aggregated results. Overall, we observe that trading performance is rather insensitive to the choice of design parameter k, which contrasts the previous results for the single-price balancing mechanism. Indeed, trading profit is similar regardless if we consider an optimal trading strategy ($k = 0$) or we just offer the expected energy production ($k = 1$). This is attributed to two reasons, namely the non-arbitrage condition imposed by the market design, and the fact that the upward and downward regulation costs do not differ significantly for the specific data set. Nonetheless, `PF` leads to an expected profit increase of 0.62% compared to `FO`, which is also associated with a reduced modeling effort, as with `PF` we employ a single data-driven model and avoid multiple forecasting models. Both `PF` and `FO` consistently outperform `SAA`, as the lowest value of the prescriptive coefficient P is 0.62. Moreover, the values of P for $k = 0$ are significantly larger than the ones achieved under a single-price balancing mechanism, which indicates that trading under a dual-price balancing mechanism is a relatively "easier" task.

Prescriptive Feature Importance

Next, we investigate how the different features affect the prescriptive performance of `PF`, as measured by the adapted MDI method. To this end, a subset of the most important features is plotted in Fig. 2.7, with the aggregated feature importance normalized to add up to one. Considering a single-price balancing mechanism, we observe that for lower values of k, market-related features that associate with the estimation of the unit regulation costs achieve higher importance. This is attributed to `PF` placing more weight on the trading cost term in the objective function. Specifically, adding the individual feature importance of the expected system margin (Margin), expected net load (Net Load Forecast), the expected temperature at the WPP site (Temp_WPP), and the lagged observations for system imbalance volume (Volume_lag96), leads to approximately 65% of the total feature importance for $k = \{0, 0.25, 0.5\}$. Note that the WPPs are located in close proximity to large metropolitan areas and interconnections with neighboring countries, thus Temp_WPP effectively serves as a proxy for electricity demand. As k increases, the importance of features related to renewable production forecasting gradually increases, with the expected wind speed at the WPP site (WindSpeed_WPP) reaching approximately 75% of the total feature importance for $k = 1$.

Under a dual-price balancing mechanism, we observe significantly fewer variations in feature importance across the different values of k, which qualitatively resembles the re-

Table 2.5: Results for renewable trading, dual-price market.

	$k = 0$		$k = 0.25$		$k = 0.50$		$k = 0.75$		$k = 1$	
	FO	PF	FO	PF	FO	PF	FO	PF	FO	PF
Total Profit (10^3 EUR)	1 130	1 137	1 130	1 140	1 130	1 140	1 130	1 141	1 141	1 138
$CVaR_{5\%}$ (EUR)	-97.29	-106.30	-97.30	-99.98	-97.27	-104.14	-97.30	-104.98	-99.94	-107.46
Coefficient P	0.62	0.66	0.63	0.68	0.64	0.69	0.66	0.72	0.86	0.86

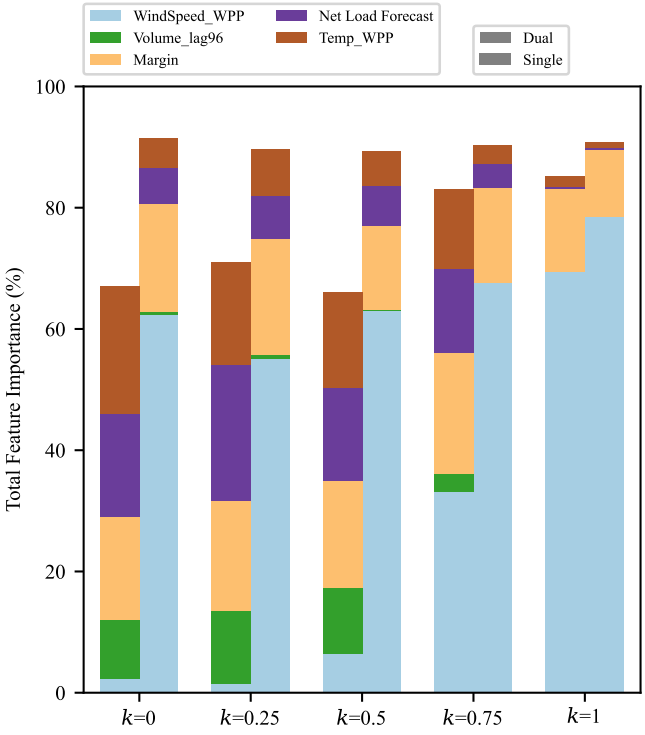


Figure 2.7: Normalized prescriptive feature importance for a subset of features.

sults presented in Table 2.5. Specifically, the expected wind speed at the WPP location (WindSpeed_WPP) is consistently the most important variable, with its feature importance ranging from 60% to 78%. Previous works on similar case studies mention that renewable forecasting is relatively more important than price forecasting when trading under a dual-price balancing mechanism [51]. The results presented in Table 2.5 and Fig. 2.7 provide quantitative evidence for these assertions by jointly considering the two sources of uncertainty in the problem formulation and measuring the impact of different features.

Finally, comparing feature importance across the two market designs indicates that forecasting market-related quantities, i.e., the unit regulation costs, is relatively more important under a single-price balancing mechanism. Conversely, renewable production forecasting should be the primary focus for producers participating in markets with a dual-price balancing mechanism.

2.5.5 Results for Trading and Operating a Renewable-Storage System

This subsection presents results for the problem of adding a storage device in an aggregation of renewable plants. Recall that we assume participation in a market with a dual-price balancing mechanism. As illustrated by the results presented in Section 2.5.4, in practice there is no significant difference between the optimal offering strategy and offering the expected energy production under such a market design. Therefore, the implemented operational control policy, i.e., using the storage device to minimize deviations from the submitted schedule, also makes sense from an economic perspective. Fig. 2.8 plots the estimated coefficient matrix $\mathbf{D}^{\text{ch}}$ for the `PF` model, which illustrates the potential corrective charging actions based on realized forecast error, for each hour of the day. For instance, if the actual production is underestimated at 05:00, a corrective charging action is implemented — see Fig. 2.9 for an illustration of how the implemented control policy mitigates the total imbalance volume, by taking corrective actions given the realization of uncertainty. Note that our goal in this section is not to evaluate the performance of the specific control policy; rather, given a specific control policy imposed, our goal is to evaluate the relative effect of different modeling approaches.

Table 2.6 presents the overall results for $k = 0.75$. Specifically, `PF` leads to an expected profit increase of 3.07% compared to `FO`, accompanied by an additional improvement in terms of $\text{CVaR}_{5\%}$. Moreover, both `PF` and `FO` perform significantly better than `SAA` and close to `Oracle`, with an average coefficient of prescriptiveness P of approximately 0.91. Compared to trading without storage—see Table 2.5— we observe that trading profits are significantly higher, namely 47% for `PF` and 44% for `FO` for $k = 0.75$. This result highlights the ability of the storage device to support renewable energy sources in market applications and further validates the applicability of the proposed control policy.

2.6 Conclusions

This chapter presented an integrated forecasting and optimization approach to maximize forecast value and enable improved decision-making, with a view toward applications in power systems and electricity markets. We developed tree-based algorithms that minimize task-specific costs for contextual stochastic optimization problems, employing a random split criterion to reduce computational costs. Further, we formulated a generic framework to measure the importance of features on decision efficacy under different objective functions.

The proposed approach was validated in different applications related to participation in electricity markets. In a price arbitrage problem with a storage device, the proposed approach led to a 10% improvement in terms of decision cost, even though it showcased worse forecast accuracy. In a problem of short-term trading of renewable production under different balancing mechanisms, the proposed approach led to an average increase in aggregate profit of 3.82% and 0.62% compared to the standard forecast-then-optimize modeling

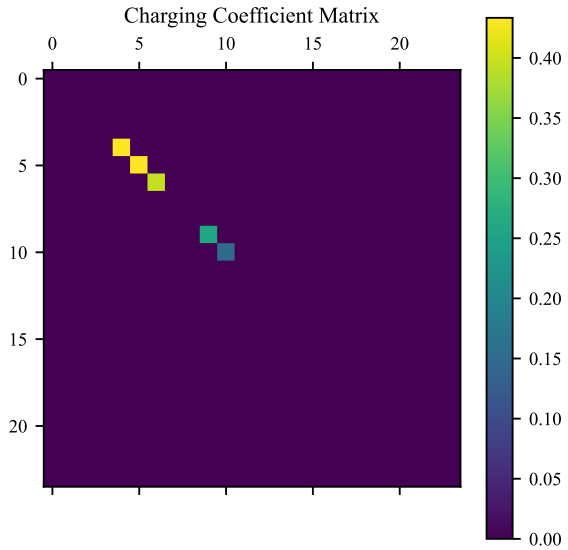


Figure 2.8: Estimated matrix $\mathbf{D}^{\text{ch}}$.

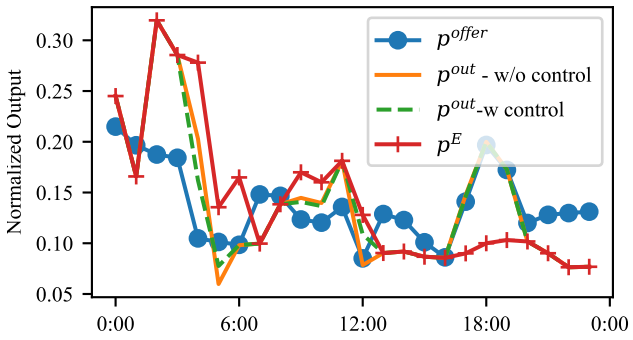


Figure 2.9: Example of trading offer and actual output of the aggregation for a single day.

Table 2.6: Results for trading and operating a storage device.

	$k = 0.75$	
	FO	PF
Total Profit (10^3 EUR)	1 628	1 678
$\text{CVaR}_{5\%}$ (EUR)	-8.88	-6.12
Coefficient P	0.89	0.92

approach, considering a market under a single- and dual-price balancing mechanism, respectively. In a more complex problem, we combined renewable plants with a generic storage system and coordinated the storage operation and the renewable trading problem. In this case, the proposed approach led to a 3.07% profit increase compared to the standard mod-

eling approach. Overall, we observed consistently better or similar prescriptive performance against the current state of the art, which was also associated with a significant reduction in modeling effort. Moreover, we examined feature importance under different objectives and across different market designs, demonstrating the capability of the proposed solution to evaluate the impact of feature data on decision quality, and provided insights on the trading of renewable production under different regulatory frameworks.

Future work could focus on extending the proposed methodology in an online learning setting, enabling adaptation to potential distribution shifts. Moreover, it is interesting to consider a setting where the model adapts to changes in the underlying problem structure, such as a different objective function or adding new constraints.

Chapter 3

An Interpretable Machine Learning Approach to Forecast Optimization Solutions

Résumé en Français

L'incertitude accrue due à l'intégration des sources d'énergie renouvelables stochastiques nécessite de résoudre les problèmes de flux de puissance optimal (OPF) à plusieurs reprises et pour un grand nombre de scénarios. Les méthodes d'apprentissage automatique ont un potentiel important pour réduire le temps de calcul des problèmes OPF en apprenant un mappage des charges d'entrée variables aux décisions, contournant ainsi le besoin d'un solveur d'optimisation lors de l'inférence. Cependant, les méthodes actuelles d'apprentissage automatique pour l'OPF manquent d'interprétabilité et peuvent produire des décisions irréalisables, ce qui entrave leur adoption par les parties prenantes de l'industrie. Pour cela, nous proposons une nouvelle approche d'apprentissage interprétable des solutions OPF avec des garanties de faisabilité. Plus précisément, nous développons des arbres de décision prescriptifs qui apprennent la relation entre les données d'entrée et les solutions d'un problème d'optimisation sous contraintes, en utilisant une optimisation robuste pour garantir que les décisions sont réalisables de manière raisonnée. Une contribution importante de notre travail est le développement d'une méthode d'apprentissage basée sur des arbres qui utilise des divisions avancées de l'espace des données d'entrée en utilisant une connaissance experte du domaine, y compris la congestion du réseau et la courbe d'ordre de mérite. En incorporant ces informations, notre approche est capable d'améliorer à la fois l'interprétabilité et les performances du modèle. Nous présentons en outre un algorithme d'apprentissage de substitution pour gérer des problèmes à grande échelle. L'approche proposée est évaluée sur plusieurs réseaux de test, jusqu'à 300 bus, sous différents types d'incertitude et de conditions de fonctionnement, et est comparée à des modèles basés sur des réseaux de neurones, qui ne garantissent pas la faisabilité. Notamment, nos résultats démontrent que les arbres prescriptifs interprétables et peu profonds fonctionnent de manière comparable aux modèles basés sur les réseaux de neurones, qui sont considérés comme l'état actuel de l'art. À notre connaissance, ce travail est le premier à introduire une approche d'apprentissage automatique interprétable pour apprendre directement des solutions OPF avec une faisabilité garantie.

The work presented in this chapter appears in [**J3**] which is under review.

3.1 Introduction

The integration of renewable energy sources in the generation mix necessitates operating modern power systems at a higher speed and scale, to deal with the increased variability and uncertainty. In many cases, traditional workflows may struggle to cope with these requirements, and advanced data-driven methods, such as machine learning, hold significant potential to streamline decision-making processes.

The OPF problem plays a crucial role in power system operation and planning and in electricity markets. It belongs to the class of network flow problems and its objective is to minimize the overall cost of power generation subject to power flow equations and operational constraints, e.g., transmission line limits. In its original form, the OPF problem is a non-convex problem that is difficult to solve. In various important use cases, a linearized version of the OPF that considers only active power, referred to as DC-OPF [74], is utilized. The DC-OPF is especially popular in market clearing, contingency analysis, and techno-economic studies. In particular, the DC-OPF is the cornerstone of deregulated electricity markets as it is widely adopted to determine locational marginal prices which are influenced by network congestion. Further, the DC-OPF is also important for ensuring a reliable operation by considering variants that incorporate steady-state security constraints, such as the Security Constrained DC-OPF. The DC-OPF problem is especially appealing as it can be expressed as an LP problem that can be solved efficiently.

Although general-purpose optimization solvers have made solving LP problems efficient, certain settings can present computational challenges. Specifically, the increasing integration of renewable energy sources introduces significant uncertainty and variability in both power supply and demand, resulting in the need to solve DC-OPF problems repeatedly and at a higher speed and scale. To cope with the uncertainty of renewable production, future electricity markets are expected to move closer to real-time, e.g., operating on a 5-minute ahead basis [75]. Further, it is often assumed that the generation adjusts to real-time variability with an affine control policy, which resembles the widely used Automatic Generation Control (AGC). However, an affine control policy may be restrictive and suboptimal, which motivates resolving the DC-OPF problem in even more granular time scales. In this setting, traditional LP solvers, which have a worst-case complexity that scales polynomially with the size of the grid, may create a computational bottleneck.

Machine learning has been rapidly evolving in recent years, revolutionizing many industries, including power systems [76]. Due to their fast inference times, machine learning models have been proposed as an alternative to traditional optimization solvers for power system problems, such as the DC-OPF. However, power systems are critical infrastructure and stakeholders are naturally risk averse, which presents obstacles to the adoption of these tools [17]. Transparency, interpretability, and performance guarantees are necessary for the practical implementation of machine learning-based solutions for problems such as the DC-OPF. For instance, European Union legislation establishes the need for the so-called

"right to explanation" [77], i.e., the requirement of automated systems to provide information about their internal logic, which necessitates interpretable and transparent methods. Furthermore, interpretability should not compromise model performance but rather should be used to guide domain-agnostic methods with domain knowledge.

3.1.1 Related Work

Leveraging machine learning to accelerate the solution of the DC-OPF problem has attracted significant attention in recent years. This work can be divided into two main research directions. The first focuses on end-to-end learning methods that directly predict the DC-OPF decisions, effectively emulating the LP solver. The second direction explores methods to find a reduced, and therefore easier to solve, DC-OPF problem.

The majority of research on end-to-end learning for DC-OPF focuses on utilizing Neural Network (NN) models to map varying load profiles to problem decisions [78–82]. For instance, [78] proposes an NN model with a constraint violation penalty to predict the DC-OPF solutions; a similar model is developed in [79] for Security Constrained DC-OPF. To ensure the feasibility of decisions, both models require a post hoc projection step. This projection onto the feasible set is itself an optimization problem that needs to be solved, which might be of the same complexity as the original problem, and may potentially negate any computational benefits. In [80], worst-case constraint violations and suboptimality gap are estimated to verify the NN performance; a heuristic method to improve these worst-case guarantees by reducing the input domain is also proposed. In [81], physics-informed NNs demonstrate improved guarantees over standard NNs. However, ensuring that predicted decisions satisfy the problem constraints remains a challenge for end-to-end learning methods as prediction errors are inevitable. To address this issue, [82] develops a preventive learning framework to systematically calibrate inequality constraints to ensure feasibility; however, it relies on estimating the worst-case NN prediction error, which could be challenging. Overall, NN-based models have the modeling capacity to approximate the optimal function that maps load profiles to problem decisions. Nevertheless, even with feasible solutions guaranteed, NN models still lack the interpretability of other ML methods, such as decision trees [83], which is critical for adoption in real-world applications, especially in critical infrastructure. Decision trees are inherently interpretable and have been used in power systems for decades — see, e.g., [84] for an early and [85] for a recent contribution to decision trees for dynamic security assessment.

Predicting the set of problem constraints that are binding at the optimal solution (*active set*) is generally simpler than directly predicting the optimal solutions. Motivated by this observation, the second research direction of leveraging machine learning for DC-OPF focuses on identifying the most probable active sets of constraints to find a reduced version of the original problem [86–90]. Specifically, [86] and [87] utilize statistical learning to identify the most probable critical regions, i.e., parameter regions where the active set of

constraints remains unchanged, which then inform an ensemble policy. In [88], the problem of finding the active sets of constraints is formulated as a multiclass classification task. A neural decoding strategy is developed in [89] to first learn the active set of constraints, mapping uncertain load to the problem objective value, and then find solutions that satisfy the constraints. In the same line of work, [90] proposes a two-step process that combines the prediction of active sets of constraints with an iterative method to recover feasible solutions. While approaches based on learning the active sets of constraints are typically more interpretable and, in many cases, guarantee feasibility, they lack the inference speed of end-to-end learning. Nevertheless, this line of research offers a key insight: while the total number of active constraint sets is exponentially large, only a small number of them are relevant in practice. For instance, [87] finds that the number of critical regions observed for various networks is less than 10 and that this number is not correlated with the network size but rather depends on the load distribution and other network characteristics.

In this chapter, we aim to reconcile these two research directions by proposing an end-to-end learning approach that combines the strengths of both methods and addresses their limitations. Drawing inspiration from recent progress in explainable prescriptive analytics [91], we leverage the insight that only a small number of active constraint sets are practically relevant to enhance both the performance and interpretability of our method. As such, rather than seeking a reduced DC-OPF problem, we develop an end-to-end learning method that is simpler in complexity.

It is worth noting that multiparametric programming [92] is another research area relevant to leveraging machine learning for DC-OPF. Multiparametric programming aims to solve constrained optimization problems as a function of uncertain parameters by identifying critical regions and explicitly constructing a parameter-dependent solution for the whole parameter space. The key difference from our work is that we do not aim to explore the whole parameter space but rather derive an interpretable policy that encodes a few key rules selected in a data-driven manner, starting from available data, and ensuring feasibility for the whole (unobserved) parameter space.

3.1.2 Aim and Contribution

In this chapter, we present a novel method for affine prescriptive trees, i.e., decision trees that learn a piecewise affine mapping from varying input data to the solutions of a constrained optimization problem, namely the DC-OPF problem. We develop a new learning algorithm that combines axis-parallel and *domain-informed*, non-orthogonal splits that encode network information, namely the merit order curve and network congestion. We formulate the expectation of network congestion, conditioned on load, as a classification task and model it with Support Vector Machine (SVM) classifiers. The separating hyperplanes derived from the SVM models are then used as input in the tree learning algorithm, simultaneously improving model performance and interpretability. We also use robust optimization to ensure

the feasibility of the predicted decisions for the whole parameter space in a principled manner. A surrogate learning algorithm is also developed to address the case of potentially prohibitive training time for large-scale problem instances. We provide comprehensive numerical experiments for several test cases ranging from 5 to 300 bus systems, under different assumptions for the distribution of uncertainty and operating conditions. The results show that our method achieves similar performance with state-of-the-art end-to-end learning approaches, namely neural network-based models, while also maintaining interpretability and ensuring the feasibility of decisions.

In summary, our main contribution is twofold. Firstly, we propose an interpretable end-to-end learning method for DC-OPF that offers fast solutions during inference, is computationally tractable, and provides feasibility guarantees. Secondly, we propose a two-step process to learn decision trees with non-orthogonal splits that encode domain-specific information, thereby improving performance and retaining interpretability. To the best of our knowledge, our work is the first to develop interpretable end-to-end machine learning for the DC-OPF problem with feasibility guarantees.

3.1.3 Chapter Outline

The remainder of this chapter is organized as follows. Section 3.2 formulates the problem of learning DC-OPF solutions. Section 3.3 develops the tree-based methodology. Section 3.4 illustrates the proposed methodology in a small test case. Section 3.5 presents our numerical experiments. Section 3.6 concludes and provides directions for future work.

3.2 DC-OPF and Learning Problem Formulation

This section introduces the DC-OPF problem (in Subsection 3.2.1), describes the proposed learning problem (in Subsection 3.2.2), and illustrates how to reformulate it into a tractable problem (in Subsection 3.2.3).

3.2.1 The DC-OPF Problem

This section formulates the DC-OPF problem. We consider a transmission network where $\mathcal{V}$ is the set of buses, $\mathcal{E}$ is the set of lines, and $\mathcal{G}$ is the set of generators. The deterministic DC-OPF problem writes

$$\min_{\mathbf{p}} \quad \mathbf{c}^\intercal \mathbf{p}, \tag{3.1a}$$

$$\text{s.t.} \quad \mathbf{1}^\intercal \mathbf{p} - \mathbf{1}^\intercal \mathbf{d} = 0, \tag{3.1b}$$

$$-\overline{\mathbf{f}} \leq \mathbf{M}(\mathbf{A}\mathbf{p} - \mathbf{d}) \leq \overline{\mathbf{f}}, \tag{3.1c}$$

$$\mathbf{0} \leq \mathbf{p} \leq \overline{\mathbf{p}}, \tag{3.1d}$$

where $\mathbf{p} \in \mathbb{R}^{|\mathcal{G}|}$ denotes the active power of dispatchable generators, $\mathbf{d} \in \mathbb{R}^{|\mathcal{V}|}$ is the stochastic net demand (load demand minus renewable generation) at each bus, $\mathbf{M} \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{V}|}$ is the

Power Transfer Distribution Factors (PTDF) matrix, $\mathbf{A} \in \mathbb{R}^{|\mathcal{G}|\times|\mathcal{V}|}$ is an incidence matrix mapping generators to buses, and $\mathbf{1}(\mathbf{0})$ is a vector of ones (zeros) with appropriate size. Further, $\mathbf{c}, \overline{\mathbf{p}}$, and $\overline{\mathbf{f}}$ are known positive parameters that define the generation cost, the generator capacity, and the line capacity, respectively. The problem objective (3.1a) minimizes the total generation cost, (3.1b) ensures balance of demand and supply, while (3.1c) and (3.1d) denote the generation and transmission line limits, respectively. Without loss of generality, we assume a linear cost function in the objective; quadratic cost functions can always be approximated by a piecewise linear function. Note that the DC-OPF problem (3.1) can be straightforwardly reformulated as a problem of adjusting generation output to the realization of forecast errors in real-time operations by subtracting realized forecast errors from the expected net load.

Also, note that relaxing (3.1b) into a $\geq$ inequality maintains an equivalent solution at optimality. To see this, note that the dual variable of (3.1b) is equal to the negative marginal cost of energy at the slack bus [93]. As the cost vector $\mathbf{c}$ is non-negative, the dual variable of (3.1b) is upper bounded by zero. By performing a sensitivity analysis, we see that adding a positive parameter at the righthand side of (3.1b) would lead to an increase in total generation cost. Therefore, if (3.1b) is relaxed into a $\geq$ inequality, it will be tight at the optimal solution.

Next, we present some assumptions that apply in this work regarding the DC-OPF problem (3.1).

Assumption 3.1 (Bounded uncertainty) *The net load* $\mathbf{d}$ *is restricted in the polyhedron*

$$\mathcal{U} = \{\mathbf{d} \in \mathbb{R}^{|\mathcal{V}|} \,|\, \mathbf{H}\mathbf{d} \leq \mathbf{h}\}. \tag{3.2}$$

This is a standard assumption. In practice, the net load at each bus may vary within a pre-specified range. Formally, this is defined as

$$\mathcal{A} = \{\mathbf{d} \in \mathbb{R}^{|\mathcal{V}|} \,|\, \underline{\mathbf{d}} \leq \mathbf{d} \leq \overline{\mathbf{d}}\}, \tag{3.3}$$

where $\underline{\mathbf{d}}\,(\overline{\mathbf{d}})$ denotes the minimum (maximum) values, with renewable production being defined with negative values. Observe that (3.3) is a special case of (3.2), where $\mathbf{H} = [\mathbf{I}, -\mathbf{I}]^\intercal$ and $\mathbf{h} = [\overline{\mathbf{d}}, \underline{\mathbf{d}}]^\intercal$, where $\mathbf{I}$ denotes an identity matrix of appropriate size.

Assumption 3.2 (Feasibility) *Problem* (3.1) *is feasible* $\forall \mathbf{d} \in \mathcal{U}$.

Note that if the deterministic formulation of the DC-OPF problem is infeasible, then slack variables need to be included in (3.1). For simplicity, we assume that (3.1) is always feasible; however, our proposed method can be straightforwardly extended to address the case when slack variables are required.

Assumption 3.3 (Uniqueness) *Problem* (3.1) *admits a unique solution* $\forall \mathbf{d} \in \mathcal{U}$.

This is also a standard assumption, which holds almost surely for appropriate cost vectors [94].

3.2.2 Data-driven Piecewise Affine Policy

This section presents the proposed data-driven piecewise affine policy for end-to-end learning of the DC-OPF problem.

Instead of solving (3.1), our goal is to learn a function (policy) that maps realizations of net load injections $\mathbf{d}$ to generator setpoints $\mathbf{p}$. From the theory of multiparametric programming [92], we know that the optimal dispatch $\mathbf{p}^*$ with respect to $\mathbf{d}$ takes the form of a piecewise affine function defined over a polyhedral partition of the feasible space. First, we define a polyhedral partition of the feasible space $\mathcal{U}$.

Definition 3.1 (Polyhedral partition [95]) *A collection of L polyhedra $\{\mathcal{U}_\ell\}_{\ell=1}^L$ is a polyhedral partition of a set $\mathcal{U}$ if $\mathcal{U} = \cup_{\ell=1}^L \mathcal{U}_\ell$ and $(\mathcal{U}_i \setminus \partial\mathcal{U}_i) \cap (\mathcal{U}_j \setminus \partial\mathcal{U}_j) = \emptyset$, $\forall i \neq j$, where $\partial\mathcal{U}_i$ denotes the boundary of $\mathcal{U}_i$ and $\setminus$ denotes the set difference operator. In other words, the union of the individual polyhedra $\mathcal{U}_\ell$ covers the feasible space of the net load, and the interiors of the polyhedra do not overlap.*

If the polyhedral partition $\{\mathcal{U}_\ell\}_{\ell=1}^L$ recovers the critical regions of the parameter space, i.e., the regions where the active set of constraints at the optimal solution remains constant, then learning a piecewise affine function over $\{\mathcal{U}_\ell\}_{\ell=1}^L$ is optimal. An explicit solution for finding the optimal piecewise policy can be derived by recasting the problem as a multi-parametric LP problem, but it might be intractable as the number of critical regions grows exponentially with the number of problem constraints in the worst case. In practice, however, only a small number of critical regions are relevant — see, e.g., [87].

Since it is established that a piecewise affine policy is optimal, in this work, we propose learning a simpler, *data-driven* piecewise affine policy, which retains good performance and interpretability. We assume that a data set $\mathcal{D} = \{(\mathbf{d}_i, \mathbf{p}_i^*)\}_{i=1}^N$ of N training observations is available, where $\mathbf{d}_i$ denotes the net load and $\mathbf{p}_i^*$ denotes the vector of optimal decisions derived from solving (3.1) for the i-th sample. In a data-driven setting, a polyhedral partition $\{\mathcal{U}_\ell\}_{\ell=1}^L$ also implies a respective partition of training data $\{\mathcal{D}_\ell\}_{\ell=1}^L$, i.e., subsets of data that fall in each polyhedron. Formally, we define

$$\mathcal{D}_\ell = \{(\mathbf{d}_i, \mathbf{p}_i^*), i \in [N] \,|\, \mathbf{d}_i \in \mathcal{U}_\ell\} \subseteq \mathcal{D}, \tag{3.4}$$

where $[N]$ is shorthand for $\{1, \ldots, N\}$.

In the following, we present the proposed data-driven piecewise affine policy that maps net load observations to decisions. First, we particularize Definition 3.1 to the current data-driven setting.

Definition 3.2 (N_{min}-admissible polyhedral partition) *Consider a scalar $N_{min} > 0$, a polyhedral partition $\{\mathcal{U}_\ell\}_{\ell=1}^L$, and a corresponding data partition $\{\mathcal{D}_\ell\}_{\ell=1}^L$. We say that $\{\mathcal{U}_\ell\}_{\ell=1}^L$ is N_{min}-admissible, if $|\mathcal{D}_\ell| \geq N_{min}$, $\forall \ell \in [L]$.*

Therefore, Definition 3.2 only considers polyhedral partitions where each polyhedron includes a minimum number of data observations; here is also where our approach differentiates from multiparametric programming [92]. As shown in previous works [87], the number of critical regions populated with data observations is small in practice. The tree-learning algorithm developed in the next section effectively learns a partition of the form of Definition 3.2 that is as close as possible to the critical regions of the parameter space with data observations. In that case, N_{min}, which is a user-defined hyperparameter, corresponds to the minimum number of observations per each tree leaf and controls the complexity of the learned policy.

The proposed data-driven piecewise affine policy is defined as follows.

Definition 3.3 (Data-driven piecewise affine policy) *We consider a data-driven piecewise affine policy $f : \mathcal{U} \to \mathbb{R}^{|\mathcal{G}|}$ that maps net load $\mathbf{d}$ to generator setpoints $\mathbf{p}$, given by $f(\mathbf{d}) = \mathbf{W}_\ell \mathbf{d} + \mathbf{b}_\ell$, $\mathbf{d} \in \mathcal{U}_\ell$, $\ell = 1, \ldots, L$, where $\mathbf{W}_\ell \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{V}|}$ is a matrix of linear decision rules, $\mathbf{b}_\ell$ is the intercept vector, and $\{\mathcal{U}_\ell\}_{\ell=1}^{L}$ is an N_{min}-admissible polyhedral partition of $\mathcal{U}$, defined over a data set $\mathcal{D}$.*

Given an N_{min}-admissible polyhedral partition $\{\mathcal{U}_\ell\}_{\ell=1}^{L}$, the problem of finding the optimal decision rules, for each $\ell \in [L]$, is given by

$$\min_{\mathbf{W}_\ell, \mathbf{b}_\ell} \quad \frac{1}{|\mathcal{D}_\ell|} \sum_{i \in \mathcal{D}_\ell} \mathbf{c}^\intercal (\mathbf{W}_\ell \mathbf{d}_i + \mathbf{b}_\ell), \tag{3.5a}$$

$$\text{s.t.} \quad \mathbf{1}^\intercal (\mathbf{W}_\ell \mathbf{d} + \mathbf{b}_\ell) - \mathbf{1}^\intercal \mathbf{d} \geq 0, \qquad \forall \mathbf{d} \in \mathcal{U}_\ell, \tag{3.5b}$$

$$-\overline{\mathbf{f}} \leq \mathbf{M}(\mathbf{A}(\mathbf{W}_\ell \mathbf{d} + \mathbf{b}_\ell) - \mathbf{d}) \leq \overline{\mathbf{f}}, \qquad \forall \mathbf{d} \in \mathcal{U}_\ell, \tag{3.5c}$$

$$\mathbf{0} \leq \mathbf{W}_\ell \mathbf{d} + \mathbf{b}_\ell \leq \overline{\mathbf{p}}, \qquad \forall \mathbf{d} \in \mathcal{U}_\ell, \tag{3.5d}$$

where the decision vector $\mathbf{p}$ has been replaced by the affine policy $\mathbf{W}_\ell \mathbf{d} + \mathbf{b}_\ell$. Problem (3.5) finds the affine decision rules that minimize the in-sample dispatch cost (3.5a) for the given partition. Effectively, by solving problem (3.5) for each $\mathcal{U}_\ell$ we learn the parameters of the proposed data-driven policy, which is of the form of Definition 3.3. Note that each row of $\mathbf{W}_\ell$ defines a vector of coefficients that maps net load to a specific generator. The robust constraints (3.5b)-(3.5d) further ensure a feasible policy, i.e., decisions are feasible for all realizations of the uncertainty within $\mathcal{U}_\ell$. At test time, for an out-of-sample observation $\mathbf{d}_0$, we first locate the respective partition $\mathcal{U}_\ell$ it falls into, and then derive the generator production from $f(\mathbf{d}_0) = \mathbf{W}_\ell \mathbf{d}_0 + \mathbf{b}_\ell$.

Note that in the robust formulation, we replaced the equality constraint (3.1b) with an inequality constraint (3.5b); thus, problem (3.5) ensures that the total net load is always covered by the aggregated production. The reason for this is twofold. First, equality constraints with uncertain parameters drastically reduce the feasible set, leading to over-conservative solutions or even infeasibility [67, Ch. 12]. Working with an inequality allows us to "free" the parameters of the affine decision rules and attain higher performance. In the end, as (3.5a) minimizes the total production cost, the forecast decisions obtained from

the affine policy will try to be as close as possible to $\mathbf{1}^\intercal\mathbf{d}$. Second, in reality, generation must always be larger than demand due to line losses. Moreover, there are always small deviations between aggregated production and net load, which results in frequency variations; as it is easier to provide downward frequency regulation via, e.g., curtailment of renewable production, we ensure that the policy never underestimates the total demand.

Formally, given an N_{min}-admissible polyhedral partition $\{\mathcal{U}_\ell\}_{\ell=1}^{L}$ that covers the whole feasible space $\mathcal{U}$ and robust constraints (3.5b) – (3.5d) that ensure forecast decisions are feasible $\forall\mathbf{d} \in \mathcal{U}_\ell$, it follows that the forecast decisions will always satisfy the constraints of (3.1), where (3.1b) has been relaxed into an inequality constraint. Thus, we obtain guarantees about the feasibility of decisions for any realization of $\mathbf{d} \in \mathcal{U}$.

Remark 3.1 *If Assumption 3.2 does not hold, then* (3.1) *requires additional slack variables. In this case, we introduce additional rules in* (3.5) *that map realizations of* $\mathbf{d}$ *to each slack variable.*

The objective (3.5a) minimizes the prescriptive cost, i.e., the expected in-sample dispatch cost. Alternatively, the MSE between the optimal and forecast decisions can be minimized, given by

$$\frac{1}{|\mathcal{D}_\ell|}\sum_{i\in\mathcal{D}_\ell}\|\mathbf{W}_\ell\mathbf{d}_i + \mathbf{b}_\ell - \mathbf{p}_i^*)\|_2^2, \tag{3.6}$$

as is the case in many relevant works — see, e.g., [80]. The MSE measures the predictive error of forecast decisions. However, here we focus primarily on the prescriptive cost, as the ultimate goal is to minimize the total dispatch cost.

3.2.3 Robust Constraint Reformulation

Problem (3.5) involves semi-infinite robust constraints. As we deal with an LP problem and polyhedral uncertainty sets, we apply techniques from robust optimization [67] to reformulate (3.5) into a deterministic LP problem.

For illustration purposes, consider the upper generation limit at the left-hand side of (3.5d). Considering that the inequality holds $\forall\mathbf{d} \in \mathcal{U}_\ell$, i.e., the worst-case of $\mathbf{d}$, we write equivalently

$$\max_{\mathbf{d}}\{\mathbf{W}_\ell\mathbf{d} \,|\, \mathbf{H}_\ell\mathbf{d} \le \mathbf{h}_\ell\} \le \overline{\mathbf{p}} - \mathbf{b}_\ell.$$

As the max problem is linear in $\mathbf{d}$, it can be replaced by its dual

$$\min_{\boldsymbol{\lambda}}\{\mathbf{h}_\ell^\intercal\boldsymbol{\lambda} \,|\, \mathbf{H}_\ell^\intercal\boldsymbol{\lambda} = \mathbf{W}_\ell, \boldsymbol{\lambda} \ge \mathbf{0}\} \le \overline{\mathbf{p}} - \mathbf{b}_\ell,$$

where $\boldsymbol{\lambda}$ is a dual variable of appropriate size. Evidently, the min operator becomes redundant. Hence, the upper generation limit constraint in the left-hand side of (3.5d) is replaced by the following constraints

$$\mathbf{h}_\ell^\intercal\boldsymbol{\lambda} \le \overline{\mathbf{p}} - \mathbf{b}_\ell,\ \mathbf{H}_\ell^\intercal\boldsymbol{\lambda} = \mathbf{W}_\ell,\ \boldsymbol{\lambda} \ge \mathbf{0}.$$

The rest of the constraints are reformulated in a similar fashion, leading to a deterministic LP problem that can be solved with off-the-shelf solvers.

3.3 Tree-based Learning Methodology

This section develops the proposed tree-based method to learn an interpretable policy for the DC-OPF problem. First, we describe the tree-learning algorithm (Section 3.3.1). Next, we detail the process of finding domain-informed splits (Section 3.3.2). Finally, we describe a surrogate learning method to deal with large problem instances (Section 3.3.3). The two-step process to train the proposed tree-based model is illustrated in Fig. 3.1.

3.3.1 Affine Prescriptive Trees

In this section, we present our decision tree algorithm for learning a piecewise affine policy.

Decision trees use available data to partition the feature space into L leaves by minimizing a predefined loss criterion, e.g., minimizing the variance of each leaf. The resulting partition also provides information about the joint distribution of the target variable and associated features and, therefore, can be used to predict instances of the target variable given out-of-sample feature observations. Here, our primary goal is to use a tree-based algorithm to learn a polyhedral partition of the form of Definition 3.2 that is as close as possible to the critical regions of the parameter space, using data set $\mathcal{D}$.

Our proposed algorithm combines *axis-parallel* and *non-orthogonal* splits during the tree-learning process. To clarify, axis-parallel splits refer to splits that only consider a single feature, while non-orthogonal splits refer to splits that consider a linear combination of different features. Mathematically, both axis-parallel and non-orthogonal splits are represented as a set of hyperplanes. Using this combination of splits is a departure from most state-of-the-art tree algorithms that focus solely on binary trees with axis-parallel splits—see, e.g., [55] for single trees and [56] for tree-based ensembles. Oblique decision trees [96] allow for non-orthogonal splits and have been shown to lead to significant performance improvements; however, they can be computationally challenging and less interpretable [83]. To address this challenge, we construct a set of domain-informed non-orthogonal splits prior to the learning phase; the process of identifying these splits is detailed in Section 3.3.2.

Algorithm 3.1 describes our decision tree algorithm in detail. Consider a root node (equivalently, partition) $\mathcal{U}_0 = \{\mathbf{d} \,|\, \mathbf{H}_0\mathbf{d} \leq \mathbf{h}_0\}$, a corresponding data set $\mathcal{D}_0$, and a set of K candidate hyperplanes to split on $\{(\boldsymbol{\alpha}_k, \beta_k)\}$, parameterized by vectors $\boldsymbol{\alpha}_k$ and scalars β_k. These hyperplanes model both non-orthogonal and axis-parallel splits as a special case, e.g., if we want to split in value s of feature d_1, then $\boldsymbol{\alpha}_k = [1, \mathbf{0}]^\intercal$, and $\beta_k = s$.

A node split partitions a parent node into two child nodes $\mathcal{U}_0 = \mathcal{U}_l \cup \mathcal{U}_r$, such that $\mathcal{U}_l = \{\mathbf{d} \,|\, \boldsymbol{\alpha}_k^\intercal \mathbf{d} \leq \beta_k,\, \mathbf{d} \in \mathcal{U}_0\}$ and $\mathcal{U}_r = \{\mathbf{d} \,|\, \boldsymbol{\alpha}_k^\intercal \mathbf{d} > \beta_k,\, \mathbf{d} \in \mathcal{U}_0\}$. The training algorithm starts at root node $\mathcal{U}_0$ and sets the current depth $\delta = 0$. Next, it iterates over the K candidate splits and solves (3.5) for each child partition; note that, to deal with the strict inequality induced by the node split, the right child node is evaluated at its closure $\text{cl}(\mathcal{U}_r)$. Embedding (3.5) within the tree-learning algorithm ensures that node splits are selected

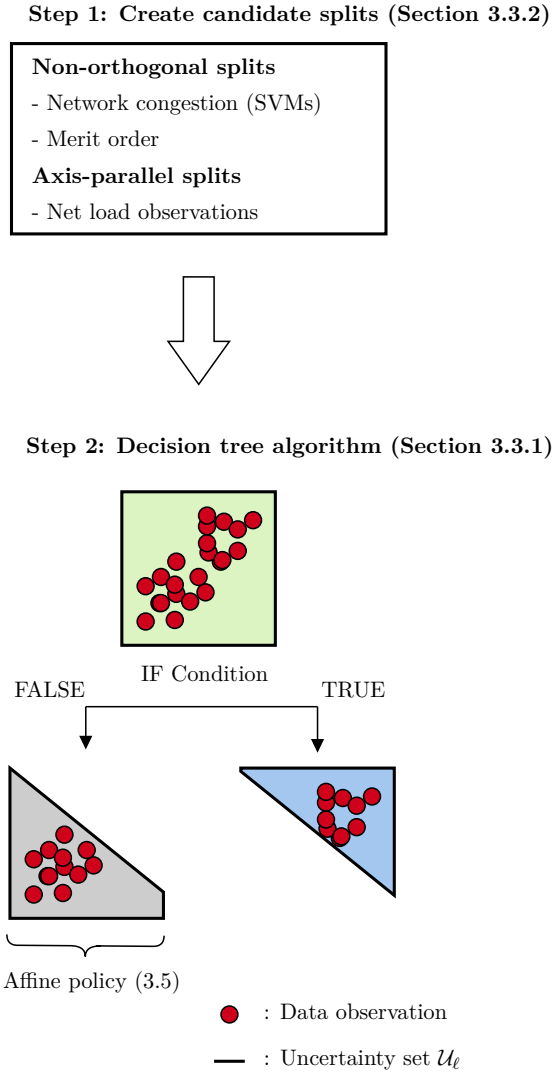


Figure 3.1: Flowchart of the proposed two-step training process. Step 1 creates a set of candidate node splits. Step 2 uses the candidate node splits as input to grow a decision tree. The tree graph at the bottom visualizes a non-orthogonal node split.

based on their impact on the true decision cost of the DC-OPF problem. Specifically, the split that minimizes the prescriptive cost of the piecewise affine policy is selected and the corresponding polyhedral partition is added to the tree, updating the tree structure accordingly. For reference, Fig. (3.1) visualizes splitting a tree node using a non-orthogonal split. At each iteration, the current tree leaves define an N_{min}-admissible polyhedral partition $\{\mathcal{U}_\ell\}_{\ell=1}^{L}$ and an equivalent data partition $\{\mathcal{D}_\ell\}_{\ell=1}^{L}$. The process is repeated recursively in a top-down fashion until a stopping criterion is met. Typical stopping criteria include a minimum number of observations per leaf N_{min} and the maximum tree depth δ^{max}.

The proposed tree-learning algorithm grows trees that minimize decision costs and map data to prescriptions. We take an intermediate approach to split selection, avoiding the well-known shortcoming of CART-like methods [55], which is determining each split without considering the possible impact on future splits[1]. Specifically, we apply a semi-greedy split selection, which prioritizes non-orthogonal splits over axis-parallel ones, as the former encode domain knowledge. To implement the semi-greedy split selection, we use an auxiliary function called `ispar`, which takes a vector $\boldsymbol{\alpha}_k$ as input and returns a logical value of `True` if $\boldsymbol{\alpha}_k$ is parallel and `False` otherwise. In the tree-learning algorithm, if the current best split is non-orthogonal, we only evaluate the remaining non-orthogonal splits. If the current best split is axis-parallel, then we evaluate all the remaining splits, including the rest of the axis-parallel ones. This is described in Steps 4-5 of Algorithm 3.1, where $\neg, \wedge$ denote the logical negation and conjunction (*and*) operators, and the **continue** statement interrupts the current step of a loop and continues with the next iteration. This approach prioritizes domain-informed non-orthogonal splits while still allowing for data-driven axis-parallel splits to be considered if the former are insufficient.

The hyperparameters of the decision tree include the minimum number of observations N_{min} per leaf and the maximum tree depth δ^{max}, both controlling the complexity of the learned policy. Namely, N_{min} controls the bias-variance trade-off, with smaller values increasing the risk of overfitting, and ensures that the final polyhedral partition is admissible as per Definition 3.2. Conversely, larger values of δ^{max} lead to improved performance, but may also result in overfitting and reduced interpretability. The maximum number of partitions that can be recovered is $2^{\delta^{max}}$ and is independent of the size of the underlying network. For a sufficiently complex policy, i.e., one with small N_{min} and large δ^{max}, we expect that the number of partitions recovered scales with the number of critical regions that are populated with data observations. Thus, we avoid the shortcoming of multiparametric LP, where the number of partitions scales exponentially with the problem constraints. To promote interpretability and avoid potential overfitting, we suggest using larger values of N_{min} and smaller values of δ^{max}.

[1]Note that globally optimal trees [83] address this shortcoming using a mixed-integer LP formulation, at the expense, however, of increased computational cost.

Algorithm 3.1 `AffinePrescrTree`

Input: current partition $\mathcal{U}_0$, current data set $\mathcal{D}_0$, current depth δ, hyperparameters $\{N_{min}, \delta^{max}\}$, set of candidate splits $\{(\boldsymbol{\alpha}_k, \beta_k)\}_{k=1}^K$, auxiliary function `ispar`

Output: tree τ

find $v_0 = \min_{\mathbf{d} \in \mathcal{U}_0} (3.5)$, set $v_{min} \leftarrow |\mathcal{D}_0| \cdot v_0$, `split`$\leftarrow$`False`, $k^* \leftarrow$ empty
if $\delta < \delta^{max}$ and $N_0 \geq 2N_{min}$ **then**
  for $k = 1, \ldots, K$ **do**
    if $\neg$`ispar`$(\boldsymbol{\alpha}_{k^*}) \wedge$ `ispar`$(\boldsymbol{\alpha}_k) ==$ `True` **then**
      continue
    else
      find left and right child nodes $\mathcal{U}_l, \mathcal{U}_r$, and corresponding data partitions $\mathcal{D}_l, \mathcal{D}_r$
      if $|\mathcal{D}_l| \geq N_{min}$ and $|\mathcal{D}_r| \geq N_{min}$ **then**
        $v_k = |\mathcal{D}_l| \cdot \min_{\mathbf{d} \in \mathcal{U}_l} (3.5) + |\mathcal{D}_r| \cdot \min_{\mathbf{d} \in \mathrm{cl}(\mathcal{U}_r)} (3.5)$
        if $v_k < v_{min}$ **then**
          update $v_{min} \leftarrow v_k$, `split` $\leftarrow$ `True`, $k^* \leftarrow k$
        end if
      end if
    end if
  end for
  if `split == True` **then**
    append $(\boldsymbol{\alpha}_{k^*}, \beta_{k^*})$ to $\mathbf{H}_0, \mathbf{h}_0$ for each new partition $\mathcal{U}_l, \mathcal{U}_r$, find $\mathcal{D}_l, \mathcal{D}_r$
    $\tau_l =$ `AffinePrescrTree`$(\mathcal{U}_l, \mathcal{D}_l, \delta + 1)$
    $\tau_r =$ `AffinePrescrTree`$(\mathcal{U}_r, \mathcal{D}_r, \delta + 1)$
    update tree structure τ
  end if
end if
return τ

3.3.2 Domain-Informed, Non-Orthogonal Splits

This section describes how to identify the set of K candidate splits.

Axis-parallel splits only check whether an entry of $\mathbf{d}$ exceeds a threshold value; they are purely data-driven and the standard approach to growing binary trees, e.g., CART. In this work, the set of axis-parallel splits comprises a number of equally spaced quantiles of the empirical net load distribution over data set $\mathcal{D}$; i.e., for each net load at each node, we estimate a set of quantiles from its marginal distribution and evaluate the splitting criterion there.

A key contribution of this work is proposing domain-informed, non-orthogonal splits that are potentially more effective than data-driven axis-parallel splits. The proposed splits are derived from hyperplanes that encode information about the active set of constraints conditioned on the load profile, namely the merit order curve and network congestion.

Merit Order Splits

For ease of discussion, further assume the generators in $\mathcal{G}$ are ordered in ascending order based on their cost, i.e., for $i, j \in \mathcal{G}$, if $i < j$, then $c_i < c_j$. Hence, for an optimal solution $\mathbf{p}^*$, assuming no line congestion, we have $p_i = \overline{p}_i$ whenever $p_j > 0$. This means that generator j will be dispatched only if the total net load is larger than the aggregated production of the generators that rank lower in terms of cost.

To encode this information, we construct a set of hyperplanes $\{\mathbf{1}^\intercal \mathbf{d} \geq \sum_{i=1}^{j} \overline{p}_i\}$ for $j \in \mathcal{G}$. That is, each hyperplane corresponds to a supply curve that renders the respective generator as the marginal one, and checks whether the aggregated demand exceeds the total generation capacity.

Network Congestion Splits

Here, we propose non-orthogonal splits that encode information about expected network congestion conditioned on input net load profiles. To this end, we train a set of classifiers, namely SVMs [97] with a linear kernel to predict whether a line gets congested. However, we do not use the SVMs for out-of-sample prediction; instead, we retrieve the maximum margin hyperplane learned for each SVM and use it as a candidate split in the tree learning process.

The process of creating non-orthogonal splits that model network congestion is described as follows. First, we inspect the full training data set $\mathcal{D}$ for line congestions. For each congested line, we formulate a binary classification problem with the line status as the target variable and the net load observations $\mathbf{d}_i$ as features. We then train an SVM model with a linear kernel for each classification task, which effectively learns a separating hyperplane, parameterized by linear coefficients $\mathbf{w}$ and the intercept b. These separating hyperplanes are subsequently used as candidate splits during the decision tree learning process, as shown in Fig. 3.1 and detailed in Algorithm 3.1.

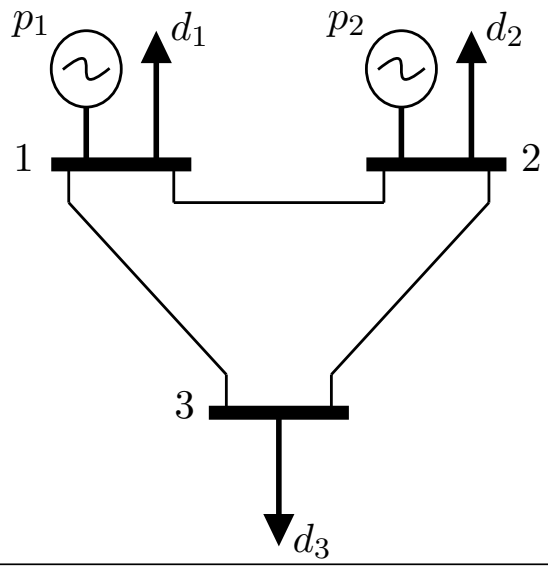
3.3.3 Dealing with Large-scale Problems

Training the proposed affine prescriptive trees requires solving (3.5) repeatedly during training. Specifically, for a tree of depth δ, assuming K candidate splits at root node, problem (3.5) need to be solved up to $\sum_{\delta=0}^{\delta^{max}} 2^{\delta}(K-\delta)$ times during the offline training phase. However, the training process might become computationally prohibitive for larger networks. To mitigate this issue, we explore two directions to reduce the offline computational cost, namely, to speed up the solution of (3.5) and to reduce the time to find the polyhedral partition.

Firstly, we use an iterative algorithm to speed up the solution of (3.5). Section 3.2.3 uses duality theory to reformulate (3.5) into a deterministic optimization problem. Depending on the problem size, however, iterative cutting-plane methods may be faster [98]. Here, we propose an intermediate approach that leverages the fact that only a small number of line constraints are binding at the optimal solution. We initialize our master problem by reformulating (3.5b) and (3.5d) using duality, ignoring all line constraints (3.5c). Next, we solve the master problem and retrieve $\mathbf{W}_{\ell}^{*}, \mathbf{b}_{\ell}^{*}$. We then iterate over all the lines, fix the affine decision rules, and estimate the worst-case constraint violation, which is a maximization problem over $\mathbf{d}$. The line that leads to the highest violation is selected, and the respective row of (3.5c) is reformulated via duality and added to the master problem. The algorithm terminates when there is no violation. The training data set $\mathcal{D}$ can also inform us of which lines might lead to violations; thus, we can warm-start the iterative algorithm by adding these lines to the initial master problem. In this case, the algorithm typically terminates after a small number of iterations.

Secondly, we propose a surrogate tree-learning algorithm that "relaxes" the training process, thus reducing the time to find the polyhedral partition. Instead of training the tree in a fully prescriptive fashion as detailed in Algorithm 3.1, we take a sequential approach. First, we grow a decision tree minimizing the MSE (3.6) criterion with no constraints, for which a closed-form solution exists, and maintain the semi-greedy split selection. After retrieving an N_{min}-admissible polyhedral partition, we iterate over each leaf and estimate the affine decision rules that minimize the within-leaf dispatch cost by solving (3.5). Note that the original algorithm jointly estimates the polyhedral partition and the policy, i.e., the affine decision rules, in a semi-greedy, top-down fashion. The "relaxed" version, on the other hand, takes a sequential approach: first, we find the polyhedral partition, then we learn the affine decision rules. The surrogate learning algorithm could also be utilized with more computationally demanding variants of the DC-OPF problem.

These approaches can significantly reduce the offline computational cost, making the proposed tree-based method computationally tractable for larger networks.



Gen.	Cost ($/MWh)	$\overline{p}$ (MW)	Line	From Bus	To Bus	X	$\overline{f}$ (MW)
p_1	5	1000	1	1	3	0.62	9000
p_2	1.2	270	2	3	2	0.75	90
			3	1	2	0.9	9000

Figure 3.2: Modified 3-bus system.

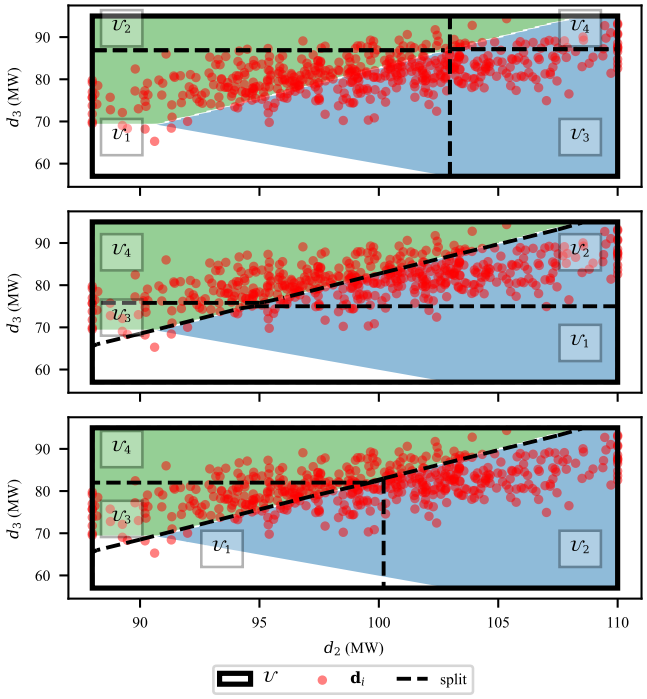


Figure 3.3: Top: Tree with axis-parallel splits. Middle: Tree with non-orthogonal splits. Bottom: Tree with non-orthogonal splits, trained with the surrogate method. Colored subregions indicate critical regions and red points indicate training observations. Solid lines show the load domain, $\mathcal{U}_i$ represents the i-th leaf.

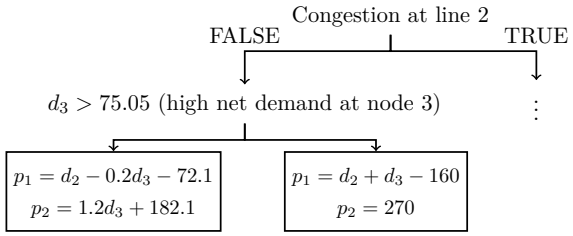


Figure 3.4: Visualization of piecewise affine policy.

3.4 Illustrative Example

We illustrate the most salient features of our approach using the 3-bus system from the PGLib-OPF library [99], which we modify by setting the maximum capacity of the cheapest generator at $\overline{p}_2 = 270\,\text{MW}$, and the capacity of the line connecting buses 2 and 3 at $\overline{f}_2 = 90\,\text{MW}$ —see Fig. 3.2 for details. If neither of these limits is reached, then at the optimal solution $p_1^* = 0$ and $p_2^* = \mathbf{1}^\intercal\mathbf{d}$; else, $p_1^* > 0$. We further assume that $d_1 = 110\,\text{MW}$ and that d_2, d_3 follow a multivariate normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\boldsymbol{\mu} = (99, 81)\,\text{MW},\ \boldsymbol{\Sigma} = \begin{bmatrix} 30.25 & 15.75 \\ 15.75 & 22.65 \end{bmatrix}\text{MW}^2,$$

are the mean vector and covariance matrix, respectively, and lie within intervals $d_2 \in [88, 110]\,\text{MW}$ and $d_3 \in [57, 95]\,\text{MW}$.

We generate 1 000 random observations and apply a 50/50 training/test split to examine the performance of prescriptive trees with respect to hyperplane splits, setting $\delta^{max} = 2$ and $N_{min} = 25$. Performance is evaluated by estimating the mean increase in decision cost over the test set compared to a traditional LP solver. Three models are trained: one using only axis-parallel splits, one using both axis-parallel and non-orthogonal splits, and one using both splits but trained with the surrogate method developed in Section 3.3.3. For axis-parallel splits, we examine 9 equally spaced quantiles estimated from the training observations. For non-orthogonal splits, we consider a merit order split that checks whether $\mathbf{1}^\intercal\mathbf{d} \geq \overline{p}_2$ and a network congestion split derived from an SVM that predicts when line 2 gets congested.

Fig. 3.3 plots the tree splits as a function of d_2, d_3, where the colored subregions indicate the load profiles for which the set of active constraints does not change. Specifically, the green subregion indicates line 2 is congested, the blue subregion indicates that the maximum capacity of the cheapest generator is reached ($p_2^* = \overline{p}_2$), and the white subregion indicates that no upper limit is reached ($p_1^* = 0$, $p_2^* = \mathbf{1}^\intercal\mathbf{d}$). Evidently, the optimal policy is piecewise linear with respect to each subregion, and a tree that recovers this partition would yield an optimal policy.

Considering only axis-parallel splits cannot recover a near-optimal partition and leads to an out-of-sample mean cost increase of 3.19%—see top of Fig. 3.3. Conversely, non-orthogonal splits lead to significantly better decisions with an out-of-sample mean cost

increase of 0.37%, as the root node is split at the hyperplane provided by the SVM — see the middle of Fig. 3.3. A small decision error persists as the critical regions are not recovered exactly by the polyhedral partition; thus, leaves that extend to more than one subregion, i.e., $\mathcal{U}_3$,$\mathcal{U}_1$, lead to slightly suboptimal decisions. Specifically, perfectly separating between instances of line congestion (green subregion) and the rest requires a piecewise affine function. The hyperplane learned from the SVM model cannot provide a perfect separation. Nonetheless, its combination with the subsequent axis-parallel splits leads to a very good approximation of the optimal solution. The surrogate method leads to a mean cost increase of 1.72%, which ranks in between the other models. Compared to the fully prescriptive method, the increased cost of the surrogate algorithm is attributed to the selection of axis-parallel splits. First, the split on $d_2 > 100.36$ creates two partitions that extend over two critical regions — see $\mathcal{U}_1$,$\mathcal{U}_2$ in the bottom of Fig. 3.3. Second, the split that separates $\mathcal{U}_3$,$\mathcal{U}_4$ ($d_3 > 81.78$) leads to a similar number of observations at each leaf. Conversely, the respective split at the middle of Fig. 3.3 ($d_3 > 75.75$) explicitly maximizes the coverage of each critical region, i.e., maximizes the area of $\mathcal{U}_4$. Interestingly, the merit order split is not selected in either case. Even though it perfectly separates the white from the blue subregion, there are too few observations within the white region to merit splitting a node there. If d_2 and d_3 were, in contrast, uniformly distributed within their respective intervals, the merit order split would become highly prescriptive and, thus, selected by Algorithm 3.1.

Fig. 3.4 provides an interpretable visualization of the piecewise affine policy of the prescriptive tree with hyperplanes (middle of Fig. 3.3), focusing on $\mathcal{U}_1$,$\mathcal{U}_2$. Intuitively, the root node examines if congestion in line 2 is expected; if not, then we evaluate d_3. If $d_3 > 75.05$, i.e., we reach $\mathcal{U}_2$, then $p_2 = \overline{p}_2$ and p_1 covers the excess demand (recall that $d_1 = 110$MW). Conversely, when $d_3 <= 75.05$, we reach $\mathcal{U}_1$, which extends to the white and blue subregions; here, both p_1, p_2 linearly depend on the varying demands.

3.5 Numerical Experiments

In this section, we describe our experimental setup (in Subsection 3.5.1), present our main results (in Subsection 3.5.2), and provide additional results under challenging operating conditions (in Subsection 3.5.3). The code to reproduce the results is made available in [100].

3.5.1 Experimental Setup

The proposed methodology is demonstrated on a range of PGLib-OPF networks v21.07 [99] of up to 300 buses. The net load domain is defined as $\mathcal{U} = \{\overline{\mathbf{d}} - 0.4|\overline{\mathbf{d}}| \leq \mathbf{d} \leq \overline{\mathbf{d}}\}$, where $\overline{\mathbf{d}}$ denotes the nominal load values from the base case specified in [99]. Thus, positive loads vary within 60% and 100% of their nominal value. Two settings with respect to uncertainty are considered. First, each net load is independently and uniformly distributed within $\mathcal{U}$.

Second, the net loads follow a multivariate normal distribution. For each net load d_j, the mean value is set at $\mu_j = 0.8\overline{d}_j$ and its standard deviation at $\sigma_j = 0.05\overline{d}_j$. We further sample correlations across net loads uniformly from $[0, 1]$ and use it to create the covariance matrix. In both cases, we generate 20 000 samples, and apply a 50/50 training/test split.

The following models are examined:

- `APT`: an affine prescriptive tree using only axis-parallel splits.
- `APTH`: an affine prescriptive tree using both axis-parallel and non-orthogonal splits.
- `APTH-rlx`: an affine prescriptive tree using both axis-parallel and non-orthogonal splits and trained with the surrogate algorithm of Section 3.3.3.
- `NN-prj`: an NN-based end-to-end learning model, coupled with an additional projection step.

For the tree-based models, namely `APT`, `APTH`, `APTH-rlx`, we set $N_{min} = 25$ and $\delta^{max} = 3$, which are values that enable interpretability and avoid overfitting. Axis-parallel splits are evaluated at 19 equally spaced quantiles estimated from the training observations. We further consider a hard time-limit constraint of 10 000 seconds; that is, if the time limit is reached, we stop growing the tree and each node becomes a leaf. For the larger networks, i.e., case118, case300, we use the iterative algorithm described in Section 3.3.3 to solve (3.5). For `NN-prj`, we consider a multi-layer feed-forward structure with 4 hidden layers and 100 nodes per layer, using the MSE loss and the ReLU activation function in the hidden layers. Following [78, 79, 82], we apply a sigmoid activation function in the output layer, thus ensuring that the predicted decisions satisfy the generation capacity constraints. We further add a regularization term in the objective that penalizes excessive line flows, following [79]. The rest of the hyperparameters are also set according to [79] and the NN model is trained with early stopping to avoid overfitting. An ℓ_1-projection step is applied post hoc to ensure feasible decisions. For the ground truth solution of the DC-OPF problem, we use the `Gurobi` solver [101] with default settings. All experiments are run on a standard PC featuring an Intel Core i7 CPU with a clock rate of 2.7 GHz and 16GB of RAM.

For performance evaluation, we measure the suboptimality of predicted decisions by estimating the percentage of Mean Cost Increase (MCI) over a test set of N_{test} observations, given by

$$100\frac{1}{N_{\text{test}}}\sum_{i\in[N_{\text{test}}]}\frac{\mathbf{c}^\intercal(\hat{\mathbf{p}}_i - \mathbf{p}_i^*)}{\mathbf{c}^\intercal\mathbf{p}_i^*},$$

where $\mathbf{p}_i^*$ is the optimal solution derived from `Gurobi` and $\hat{\mathbf{p}}_i$ the predicted solution for the i-th test sample.Evidently, MCI is non-negative.

3.5.2 Results

Table 3.1 summarizes the results of the SVM classifiers, namely the number of lines that face congestion at least once, the number of SVM classifiers trained, and the average and

Table 3.1: Number of congested lines, number of SVM models trained, and classifier accuracy (%).

	Uniform		Normal	
	No. lines/models	mean/min acc. (%)	No. lines/models	mean/min acc. (%)
case5	1 / 1	99.97	1 / 1	99.99
case30	1 / 1	99.79	1 / 1	99.91
case39	2 / 1	99.83	2 / 1	99.60
case57	0 / 0	-	0 / 0	-
case118	5 / 3	93.10 / 80.96	5 / 4	95.72 / 84.60
case300	13 / 8	97.59 / 93.36	17 / 8	96.83 / 89.92

Table 3.2: Percentage (%) of MCI, $\delta^{max} = 3$. Parentheses show the rate of infeasibility (%).

	Uniform				Normal			
	`APT`	`APTH`	`APTH-rlx`	`NN-prj`	`APT`	`APTH`	`APTH-rlx`	`NN-prj`
case5	1.62	**0.40**	0.46	0.96 (5.35)	**0.30**	0.33	1.39	0.86 (1.61)
case30	4.20	0.76	0.85	**0.52 (5.12)**	1.88	1.19	1.58	**0.60 (14.18)**
case39	2.07	0.22	0.23	**0.21 (3.37)**	1.54	**0.16**	0.48	0.35 (1.17)
case57	**0.00**	**0.00**	**0.00**	0.18 (0.11)	**0.00**	**0.00**	**0.00**	0.18 (0.27)
case118	1.17	0.42	0.28	**0.19 (7.73)**	1.17	1.14	0.37	**0.16 (21.85)**
case300	3.10	2.81	2.44	**1.80 (43.29)**	3.12	3.12	2.43	**1.20 (59.48)**

minimum classifier accuracy (%) per test case. Note that to train an SVM classifier we require at least N_{min} observations per class label; that is, if a line is almost always or almost never congested, we do not train a model— see, e.g., case39, case118, and case300. Overall, the SVM classifiers, even though they only utilize a linear kernel, provide very good out-of-sample performance. For the small and medium-sized cases, the SVM models provide almost perfect separation with close to 100% accuracy. For the larger cases, i.e., case118 and case300, the average accuracy still exceeds 93% for both uniform and normal distribution. However, there is increased variability in individual models, as indicated by the worst-case performance. This is more pronounced for case118, where the worst-case performance is below 85% for both types of uncertainty distribution.

Table 3.2 presents the out-of-sample MCI for the examined test cases. For `NN-prj`, we also report the percentage of infeasible solutions, i.e., the percentage of solutions that require a post hoc projection step to recover feasibility. Clearly, tree-based solutions are feasible by design and their infeasibility rate is always zero, thus we omit it from Table 3.2. In almost all cases, the lowest MCI is smaller than 1%, which is on par with previous works. The worst performance is observed for case300, which is probably attributed to the large number of lines facing congestion.

Overall, considering non-orthogonal splits significantly improves the prescriptive performance of the tree-based method. Specifically, the average (maximum) improvement of `APTH` compared to `APT` is 53% (89%) for uniform distribution and 20% (90%) for normal distribu-

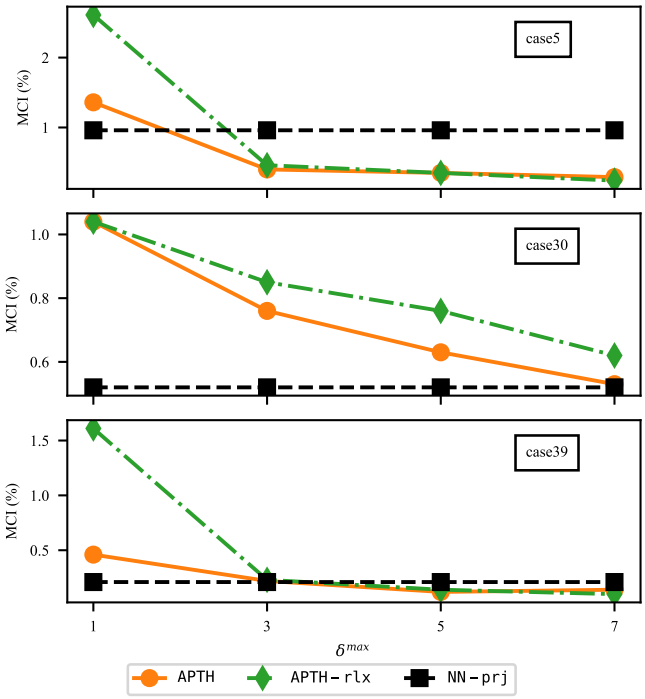


Figure 3.5: MCI versus maximum tree depth δ^{max} (uniform uncertainty).

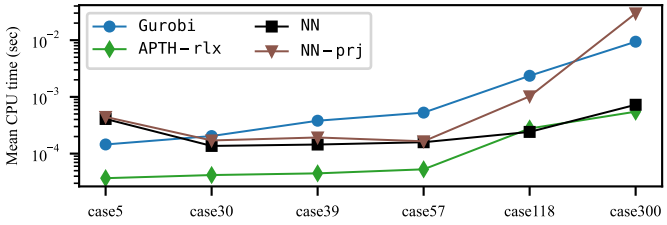


Figure 3.6: Mean CPU time to solve a single problem instance. `NN` denotes the inference time of the NN-based model without projection. The y-axis is in logarithmic scale.

tion, respectively. The only exception is for case5 and normal distribution, where `APT` is 10% better than `APTH`. Evidently, the effect of non-orthogonal splits using hyperplanes is more pronounced when net loads are uniformly distributed, as we observe that `APT` performs, on average, much better under a normal distribution. This could be attributed to the training data extending to a smaller number of critical regions when loads are normally distributed, which, in turn, nullifies the impact of a number of candidate splits.

We further observe that prescriptive trees perform competitively with `NN-prj` in terms of decision performance, resulting in a lower MCI in 5/12 cases examined. However, a significant percentage of `NN-prj` solutions may be infeasible and require a projection step. The rate of infeasibility seems to be increasing with the size of the network, with the worst-case being observed for case118 and case300, for both types of uncertainty.

We now discuss the efficacy of the surrogate learning algorithm proposed in Section 3.3.3. When `APTH` is fully grown, i.e., the algorithm terminates before the imposed time limit is

reached, it outperforms `APTH-rlx`, with the differences being small in general, except for case57, where both are optimal. For case188 and case300, the time limit is reached before `APTH` is fully grown, which leads to `APTH-rlx` outperforming `APTH`. Moreover, `APTH-rlx` outperforms `APT`, which considers only axis-parallel, in all cases but one, and is on par with `NN-prj`. Notably, `APTH-rlx` reduces the training time by over 95% in all cases compared to `APTH`; thus, `APTH-rlx` achieves a good trade-off between computational efficiency and prescriptive performance.

The results presented in Table 3.2 concern shallow trees ($\delta^{max} = 3$). Evidently, increasing the tree depth is expected to improve decision performance. We investigate this claim by evaluating the sensitivity of decision quality with respect to the maximum tree depth δ^{max}. Fig. 3.5 plots the out-of-sample MCI of `APTH` and `APTH-rlx` as a function of δ^{max} for three test cases and uniform uncertainty; the performance of `NN-prj` is also plotted for reference. In all examined cases, increasing δ^{max} leads to significant gains in performance for `APTH` and `APTH-rlx`, with the relative improvement being more pronounced for smaller values of δ^{max}. Moreover, `APTH` converges to better performance than `NN-prj` as δ^{max} increases, with a relatively small depth of $\delta^{max} = 5$ being sufficient for adequate performance.

We further investigate whether end-to-end learning improves over `Gurobi` in terms of inference speed. Fig. 3.6 plots the mean CPU time to solve or predict a single problem instance for a selection of models for uniform uncertainty (y-axis is in logarithmic scale). We denote `NN` as the NN-based model prior to projection. For `NN-prj`, we sum the inference time of `NN` and the time to solve the projection step, weighted by the probability of infeasibility. For `Gurobi`, we only consider CPU time to solve the problem and not the time to formulate it. As all tree-based models exhibit similar inference time, we only plot `APTH-rlx`.

Overall, `APTH-rlx` consistently leads to smaller CPU time compared to both `Gurobi` and `NN-prj`, and even outperforms `NN`. As expected, the mean CPU time of `Gurobi` increases with the size of the network. The `NN-prj` performance varies with its out-of-sample infeasibility rate. For medium to large-sized cases, when the infeasibility rate of `NN-prj` is below 10% and a post hoc projection is rarely required, e.g., case30 through case118, the inference time of `NN-prj` is smaller than that of `Gurobi`. However, in case300, when the infeasibility rate of `NN-prj` reaches over 40%, the required projection step to recover a feasible solution negates any improvement in inference speed and leads to higher CPU time than `Gurobi`. Thus a high infeasibility rate may nullify the intended purpose of applying end-to-end learning in the first place.

3.5.3 Results for more Challenging Test Cases

To evaluate the sensitivity with respect to the number of lines that face congestion, we repeat the previous experiment on more challenging test cases. Specifically, we examine performance on the *active power increase* (api) test cases [99], where the nominal $\mathbf{d}$ is increased.

Table 3.3: Number of congested lines, number of SVM models trained, and classifier accuracy (%), API test cases.

	Uniform		Normal	
	No. lines/models	mean/min acc. (%)	No. lines/models	mean/min acc. (%)
case5_api	3/3	99.83/99.59	2/1	99.95
case30_api	0/0	-	0/0	-
case39_api	10/4	97.59/92.92	7/4	99.46/98.81
case57_api	0/0	-	0/0	-
case118_api	16/11	95.52/85.54	15/12	94.56/78.81
case300_api	16/10	94.47/72.07	14/13	95.11/65.66

Table 3.4: Percentage (%) of MCI, $\delta^{max} = 3$, API test cases. Parentheses show the rate of infeasibility (%).

	Uniform				Normal			
	`APT`	`APTH`	`APTH-rlx`	`NN-prj`	`APT`	`APTH`	`APTH-rlx`	`NN-prj`
case5_api	0.05	**0.01**	0.02	0.85 (0.68)	0.02	**0.01**	0.13	0.71 (0.71)
case30_api	**0.00**	**0.00**	**0.00**	0.73 (3.03)	**0.00**	**0.00**	**0.00**	0.49 (3.01)
case39_api	0.93	0.65	0.75	**0.47 (13.11)**	0.57	0.52	**0.40**	0.68 (17.19)
case57_api	0.49	**0.00**	**0.00**	0.05 (0.03)	0.34	**0.00**	**0.00**	0.03 (0.02)
case118_api	22.07	17.65	16.04	**3.19 (86.91)**	19.27	18.74	18.36	**4.03 (93.08)**
case300_api	3.36	2.68	1.87	**1.52 (71.95)**	3.43	2.67	2.08	**1.75 (74.15)**

Table 3.3 presents the performance of the SVM classifiers on the more challenging test cases. Compared to Table 3.1, it is evident that the api test cases face congestion more frequently. For the smaller cases, the SVMs still perform quite well, with an average accuracy of over 97%. For case188_api and case300_api, the average accuracy remains around 95% for both types of uncertainty. However, we observe large variability based on the worst-case SVM performance, which is more pronounced for case300_api, where the worst-case performance is approximately 72% and 66% for uniform and normal distributions, respectively.

Table 3.4 presents the out-of-sample MCI under the more challenging operating conditions, alongside the infeasibility rate for `NN-prj`. Compared to Table 3.2, we observe an increase in MCI for larger networks, which is attributed to the more challenging nature of the underlying problems. This is especially pronounced for case118_api where the number of lines that face congestion is three times larger than case118. Furthermore, the infeasibility rate of `NN-prj` increases significantly for the larger cases, with an average infeasibility rate of approximately 90% for case118_api and 74% for case300_api, indicating the difficulty in predicting feasible decisions.

In terms of relative performance, the results are consistent with the previous experiments. Specifically, `APTH` consistently outperforms `APT`, while `APTH-rlx` performs similarly to `APTH` and outperforms `APT`. Interestingly, `APTH-rlx` even outperforms `APTH` for case39_api and normally distributed net loads. When comparing the tree-based models with `NN-prj`, we observe that `NN-prj` performs better only when its infeasibility rate is high. Notably,

`NN-prj` leads to significantly lower MCI for case118_api and case300_api for both types of uncertainty but has a high infeasibility rate in both cases. However, as previously shown in Fig. 3.6, a high infeasibility rate negates the respective gains of end-to-end learning over the traditional LP solver in terms of inference speed, making the choice of `NN-prj` counter-productive.

3.6 Conclusions

This chapter presented an interpretable approach for end-to-end learning of the solutions to a constrained optimization problem with feasibility guarantees, with an application to the DC-OPF problem. We developed prescriptive decision trees that learn a piecewise affine mapping from varying load data to DC-OPF solutions, using robust optimization to ensure the feasibility of decisions. We proposed domain-informed, non-orthogonal splits, using a set of hyperplanes to model the merit order curve and network congestion; for the latter, we utilized SVM classifiers that model expected line congestion as a function of varying load data. A comprehensive evaluation was conducted considering a number of test cases, different types of uncertainty, and various operating conditions. The results highlighted the efficacy of the proposed domain-informed, non-orthogonal splits, which led to an average performance increase of 36% compared to tree models using only axis-parallel splits. Further, shallow prescriptive trees with non-orthogonal splits of maximum depth of 3 outperformed NN-based benchmarks in approximately 46% of the experiments; a sensitivity analysis with respect to model complexity illustrated that the performance of prescriptive trees further improved as their depth increased. The proposed approach was also significantly faster than a state-of-the-art LP solver. Additional experiments under challenging operating conditions further validated the efficacy of the proposed approach. Overall, this study highlighted the benefits of encoding domain knowledge during model development, which not only achieves comparable performance to black-box, state-of-the-art benchmarks but also enables interpretability.

Future work may explore mapping contextual information, e.g., calendar variables or temperature forecasts, to OPF decisions. Another direction to explore is to use non-linear classifiers that also retain the computational tractability of the proposed policy, e.g., SVMs with a piecewise linear feature mapping. Finally, we aim to extend the proposed method to other variations of the DC-OPF problem, e.g., Security Constrained DC-OPF, as well as other linearized power flow formulations that also consider reactive power and voltage constraints.

Chapter 4

Resilient Energy Forecasting Against Missing Features

Résumé en Français

Les modèles de prévision énergétique déployés dans les applications industrielles sont confrontés à des incertitudes quant à la disponibilité des données, en raison de la latence du réseau, de dysfonctionnements des équipements ou d'attaques contre l'intégrité des données. En particulier, le cas où un sous-ensemble de données d'entrée qui a été utilisé pour l'apprentissage du modèle devient indisponible lorsque le modèle est utilisé de manière opérationnelle pose un défi majeur aux performances de prévision. Les solutions ad hoc, par exemple, l'apprentissage du modèle en considérant les données manquantes, peuvent fonctionner pour un petit nombre de données manquantes, mais elles deviennent rapidement impraticables, car le nombre de modèles augmente de façon exponentielle avec le nombre de données manquantes. Dans ce travail, nous présentons une approche fondée sur des principes pour introduire la résilience contre les données manquantes dans les applications de prévision énergétique via une optimisation robuste. Plus précisément, nous formulons un modèle de régression robuste qui résiste de manière optimale aux données manquantes au moment du test, en tenant compte à la fois des prévisions ponctuelles et probabilistes. Nous développons trois méthodes de solution pour la formulation robuste proposée, toutes conduisant à des problèmes de programmation linéaire, avec des degrés variables de maniabilité et de prudence. Nous fournissons une validation empirique approfondie des méthodes proposées dans les applications de prévision courantes, à savoir le prix de l'électricité, la charge, la production éolienne et la production solaire, la prévision, et nous comparons en outre avec des modèles de référence bien établis et des méthodes de traitement des caractéristiques manquantes, c'est-à-dire, imputation et réapprentissage. Ensuite, nous appliquons l'approche proposée dans un cadre intégré de prévision-optimisation, dans lequel nous prévoyons directement les décisions d'un agrégateur d'énergies renouvelables participant aux marchés de l'électricité. Nos résultats démontrent que l'approche d'optimisation robuste proposée surpasse les méthodes d'"imputation puis régression" et présente des performances similaires à l'approche de réapprentissage sans les données manquantes, tout en conservant un cout computationnel faible. À notre connaissance, il s'agit du premier travail qui introduit la résilience contre les données manquantes dans les prévisions énergétiques.

The work in this chapter extends the work previously published in [**J1**].

4.1 Introduction

Over the last decades, power systems have become increasingly data-centric [5], with short-term forecasting applications, in particular, being heavily reliant on available data. Short-term forecasting, ranging from a few minutes to a few days ahead, is key to ensuring the safe, reliable, and economic operation of modern power systems. It pertains to several applications, such as load [102], electricity price [103], wind production [104], and solar production [105], forecasting, which we refer to as *energy forecasting* [9]. The overarching goal in all applications is to estimate some characteristics of a target variable, such as the mean or a set of quantiles, at a future time interval as a function of associated features, which is subsequently used as input in a decision-making process. For instance, wind production is associated with wind speed, load is associated with temperature, and so forth.

Arguably, most research on energy forecasting focuses on improving predictive performance, which largely depends on data quality and availability. During the development of the forecasting model, i.e., at training time, potential missing data have either been recovered from a data retrieval mechanism or are treated in a preprocessing step. The implicit assumption is that input data would be complete and always available during the deployment of the forecasting model, i.e., at test time. However, real-world industrial applications may face several operational data management challenges that would emerge only after the model is deployed [18]. Undoubtedly, missing features in an operational setting, i.e., when a subset of features used for model training becomes unavailable at test time, may severely affect forecasting performance and lead to suboptimal decisions. Ideally, models deployed in industrial applications should be resilient [106], i.e., they should maintain consistent performance, without requiring excessive manual tuning or relying on empirical solutions, in case data are not available when needed.

There are several reasons that could lead to missing features (or *feature deletion*), e.g., malicious data-integrity attacks, network latency, and sensor failures. In Europe, for instance, system operators must publish, at specific times of day, various day-ahead predictions and system data, which are subsequently used by stakeholders as input to, e.g., electricity price forecasting models. However, a European Commission survey [107] that assesses the timeliness of data published on the ENTSO-E transparency platform finds that "for every data domain, fewer than 40% of users reported that data were always there when needed." Similarly, a survey by the European Centre for Medium-Range Weather Forecasts (ECMWF) [108] identifies user dissatisfaction regarding data turnaround of NWP model forecasts that are used as input to short-term renewable production forecasting. But even if data are typically provided in a timely fashion, data availability is not 100% guaranteed, and a robust fallback solution is always desirable if not necessary. Notably, however, uncertainty with respect to data availability is largely overlooked in the energy forecasting literature.

4.1.1 Related Work

Missing data at test time is a subject that receives scarce attention, contrary to missing data for model estimation which has been studied extensively in statistics [109]. For model estimation, there are several ways to deal with missing data depending on the missingness mechanism. If data entries are Missing Completely at Random (MCAR), i.e., the probability of a feature observation missing is independent of the rest of the variables, then observations with partial information can be ignored (complete case analysis); however, MCAR is a very strong assumption and complete case analysis does not apply to out-of-sample prediction. Conversely, data entries might be Missing at Random (MAR), i.e., the fact that a feature is missing and its (unobserved) value is independent, conditional on the observed features. Note that MAR still remains a strong assumption that is difficult to verify in practice. In this case, a valid methodology is to apply an imputation method such as mean imputation or Multiple Imputation [110] and then proceed with the regression (impute-then-regress), which, however, may incur a significant computational cost. Alternatively, missing data can be directly embedded within the learning model [111–113]. Allowing the model to directly learn from the patterns of missing data is also valid when data are Missing Not at Random (MNAR), i.e., the fact that a feature is missing depends on its (unobserved) value; note that imputation methods typically become invalid under MNAR. Nonetheless, to properly model the missingness mechanism, adaptive models — see, e.g., [111, 113] — require access to a training data set with missing data. In several applications of interest in power systems, missing data are retrieved ex-post, and thus training sets are complete, while the possibility of missing data at test time still remains. For instance, delays in publishing data on the ENTSO-E transparency platform [107] lead to missing data during prediction; however, missing data are eventually uploaded and thus the missingness mechanism cannot be modeled ex-post. Therefore, there is a need to develop energy forecasting models that are completely agnostic to the missingness mechanism.

In wind power forecasting, [114] examines two methods to handle missing features operationally, namely retraining without the missing features and impute-then-regress. Retraining consistently outperforms impute-then-regress and the difference is more pronounced when data are missing in batches. However, the number of additional models required is the combination of all features, which renders retraining impractical. In contrast, [115] proposes an iterative approach to jointly impute missing values and derive forecasts for wind power forecasting. Similarly to retraining, [116] develops several models to forecast electricity demand at a household level; given data availability at test time, the appropriate model is selected from a decision tree. The same approach, i.e., training several models to deal with uncertain data availability, is also considered in [51] to directly forecast the trading decisions of a renewable producer participating in a day-ahead market. An integrated imputation procedure to replace missing features within a long short-term memory network for solar production forecasting is presented in [117]; the performance, however, deteriorates as the percentage

of missing values increases, and no comparison against retraining is provided.

A related stream of research examines energy forecasting under data-integrity attacks, mostly dealing with uncertainty in the target variable and focusing on training data. Several load forecasting models are evaluated in [118] against attacks that affect the training process by permutating historical observations; none of the models considered provides adequate performance under large-scale attacks. A subsequent work [119] leverages robust statistics and shows that selecting the ℓ_1 norm as the loss function proves resilient even under large-scale attacks. Similarly, [120] studies the robustness of short-term wind production forecasting models under false-data injection attacks, considering both point and probabilistic forecasts. Conversely, [121] formulates a poison attack methodology to exploit load forecasting models. Tangentially related to data-integrity attacks on load forecasting are works on outlier detection [122–124], which focus on identifying attacks that have occurred and replacing any corrupt data. On the other hand, [125] and [126] consider adversarial attacks at test time applied to load forecasting. Specifically, [125] shows that manipulating temperature values at test time leads to a significant decrease in accuracy and increased operational costs, whereas [126] employs Bayesian learning to enhance the robustness of deep-learning-based models under several adversarial attacks.

One way to view data-integrity attacks is as processes that introduce feature uncertainty; the same also applies to the case of missing features. Indeed, advanced forecasting models are typically cognizant of some form of feature uncertainty, even if this is unknown to the forecaster, and address it with regularization, e.g., ℓ_1-regularized (lasso) regression [127] or ℓ_2-regularized (ridge) regression. Introducing randomness during training also enhances model robustness; popular methods include bagging and sampling a subset of features, as in randomized ensembles such as Random Forests [56], using dropout layers in deep learning models, and generative adversarial networks, among others. In fact, [126] shows that regularization and treating model parameters as random variables increase robustness in load forecasting applications. Interestingly, a big part of the success of regularization methods is their "hidden" robustness. For example, both the ℓ_1-regularized [128] and the ℓ_2-regularized [129] regressions are equivalent to the solution of robust optimization problems [130]. Beyond regularized regression, several applications of robust optimization in different machine learning areas exist [131], such as classification [132] and deep learning [133]. We highlight [134], which describes a robust learning support vector machine algorithm for classification where a different set of features might be missing at each observation, as a core foundation of our current work. Uniform feature deletion, i.e., the same features missing across all observations, is considered as an alternate setting in [134], which is deemed as not efficiently solvable, except for a small number of features through enumeration. Notably, the connection between feature uncertainty, robust optimization, and regularization is rarely discussed in the context of energy forecasting.

4.1.2 Aim and Contribution

In this chapter, we present a robust optimization approach to design energy forecasting models that are optimally resilient when a subset of features used for model training becomes unavailable at test time. We formulate a robust regression model, readily applicable to point and probabilistic forecasting, which minimizes the worst-case loss when a subset of features is missing. We present three solution methods for the resulting robust optimization formulation considering the quantile loss, all leading to LP problems: *(i)* a method based on enumeration, which is practical for a small number of features; *(ii)* a deterministic reformulation, which, although tractable, provides conservative results thus being more suitable for the main setting of [134] with different features missing across observations, and *(iii)* an affinely adjustable reformulation [135], which offers an efficient solution method to the uniform feature deletion setting of [134], remains tractable, and is less conservative than the previous method. We further consider extensions to piecewise linear loss functions, which can be used to approximate quadratic, and in general convex, loss functions, and to the setting of integrated forecasting and optimization. We first evaluate the proposed methods in prevalent energy forecasting applications, namely electricity price, load, wind production, and solar production, forecasting, considering a day-ahead horizon. Next, we provide further validation in two additional case studies: wind production forecasting in an intra-hour horizon and directly forecasting the trading decisions of a renewable producer participating in a day-ahead market [51]. We compare the proposed approach against established benchmark models, including regularization and randomization-based training, coupled with different methods of handling missing data, i.e., imputation and retraining. We demonstrate that the proposed approach outperforms impute-then-regress models and exhibits similar performance to retraining without the missing features, while preserving practicality. Notably, by evenly distributing coefficient weights across features during training, it hedges against missing the most important feature at test time.

Our main contribution is two-fold. Firstly, we propose a robust regression model that is, by design, resilient against missing features at test time, with the following key advantages: *(i)* leads to consistent performance and lower model degradation when features are missing, including the worst-case scenario of missing the most important feature, *(ii)* is agnostic to the missingness mechanism, and *(iii)* offers computational tractability through LP reformulations, which can also be applied to approximations of quadratic, and in general convex, loss functions. Secondly, we benchmark against current state-of-the-art forecasting models and methods to handle feature uncertainty for both point and probabilistic forecasting, and quantify the aforementioned advantages in several prevalent energy forecasting applications, as well as, an application on trading renewable production in electricity markets. To the best of our knowledge, this is the first work that introduces resilient energy forecasting and benchmarks against missing features at test time, a situation that may emerge in industrial applications after the forecasting model is deployed.

4.1.3 Chapter Outline

The remainder of this chapter is organized as follows. Section 4.2 presents the mathematical background and the proposed model. Section 4.3 develops the solution methodology. Section 4.4 presents the experimental setup and the input data and discusses the numerical results for the energy forecasting case study. Section 4.5 presents additional numerical experiments in two case studies, namely feature-driven trading of renewable production in a day-ahead market and wind production forecasting in a very short-term horizon. Section 4.6 concludes and provides directions for future work.

4.2 Preliminaries and Proposed Model

In this section, we present preliminaries on linear regression (in Subsection 4.2.1), describe the process of modeling feature uncertainty (in Subsection 4.2.2), and present the proposed robust formulation (in Subsection 4.2.3).

4.2.1 Preliminaries

Let $y_i \in \mathbb{R}$ be the target variable (e.g., electricity prices, load, wind/solar production) and $\mathbf{x}_i \in \mathbb{R}^p$ be a p-size vector of associated features from a set $\mathcal{P} = \{1, \ldots, p\}$ (e.g., weather data, historical data), with subscript i denoting an observation from a training data set $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$ of n observations. Throughout, the term $[n]$ is used as shorthand for $\{1, \ldots, n\}$. A regression model is defined as a mapping function $f \in \mathcal{F} : \mathbf{x} \in \mathbb{R}^p \rightarrow y \in \mathbb{R}$, where $\mathcal{F}$ is a hypothesis space. Here, we focus exclusively on linear models parameterized by a set of coefficients $\mathbf{w} \in \mathbb{R}^p$. To ease the notation, we assume that the bias term is modeled by appending a constant vector of ones to $\mathbf{x}$. The problem of estimating the parameters of a linear regression model is given by:

$$\min_{\mathbf{w}} \sum_{i \in [n]} l(y_i - \mathbf{w}^\intercal \mathbf{x}_i), \tag{4.1}$$

where l is the selected loss function to be minimized [1]. Typical choices are the quadratic loss $l(\cdot) = (\cdot)^2$, which leads to a Least Squares (LS) model, and the ℓ_1 norm $l(\cdot) = |\cdot|$, which leads to a Least Absolute Deviations (LAD) model.

Both the LS and the LAD models are employed to derive point estimates of the target variable. Dealing, however, with uncertainty necessitates the usage of probabilistic forecasts as an input in many decision-making processes. Quantile Regression (QR) [136] is a general approach to derive probabilistic forecasts in the form of predictive quantiles. A QR model

[1] Note that the linear regression model can straightforwardly accommodate nonlinear dependencies by considering polynomial terms, local weights, etc.

minimizes the quantile (pinball) loss for a specific quantile τ, defined as:

$$l(y_i - \mathbf{w}^\intercal \mathbf{x}_i; \tau) = \tau(y_i - \mathbf{w}^\intercal \mathbf{x}_i)^+ + (1-\tau)(\mathbf{w}^\intercal \mathbf{x}_i - y_i)^+ \\ = \max\left(\tau(y_i - \mathbf{w}^\intercal \mathbf{x}_i), (\tau - 1)(y_i - \mathbf{w}^\intercal \mathbf{x}_i)\right), \quad (4.2)$$

where $(t)^+ = \max(0, t)$. In fact, the ℓ_1 loss can be viewed as a special case of the quantile loss estimating the 50-th quantile (median). This is straightforward to show considering that $|x| = \max(x, -x)$, $\tau = 0.5$, and that scaling the objective does not affect the solution.

4.2.2 Modeling Feature Uncertainty

Our goal is to formulate a robust regression model, which accounts for missing features after model deployment (i.e., at test time) and maintains consistent performance. To this end, we introduce binary variables $\boldsymbol{\alpha} \in \{\mathbf{0}, \mathbf{1}\}^p$ and model the availability of the i-th feature observation as $\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})$, where $\odot$ is the element-wise multiplication, and α_j equals 1 if the j-th feature is missing (i.e., missing features are set to zero).

At this point, there are two issues that relate to energy forecasting applications that warrant a discussion.

First, in practice, some features cannot be deleted at test time. It makes little sense to delete, e.g., calendar variables, which are regularly employed in energy forecasting. Let $\mathcal{J} \subseteq \mathcal{P}$ denote the subset of features that *can* be deleted at test time, and $\mathcal{C} = \mathcal{P} - \mathcal{J}$ denote the set of features that *cannot* be deleted. It is straightforward to account for this case by setting $\alpha_j = 0 \; \forall j \in \mathcal{C}$, therefore features in $\mathcal{C}$ cannot go missing.

Second, a standard technique to model nonlinear relationships within a linear regression is to include polynomial and interaction terms of associated features. A classic example in energy forecasting is to add quadratic and cubic terms of temperature in load forecasting models [137]. It follows that all features derived from the same variable should be treated as a group of features (i.e., if missing, they are all missing).

We address both the aforementioned issues by enforcing a set of equality constraints, $\boldsymbol{M}\boldsymbol{\alpha} = \mathbf{0}$, where $\boldsymbol{M} \in \mathbb{R}^{m \times p}$. Namely, if the first feature cannot be deleted (i.e., $\alpha_1 = 0$), then the row vector $[1, \mathbf{0}]$ is appended to $\boldsymbol{M}$. Similarly, if $\alpha_1 = \alpha_2$, i.e., they represent a group of features, then $[-1, 1, \mathbf{0}]$ is appended to $\boldsymbol{M}$.

Following the above, we consider the discrete uncertainty set:

$$\mathcal{U} = \{\boldsymbol{\alpha} \mid \boldsymbol{\alpha} \in \{\mathbf{0}, \mathbf{1}\}^p, \sum_{j \in [p]} \alpha_j = \Gamma, \boldsymbol{M}\boldsymbol{a} = \mathbf{0}\}, \quad (4.3)$$

that models the representation of feature availability, where Γ (integer) is the budget of robustness (for $\Gamma = 0$, all features are present, whereas for $\Gamma = p$ all features are missing).

4.2.3 Proposed Robust Formulation

The proposed robust formulation employs the representation of the availability of the i-th feature observation $\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})$, and builds a robust regression model using the uncertainty

set (4.3), as follows:

$$\min_{\mathbf{w}} \max_{\boldsymbol{\alpha}\in\mathcal{U}} \sum_{i\in[n]} l(y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha}))). \tag{4.4}$$

Inspired by [134], we refer to model (4.4) as Feature-Deletion Robust Regression (FDRR). The problem objective is to minimize the worst-case loss when Γ features are missing, assuming that the *same* features are missing across all observations[2], while also respecting additional constraints arising from the fact that a subset of features could not be deleted or that different features might be grouped. In the latter case, Γ is selected appropriately to account for feature groups.

Interestingly, minimizing the worst-case loss when a subset of features is missing (4.4) shares many similarities with feature selection and feature importance. On one hand, feature selection concerns methods to improve out-of-sample predictive accuracy by optimally selecting a feature vector. Usually, this involves gradually adding features to the model. Intuitively, a feature that improves performance will also have a significant impact when deleted; however, the problems are not equivalent. Feature importance, on the other hand, concerns post-hoc methods to assess the individual feature contribution to model performance, with the goal to improve explainability — see, e.g., the permutation importance metric proposed in [56]. Notably, our proposal effectively optimizes the model based on feature importance by design.

Next, we consider (4.4), using the quantile loss, which, along with its special case — the ℓ_1 loss — are of particular interest in energy forecasting applications. Hence, using the quantile loss representation (4.2) in (4.4), we obtain the following robust optimization problem:

$$\begin{aligned}\min_{\mathbf{w}} \max_{\boldsymbol{\alpha}\in\mathcal{U}} \sum_{i\in[n]} \max\Big(&\tau(y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha}))),\\ &(\tau - 1)(y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})))\Big).\end{aligned} \tag{4.5}$$

Note that setting $\tau = 0.5$ and scaling the objective would yield the robust formulation for the ℓ_1 regression:

$$\min_{\mathbf{w}} \max_{\boldsymbol{\alpha}\in\mathcal{U}} \sum_{i\in[n]} \big|y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha}))\big|.$$

For practical reasons, we can recast (4.5) using a robust constraint, introducing auxiliary $t \in \mathbb{R}$, as follows:

$$\min_{\mathbf{w},t} \quad t, \tag{4.6a}$$

$$\begin{aligned}\text{s.t.} \quad &\sum_{i\in[n]} \max\Big(\tau(y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha}))),\\ &\qquad (\tau - 1)(y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})))\Big) \le t, \qquad \forall \boldsymbol{\alpha} \in \mathcal{U},\end{aligned} \tag{4.6b}$$

[2]Note that [134] considers the case where *different* features are missing across observations, which leads to a more conservative problem.

which involves an inequality that contains the sum of maxima of linear functions. Indeed, in a deterministic setting, i.e., in the absence of $\forall \boldsymbol{\alpha} \in \mathcal{U}$, constraint (4.6b) could be straightforwardly and, most importantly exactly, reformulated using auxiliary variables. Consider a specific instance of $\boldsymbol{\alpha}$, say $\boldsymbol{\alpha}_k$. Then, the deterministic reformulation of (4.6b) would be:

$$\min_{\mathbf{w},t,\boldsymbol{\xi}} \quad t, \tag{4.7a}$$

$$\text{s.t.} \quad \sum_{i\in[n]} \xi_i \leq t, \tag{4.7b}$$

$$\tau(y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha}_k))) \leq \xi_i, \qquad i \in [n], \tag{4.7c}$$

$$(\tau - 1)(y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha}_k))) \leq \xi_i, \qquad i \in [n], \tag{4.7d}$$

where $\xi_i \in \mathbb{R}$ is an auxiliary variable, and $\boldsymbol{\xi}$ an appropriate vector. However, care must be given when applying deterministic reformulations in a robust setting, as they could lead to over-conservative solutions [135]. It is interesting to note that (4.7) is essentially equivalent to "retraining" for a specific combination of missing features. In fact, repeating (4.7) for all elements of all sets $\mathcal{U}$ constructed by the admissible values of $\Gamma = \{0, \ldots, |\mathcal{J}|\}$ retrieves the solution proposed in [51, 114, 116], i.e., retraining without the missing features.

Before proceeding to the solution methods of (4.6), let us revisit the uncertainty set, $\mathcal{U}$, and consider its convex hull, represented by the polyhedral uncertainty set, $\mathcal{A}$,

$$\mathcal{A} = \{\boldsymbol{\alpha} \mid \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{1}, \sum_{j\in[p]} \alpha_j = \Gamma, \boldsymbol{M}\boldsymbol{a} = \mathbf{0}\}. \tag{4.8}$$

Note that $\boldsymbol{M}$ is unimodular, as all of its entries are $0, 1$ or -1, and at most two entries per column are non-zero, at which case the column-wise sum is zero. Since Γ is also an integer, all vertices of $\mathcal{A}$ occur at integer values, therefore the LP relaxation of the inner max problem over $\boldsymbol{\alpha}$ in (4.5) is exact. Evidently, replacing $\mathcal{U}$ by its convex hull $\mathcal{A}$ in constraint (4.6b) also yields equivalent solutions [138, Ch. 10].

4.3 Solution Methods

In this section, we present three methods to solve the robust optimization problem (4.6), namely, we describe a method suitable for a small number of features (in Subsection 4.3.1), we present two reformulation methods that lead to tractable problems (in Subsections 4.3.2 and 4.3.3), and we discuss extensions to piecewise linear loss functions and an integrated forecasting and optimization setting (in Subsections 4.3.4 and 4.3.5).

4.3.1 Vertex Enumeration of FDRR (FDRR-V)

Typically, most energy forecasting problems have relatively large sample sizes (e.g., n is in the order of 10^4 for series with hourly resolution) compared to the number of features, i.e., $n >> p$. Hence, if the number of features is small, problem (4.6) could be solved by

vertex enumeration of the uncertainty set $\mathcal{A}$. In fact, since all vertices of $\mathcal{A}$ are contained in the original finite set $\mathcal{U}$, vertex enumeration of $\mathcal{A}$ is equivalent to an enumeration of the elements of $\mathcal{U}$.

Let V denote the number of elements of $\mathcal{U}$, equivalently the number of vertices of $\mathcal{A}$; assuming no grouping constraints, $V = \binom{|\mathcal{J}|}{\Gamma}$ (grouping constraints would further reduce V). Let ξ_i^k be an auxiliary variable, for each $i \in [n]$ and each vertex $k \in [V]$. Constraint (4.6b) is equivalently written as:

$$\sum_{i\in[n]} \xi_i^k \leq t, \qquad k \in [V], \tag{4.9a}$$

$$y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha}_k)) \leq \tfrac{1}{\tau}\xi_i^k, \qquad i \in [n], k \in [V], \tag{4.9b}$$

$$-y_i + \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha}_k)) \leq \tfrac{1}{1-\tau}\xi_i^k, \qquad i \in [n], k \in [V], \tag{4.9c}$$

where constraints (4.9a)–(4.9c) essentially enumerate the deterministic reformulation (4.7b)–(4.7d) for all vertices. Hence, the solution of FDDR by vertex enumeration, referred to as FDRR-V, is given by the following deterministic LP problem:

$$\text{FDRR-V:} \quad \min_{\mathbf{w},t,\boldsymbol{\xi}} t, \quad \text{s.t.} \quad (4.9a) - (4.9c), \tag{4.10}$$

where $\boldsymbol{\xi}$ is an appropriate vector that represents variables ξ_i^k. FDRR-V ensures that the worst-case $\boldsymbol{\alpha}$ remains the same across all observations and leads to an exact solution of (4.6). Evidently, for a specific realization of uncertainty, say $\boldsymbol{\alpha}_k$, retraining — see (4.7) — sets a lower bound to the in-sample error of FDRR-V, which subsumes all individual cases. However, if the number of features is not small enough, unavoidably V gets large enough to render both retraining and FDRR-V at least impractical, in terms of models to be trained and LP problems to be solved, respectively.

4.3.2 Reformulation of FDRR (FDRR-R)

An alternative approach is to first apply deterministic reformulation to the maxima terms in (4.6b), leading to:

$$\sum_{i\in[n]} \xi_i \leq t, \tag{4.11a}$$

$$y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) \leq \tfrac{1}{\tau}\xi_i, \qquad i \in [n], \forall \boldsymbol{\alpha} \in \mathcal{A}, \tag{4.11b}$$

$$-y_i + \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) \leq \tfrac{1}{1-\tau}\xi_i, \qquad i \in [n], \forall \boldsymbol{\alpha} \in \mathcal{A}. \tag{4.11c}$$

In turn, (4.11b)-(4.11c) are further reformulated to deterministic constraints. Since both constraints are similar, we illustrate the reformulation for (4.11b).

Changing the order of multiplication in the left-hand side of (4.11b), and considering that the inequality holds $\forall \boldsymbol{\alpha} \in \mathcal{A}$, i.e., the worst-case of $\boldsymbol{\alpha}$, constraint (4.11b) is equivalent to:

$$y_i - \mathbf{w}^\intercal \mathbf{x}_i + \max_{\boldsymbol{\alpha}\in\mathcal{A}} (\mathbf{w} \odot \mathbf{x}_i)^\intercal \boldsymbol{\alpha} \leq \tfrac{1}{\tau}\xi_i, \qquad i \in [n]. \tag{4.12}$$

The inner max in (4.12) can be written with explicit constraints, for the i-th observation, as follows:

$$\max_{\boldsymbol{\alpha}} \ (\mathbf{w} \odot \mathbf{x}_i)^\intercal \boldsymbol{\alpha}, \tag{4.13a}$$

$$\text{s.t.} \qquad \boldsymbol{\alpha} \leq \mathbf{1} : \quad \boldsymbol{\mu}_i^+ \geq \mathbf{0}, \tag{4.13b}$$

$$\sum_{j\in[p]} \alpha = \Gamma : \quad \zeta_i^+, \tag{4.13c}$$

$$\boldsymbol{M}\boldsymbol{\alpha} = \mathbf{0} : \quad \boldsymbol{\pi}_i^+, \tag{4.13d}$$

$$\boldsymbol{\alpha} \geq \mathbf{0}, \tag{4.13e}$$

where $\boldsymbol{\mu_i^+}, \zeta_i^+, \boldsymbol{\pi}_i^+$ are dual variables of appropriate size. Since problem (4.13a) is linear in $\boldsymbol{\alpha}$, it can be replaced by its dual:

$$\min_{\boldsymbol{\mu}_i^+\geq\mathbf{0},\zeta_i^+,\boldsymbol{\pi}_i^+} \ \sum_{j\in[p]} \mu_{ij}^+ + \Gamma\zeta_i^+, \tag{4.14a}$$

$$\text{s.t.} \quad \boldsymbol{\mu}_i^+ + \zeta_i^+ + \boldsymbol{M}^\intercal\boldsymbol{\pi}_i^+ \geq \mathbf{x}_i \odot \mathbf{w}, \tag{4.14b}$$

and hence, the inner max in (4.12) can be replaced by (4.14). Evidently, the min operator becomes redundant. Hence, constraint (4.11b) is replaced by the following inequalities:

$$y_i - \mathbf{w}^\intercal\mathbf{x}_i + \sum_{j\in[p]} \mu_{ij}^+ + \Gamma\zeta_i^+ \leq \tfrac{1}{\tau}\xi_i, \qquad i \in [n], \tag{4.15a}$$

$$\boldsymbol{\mu}_i^+ + \zeta_i^+ + \boldsymbol{M}^\intercal\boldsymbol{\pi}_i^+ \geq \mathbf{x}_i \odot \mathbf{w}, \qquad i \in [n], \tag{4.15b}$$

$$\boldsymbol{\mu}_i^+ \geq \mathbf{0}, \qquad i \in [n]. \tag{4.15c}$$

Similarly, constraint (4.11c) is replaced by:

$$-y_i + \mathbf{w}^\intercal\mathbf{x}_i + \sum_{j\in[p]} \mu_{ij}^- + \Gamma\zeta_i^- \leq \tfrac{1}{1-\tau}\xi_i, \qquad i \in [n], \tag{4.15d}$$

$$\boldsymbol{\mu}_i^- + \zeta_i^- + \boldsymbol{M}^\intercal\boldsymbol{\pi}_i^- \geq -\mathbf{x}_i \odot \mathbf{w}, \qquad i \in [n], \tag{4.15e}$$

$$\boldsymbol{\mu}_i^- \geq \mathbf{0}, \qquad i \in [n]. \tag{4.15f}$$

Summarizing, the reformulation of the FDRR, referred to as FDRR-R, yields the following deterministic LP problem:

$$\text{FDRR-R:} \qquad \min_{\substack{\mathbf{w},t,\boldsymbol{\xi},\\ \boldsymbol{\mu}^+,\boldsymbol{\mu}^-,\boldsymbol{\zeta}^+,\boldsymbol{\zeta}^-,\boldsymbol{\pi}^+,\boldsymbol{\pi}^-}} \quad t, \quad \text{s.t.} \quad (4.15\text{a}) - (4.15\text{f}). \tag{4.16}$$

Note, however, that the uncertainty is now spread over several constraints, separately optimizing the worst-case loss of each observation. This worst-case loss may occur for different $\boldsymbol{\alpha}$ per observation, i.e., different features might be missing at each observation, which leads to the representation of uncertainty considered in [134]. When modeling feature uncertainty in Section 4.2.2, however, we assumed the same $\boldsymbol{\alpha}$ across all observations. Evidently, FDRR-R considers a more general case and thus provides a conservative approximation of (4.6), which is more pessimistic.

4.3.3 Affinely Adjustable Reformulation of FDRR (FDRR-AAR)

The conservativeness introduced by the reformulation of the maxima terms is reduced using adjustable auxiliary variables [135]. As ξ_i is not a true decision variable, it may be adjusted to the realization of $\boldsymbol{\alpha}$ as long as inequalities (4.11b) and (4.11c) hold. To this end, we introduce linear decision rules $v_i \in \mathbb{R}, \boldsymbol{u}_i \in \mathbb{R}^p$, and substitute $\xi_i = v_i + \boldsymbol{u}_i^\intercal \boldsymbol{\alpha}$, i.e., ξ_i is an affine function of uncertainty. Constraint (4.6b) is written as:

$$\sum_{i\in[n]} (v_i + \boldsymbol{u}_i^\intercal \boldsymbol{\alpha}) \leq t, \qquad \forall \boldsymbol{\alpha} \in \mathcal{A}, \tag{4.17a}$$

$$y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) \leq \tfrac{1}{\tau}(v_i + \boldsymbol{u}_i^\intercal \boldsymbol{\alpha}), \qquad i \in [n], \forall \boldsymbol{\alpha} \in \mathcal{A}, \tag{4.17b}$$

$$-y_i + \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) \leq \tfrac{1}{1-\tau}(v_i + \boldsymbol{u}_i^\intercal \boldsymbol{\alpha}), \qquad i \in [n], \forall \boldsymbol{\alpha} \in \mathcal{A}, \tag{4.17c}$$

Similarly to (4.12), constraint (4.17a) is equivalent to:

$$\sum_{i\in[n]} v_i + \max_{\boldsymbol{\alpha}\in\mathcal{A}} \sum_{i\in[n]} \boldsymbol{u}_i^\intercal \boldsymbol{\alpha} \leq t,$$

and introducing dual variables $\boldsymbol{\mu}$, ζ, and $\boldsymbol{\pi}$ (similarly to (4.13b), (4.13c), and (4.13d), respectively), constraint (4.17a) is replaced by:

$$\sum_{i\in[n]} v_i + \sum_{j\in[p]} \mu_j + \Gamma\zeta \leq t, \tag{4.18a}$$

$$\boldsymbol{\mu} + \zeta + \boldsymbol{M}^\intercal \boldsymbol{\pi} \geq \sum_{i\in[n]} \boldsymbol{u}_i, \tag{4.18b}$$

$$\boldsymbol{\mu} \geq \mathbf{0}. \tag{4.18c}$$

Constraint (4.17b) is equivalent to:

$$y_i - \mathbf{w}^\intercal \mathbf{x}_i + \max_{\boldsymbol{\alpha}\in\mathcal{A}} (\mathbf{x}_i \odot \mathbf{w} - \tfrac{1}{\tau}\boldsymbol{u}_i)^\intercal \boldsymbol{\alpha} \leq \tfrac{1}{\tau} v_i, \qquad i \in [n],$$

and similarly to (4.12), constraint (4.17b) is replaced by:

$$y_i - \mathbf{w}^\intercal \mathbf{x}_i + \sum_{j\in[p]} \mu_{ij}^+ + \Gamma\zeta_i^+ \leq \tfrac{1}{\tau} v_i, \qquad i \in [n], \tag{4.19a}$$

$$\boldsymbol{\mu}_i^+ + \zeta_i^+ + \boldsymbol{M}^\intercal \boldsymbol{\pi}_i^+ \geq \mathbf{x}_i \odot \mathbf{w} - \tfrac{1}{\tau}\boldsymbol{u}_i, \qquad i \in [n], \tag{4.19b}$$

$$\boldsymbol{\mu}_i^+ \geq \mathbf{0}, \qquad i \in [n], \tag{4.19c}$$

whereas constraint (4.17c) is replaced by:

$$-y_i + \mathbf{w}^\intercal \mathbf{x}_i + \sum_{j\in[p]} \mu_{ij}^- + \Gamma\zeta_i^- \leq \tfrac{1}{1-\tau} v_i, \qquad i \in [n], \tag{4.20a}$$

$$\boldsymbol{\mu}_i^- + \zeta_i^- + \boldsymbol{M}^\intercal \boldsymbol{\pi}_i^- \geq -\mathbf{x}_i \odot \mathbf{w} - \tfrac{1}{1-\tau}\boldsymbol{u}_i, \qquad i \in [n], \tag{4.20b}$$

$$\boldsymbol{\mu}_i^- \geq \mathbf{0}, \qquad i \in [n]. \tag{4.20c}$$

Lastly, the affinely adjustable reformulation of the FDRR (FDRR-AAR) is equivalent to the following deterministic LP problem:

$$\text{FDRR-AAR:} \quad \min_{\substack{\mathbf{w},t,\boldsymbol{v},\boldsymbol{u},\boldsymbol{\mu},\boldsymbol{\mu}^+,\boldsymbol{\mu}^-,\\ \zeta,\boldsymbol{\zeta}^+,\boldsymbol{\zeta}^-,\boldsymbol{\pi},\boldsymbol{\pi}^+,\boldsymbol{\pi}^-}} t, \quad \text{s.t.} \quad (4.18a) - (4.20c). \tag{4.21}$$

Note that we are still optimizing over the worst-case loss per observation, hence FDRR-AAR is a conservative approximation of (4.6). However, allowing for adjustable auxiliary variables reduces the induced conservativeness compared to FDRR-R. On the other hand, FDRR-AAR leads to a tractable LP problem, contrary to FDRR-V which leads to an LP problem whose size grows combinatorially. This trade-off between tractability and conservativeness places FDRR-AAR as an intermediate solution between FDRR-V and FDRR-R.

4.3.4 Extension to Piecewise Linear Loss Functions

In what follows, we discuss an extension of our proposal to piecewise linear loss functions.

Consider a piecewise linear loss function

$$l(y - \mathbf{w}^\intercal \mathbf{x}; \boldsymbol{c}, \boldsymbol{b}) = \max_{j=1,\dots,m} (c_j(y - \mathbf{w}^\intercal \mathbf{x} + b_j)), \tag{4.22}$$

parameterized by the m-size vectors $\boldsymbol{c}, \boldsymbol{b}$. Note that the quantile loss is a special case of (4.22), where $m = 2$, $\boldsymbol{c} = [\tau, \tau - 1]^\intercal$, and $\boldsymbol{b} = \mathbf{0}$. Using the piecewise linear loss function (4.22), the FDRR model (4.6) becomes

$$\begin{aligned} \min_{\mathbf{w},t} \quad & t, \\ \text{s.t.} \quad & \sum_{i\in[n]} \max_{j\in[m]} \Big(c_j(y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) + b_j) \Big) \leq t, \qquad \forall \boldsymbol{\alpha} \in \mathcal{U}, \end{aligned}$$

which can be solved with any of the proposed solution methods. For the solution with vertex enumeration, FDRR-V, we enumerate the deterministic reformulation for all vertices and all m vectors; hence, (4.9b)-(4.9c) are replaced by

$$c_j(y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha}_k)) + b_j) \leq \xi_i^k, \qquad i \in [n], k \in [V], j \in [m].$$

For FDRR-R, (4.11b)-(4.11c) are replaced by

$$c_j(y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) + b_j) \leq \xi_i, \qquad i \in [n],\ j \in [m], \forall \boldsymbol{\alpha} \in \mathcal{A},$$

which are further reformulated to deterministic constraints similarly to (4.11b)-(4.11c) — see (4.15). For FDRR-AAR, (4.17b)-(4.17c) are replaced by

$$c_j(y_i - \mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) + b_j) \leq (v_i + \boldsymbol{u}_i^\intercal \boldsymbol{\alpha}), \qquad i \in [n],\ j \in [m],\ \forall \boldsymbol{\alpha} \in \mathcal{A},$$

which are further reformulated similarly to (4.11b)-(4.11c) — see (4.19) and (4.20).

The piecewise linear loss functions can be used to approximate quadratic, and in general convex, loss functions. Consider for example an FDRR model with a quadratic loss (LS). It is straightforward to solve the robust regression model with vertex enumeration, but this approach is only practical for a small number of features. For a larger number of features,

it is not straightforward to reformulate the robust problem, as the quadratic loss leads to robust constraints that are quadratic in $\boldsymbol{\alpha}$ and thus more challenging to handle — see [67, Ch. 16]. Hence, a reasonable approach would be to use a piecewise linear function to approximate the quadratic loss and solve the resulting robust problem as described above. In general, the piecewise linearization becomes relevant in first-order approximations of the loss function, e.g., in the context of adversarial training [139].

4.3.5 Extension to Integrated Forecasting-Optimization

We further discuss an extension of our proposal to the case of directly forecasting the decisions of an optimization problem.

From Chapter 2, recall the definition of a contextual stochastic optimization problem given by

$$\min_{\mathbf{z}\in\mathcal{Z}} \mathbb{E}_y[c(\mathbf{z}; y)|\mathbf{x} = \mathbf{x}_0], \tag{4.26}$$

where $\mathbf{z} \in \mathbb{R}^{d_z}$ denotes the decision vector, $\mathcal{Z}$ is a convex set of feasible solutions, $c(\cdot)$ is a convex cost function, y denotes the uncertain problem parameter[3], $\mathbf{x}$ denotes associated contextual information (features), and the expectation is taken with respect to the distribution of y conditioned on $\mathbf{x} = \mathbf{x}_0$.

Given a training data set $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$ of n observations, we consider a feature-driven policy function f that maps contextual information $\mathbf{x}$ to decisions $\mathbf{z}$, i.e., directly forecasting the problem solutions. If f belongs to the class of linear models, one approach to find the set of linear coefficients $\mathbf{w}$ that minimize the in-sample decision cost is given by

$$\min_{\mathbf{w}} \quad \sum_{i\in[n]} c(\mathbf{w}^\intercal \mathbf{x}_i; \mathbf{y}_i), \tag{4.27a}$$

$$\text{s.t.} \quad \mathbf{w}^\intercal \mathbf{x}_i \in \mathcal{Z}, \qquad i \in [n], \tag{4.27b}$$

where $\mathbf{z}$ is replaced with a linear decision rule and (4.27b) ensures that in-sample decisions are feasible. An alternative approach would be to include constraint violation penalties in the objective function, as proposed in [140]. For an out-of-sample observation, say $\mathbf{x}_0$, the optimal solution is computed directly from $\mathbf{z}_0 = \mathbf{w}^\intercal \mathbf{x}_0$, which is highly efficient and effectively bypasses the need for an optimization solver. However, as there is no guarantee that $\mathbf{z}_0$ will be feasible, an additional projection step onto the feasible set $\mathcal{Z}$ might be required.

Evidently, if a subset of features is unavailable at test time, the decision quality of (4.27) will also be affected. We consider an extension of the proposed FDRR (4.6) to the case where

[3]For simplicity, we assume y is scalar here.

we directly forecast decisions, given by

$$\min_{\mathbf{w},t} \quad t, \tag{4.28a}$$

$$\text{s.t.} \quad \sum_{i\in[n]} c(\mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1}-\boldsymbol{\alpha})); \mathbf{y}_i) \leq t, \qquad \forall \boldsymbol{\alpha} \in \mathcal{U}, \tag{4.28b}$$

$$\mathbf{w}^\intercal(\mathbf{x}_i \odot (\mathbf{1}-\boldsymbol{\alpha})) \in \mathcal{Z}, \qquad i \in [n], \forall \boldsymbol{\alpha} \in \mathcal{U}. \tag{4.28c}$$

Effectively, we are searching for a feature-driven policy that minimizes the worst-case decision cost when Γ features are missing. Constraint (4.28c) ensures that the in-sample decisions are feasible for all realizations of $\boldsymbol{\alpha}$. In practice, as the feasibility of a forecast decision depends on $\mathbf{x}$, we propose relaxing (4.28c) during training. Then, as discussed in Section 4.3.4, we can use a piecewise linear function to approximate the convex loss $c(\cdot)$ and subsequently solve the resulting problem using the proposed solution methods.

We further consider the special case of the newsvendor problem [51] where the cost function and feasible set are given by

$$c(z;y) = p(y-z)^+ + q(z-y)^+, \; \mathcal{Z} = \{z \,|\, 0 \leq z \leq 1\},$$

respectively, and p, q denote the respective costs of under/over-estimating y. In the context of electricity markets, the newsvendor problem can be used to model the problem of offering renewable production in a day-ahead electricity market with a dual-price balancing mechanism, assuming offers are normalized by the nominal capacity. In this case, p, q corresponds to the absolute values of the (expected) upward and downward unit regulation costs, respectively.

Given a set of associated features $\mathbf{x}$, we can derive a feature-driven policy for the newsvendor problem by replacing z with $\mathbf{w}^\intercal\mathbf{x}$. Observe that the cost function of the newsvendor problem is equivalent to a quantile loss (4.2) with $\tau = \frac{p}{p+q}$. Therefore, we can further robustify this feature-driven policy against missing features at test time and directly apply the proposed solution methods.

4.4 Energy Forecasting with Missing Data

In this section, we present the experimental setup and list the input data for several energy forecasting applications (in Subsection 4.4.1) and discuss the numeral results (in Subsection 4.4.2).

4.4.1 Problem Description, Experimental Setup, and Input Data

We examine four prevalent energy forecasting applications, namely *(i)* electricity price, *(ii)* load, *(iii)* wind production, and *(iv)* solar production, forecasting in a day-ahead horizon. We assume that data arrive in batches once per day and our objective is to generate forecasts 12 to 36 hours ahead. This setting is typical in applications related to electricity market

participation and operational management in power systems, such as clearing the day-ahead market.

For the numerical experiments, we first select a set of features that lead to good performance in a linear regression model following known best practices. We then train several benchmarks with the same set of features, including both linear regression models and machine learning models with randomization-based training (e.g., Random Forest),[4] which are known to perform well in energy forecasting applications. We compare their out-of-sample performance under feature deletion to the proposed FDRR and retraining without the missing features. Evidently, our goal is not to search for improved forecast accuracy, but rather for resilient energy forecasting, i.e., to examine the robustness of the models.

For point forecasting, we test the following models:

- `LS`: an LS regression.
- `LAD`: an LAD regression.
- `LS-`$\ell_1\backslash\ell_2$: an LS regression with ℓ_1 (lasso) or ℓ_2 (ridge) regularization penalties.
- `RF`: a Random Forest model.
- `RETRAIN` [114]: an LAD regression is retrained for each combination of missing features, in total $\sum_{k=0}^{|\mathcal{J}|} \binom{|\mathcal{J}|}{k}$ times. To facilitate comparisons with the proposed approach, we use LAD instead of LS models to derive equivalent performance when Γ is 0 or $|\mathcal{J}|$.
- `FDRR`(Γ): a robust regression with ℓ_1 loss, and robustness budget Γ.

For probabilistic forecasting, we test the following models:

- `QR`: a quantile regression.
- `QR-`ℓ_1: a quantile regression with ℓ_1 regularization.
- `QRF`: a QRF [70] model, a generalization of Random Forests.
- `FDRR`(Γ): a robust regression with quantile loss, and robustness budget Γ.

For the models that cannot handle missing values directly, i.e., `LS`-type, `LAD`, `RF`, `QR`-type, and `QRF`, we follow the impute-then-regress approach with mean imputation, setting missing features at their in-sample mean. We purposefully choose mean imputation as a simple method that is suitable for an operational setting, [5] thus avoiding complicated and

[4] We opt for tree-based ensembles over other machine learning models (e.g., neural networks) as they showcase exceptionally good performance in regression settings with minimal tuning effort, which makes them ideal benchmarks [141].

[5] In practice, missing data might be replaced by correlated features (which may have been removed during feature selection), if such are available, e.g., data from nearby locations. Practitioners may also apply imputation methods that rely on their experience, whose performance is assessed empirically for a specific forecasting application.

computationally costly methods, which may not add in terms of predictive performance — see e.g., [113] for a discussion in a similar context with missing data. For LS-$\ell_1\backslash\ell_2$ and RF, we use 5-fold cross-validation on the training data for hyperparameter tuning. We select the hyperparameters with the lowest cross-validation error via grid search, and we retrain each model using the full train set. The same hyperparameter values are subsequently used in the probabilistic case for QR-ℓ_1 and QRF, respectively. For FDRR(Γ), missing values are set to zero, and a different model is trained for each value of Γ. To ease the notation, FDRR refers to the group of models trained over all Γ. Clearly, as the number of missing features is known prior to derive out-of-sample forecasts, we use FDRR with Γ set at the exact number of missing features. By definition, FDRR(0) is equivalent to an LAD model. In addition, FDRR and RETRAIN are equivalent for $\Gamma = 0$ and $\Gamma = |\mathcal{J}|$. In all cases, data are scaled between $[0, 1]$ prior to training. Lastly, to derive probabilistic forecasts a different model is trained per quantile τ in all cases except for QRF.

To evaluate performance we use standard error metrics. For point forecasting, we use the MAE for electricity price and wind/solar production (both normalized with respect to nominal capacity), and the Mean Absolute Percentage Error (MAPE) for load. For probabilistic forecasting, we use the average pinball loss on 9 equally spaced quantiles, i.e., $\tau \in \{0.1, \ldots, 0.9\}$.

Table 4.1: Overview of the data sets.

Data set (# series)	Source	n	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{J}\rvert$
Electricity Prices (1)	[69]	13140	9	5
Load (21)	[142]	16200	625	4×111
Wind (10)	[143]	8807	13	2×4
Solar (3)	[143]	8784	13	12

Regarding the input data, Table 4.1 provides an overview of the selected data sets. For each energy forecasting application, it shows the number of series, the source, the training sample size, n, and the sizes of the sets $\mathcal{P}$ and $\mathcal{J}$. Note that the bias (intercept) term is included in $\mathcal{P}$ and cannot be deleted. Further, all cases involve features that capture seasonality and cannot be deleted. Thus, when $\Gamma = |\mathcal{J}|$, FDRR leads to a model that captures the seasonal component of each series.

- **Electricity prices:** We use hourly data from the French electricity market, spanning the period 2017-2019, with a 50/50 training/test split. Features include calendar variables (cannot be deleted), historical price lags, and published data from the system operator, namely net load forecast (demand minus renewable production) and system margin (ratio of net load and available thermal generation). For historical lags, we examine the PACF and select lags that are significant at the 5% level.

- **Load:** We use data from GEFCom 2012 [142], comprising 4.5 years of hourly load and temperature data from a US utility with 21 zones. Following [118, 119], 3 full years of data are used with a 75/25 training/test split. We construct the input feature vector according to the vanilla model [137], which includes a linear trend, calendar variables (one-hot encoded), polynomial terms of temperature, and interaction terms of the above, with a total of 292 features. We consider 4 distinct groups of features based on temperatures from different stations and examine performance under group deletion; this leads to 625 features in total and 111 features per group. Clearly, the subset of features that cannot be deleted (trend and calendar variables) is included only once. The results presented concern zone 21 (aggregate demand) using temperatures from stations 1-3 and a fictitious station with the average temperature across all stations.

- **Wind production:** We use data from GEFCom2014 [143], comprising 2 years of hourly production data from 10 wind farms, and apply a 50/50 training/test split. Following [120], the selected features include wind speed forecasts, with quadratic and cubic terms, wind direction forecasts (both at 10m and 100m), and Fourier terms to model the diurnal patterns (these cannot be deleted). Forecasts of both wind speed and direction are derived from forecasts of the U- and V-speed components for each height level; thus, if either is missing, all derivative features will be missing. We consider two groups of features that include wind speed and wind direction at 10m and 100m and assume that these can be missing independently. The results presented concern zone 1 of the data set.

- **Solar production:** We use data from GEFCom2014 [143], comprising 2 years of hourly production data from 3 PV plants located in Australia and 12 NWP variables, including precipitation, solar radiation, and temperature — see [143] for detail, and apply a 50/50 training/test split. We train a separate model for each hour of the day (except for `RF`, `QRF`) using the respective NWP model forecasts as input features, and assume that each NWP variable could be missing independently. Only hours with non-zero solar radiation are considered. The results presented concern zone 1 of the data set.

4.4.2 Numerical Results

In this section, we evaluate the FDRR solution methods, we compare FDRR with various benchmarks, and we perform a sensitivity analysis with respect to the number of observations with missing features. All FDRR solutions are solved with GUROBI using the Python API.

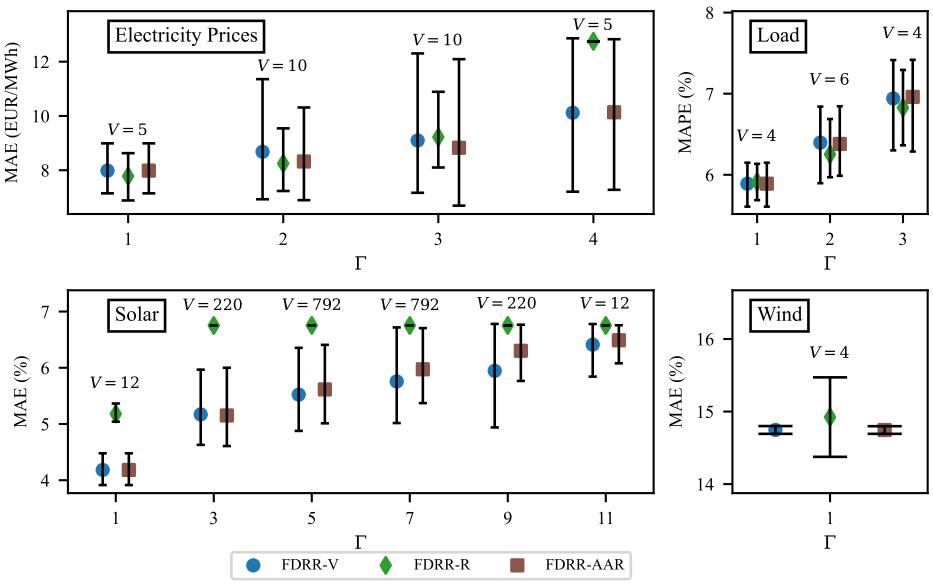


Figure 4.1: Average point forecasting error for all combinations of missing features. Bars indicate the range and V indicates the number of vertices per Γ.

Evaluation of FDRR Solution Methods

In this subsection, we assess the solution methods presented in Section 4.3, namely `FDRR-V`, `FDRR-R`, and `FDRR-AAR`, by iterating over all eligible combinations of missing features and deleting the respective feature observations from the test set.

Fig. 4.1 plots the average value (per Γ) and range of the point forecast error metrics, for each solution method, in the four energy forecasting applications. Note that for each value of Γ, we evaluate the methods for the same number of features missing at test time. To avoid cluttering, we only show the odd (and omit the even) values of Γ in the solar production forecasting plot. Unsurprisingly, we observe that the accuracy for each solution method decreases on average as Γ increases, i.e., as more features are missing. Recall that for $\Gamma = 0$, FDRR is a standard LAD, whereas for $\Gamma = |\mathcal{J}|$ all features in $\mathcal{J}$ are ignored, i.e., coefficients are set to zero; hence the three methods are equivalent in these cases (not shown in the plots).

The results in Fig. 4.1 indicate a similar performance on average for the three methods, with the exception of `FDRR-R` in electricity price forecasting — see top for $\Gamma = 4$ — and solar production forecasting — see bottom. As the number of eligible combinations increases, `FDRR-R` becomes overly conservative, setting all coefficients in $\mathcal{J}$ to zero, which in turn decreases the accuracy. For example, in solar production forecasting, `FDRR(3)-R` becomes equivalent to `FDRR(`$|\mathcal{J}|$`)-R`, which explains the plateau as Γ increases further. Notably, `FDRR-V` and `FDRR-AAR` provide similar performance in terms of average value and range in all applications. Overall, `FDRR-V` ranks higher in solar production forecasting (in about 90% of the combinations) but the differences are very small. `FDRR-AAR` yields slightly better

results compared to FDRR-V, in electricity price and load forecasting, whereas the results are the same in wind production forecasting.

We further evaluate the three solution methods in terms of computational cost, by comparing the required CPU time on an Intel Core i7 at 2.7 GHz with 16GB of RAM, using default solver settings. Our results indicate that when the number of vertices V is relatively small, all methods incur a similar cost. However, as V increases, FDRR-V incurs a computational cost that is several orders of magnitude larger than the other methods. For example, in solar production forecasting, for $\Gamma = 6$, the CPU time ranges from around 200 to over 27×10^3 seconds for FDRR(6)-V, whereas the worst case is less than 1 second and 3.5 seconds, for FDRR(6)-R and FDRR(6)-AAR, respectively. Clearly, FDRR-V incurs a much higher computational cost, which renders this method at least impractical, even for a modest number of features.

We also evaluated the performance on probabilistic forecasts, by repeating the above experiment and training a separate model for each quantile. The obtained results and remarks were very similar to the point forecasts. Pinball loss values increased with Γ, FDRR-R yielded high pinball loss values, similarly to the errors in Fig. 4.1, whereas FDRR-V and FDRR-AAR yielded quite similar performance.

Henceforth, we shall further consider only FDRR-AAR, which stands out as the best FDRR representative with good out-of-sample performance and low computational cost.

Comparison of FDRR with Benchmark Models

In this subsection, we compare FDRR with the benchmark models presented in Section 4.3. For all applications, we iterate over each day of the test set, sample a subset of features, and delete it, repeating the process 10 times.

Fig. 4.2 presents the average error metrics for point forecasting as a function of the number of missing features. In the nominal case, i.e., without missing features, performance is on par with previous works. Specifically, for each application, the best-performing model is: LAD, for electricity price forecasting, with MAE 6.79 EUR/MWh; LS-ℓ_2, for load forecasting, with MAPE 5.07%; LAD, for wind production forecasting, with MAE 13.55%, and LS, LS-ℓ_2, for solar production forecasting, with MAE 6.47%.

Overall, RETRAIN yields the best results in terms of accuracy when features are missing, followed by FDRR, which is clearly the second best. The relative average (maximum) error increase of FDRR compared to RETRAIN is 4.7% (10%) for electricity price, 1.6% (4%) for load, 0.4% (1.7%) for wind production, and 21% (38%) for solar production forecasting. The underlying trend suggests that the gap between FDRR and RETRAIN increases as the number of eligible combinations increases, with its worst case observed for solar production forecasting with 6 missing features, i.e., $\binom{12}{6} = 924$ combinations. FDRR outperforms impute-then-regress benchmarks, namely LS-type, LAD, and RF, in almost all cases, with an average error reduction of 2% for electricity price, 37% for load, 9% for wind production, and 5% for

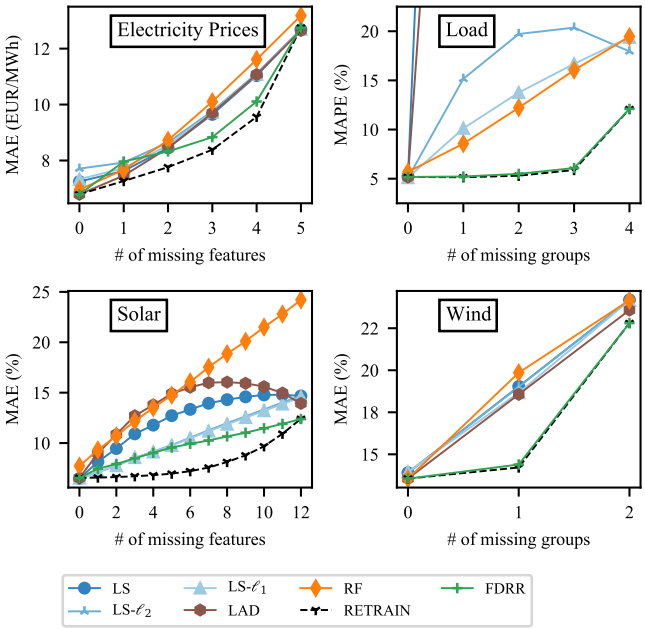
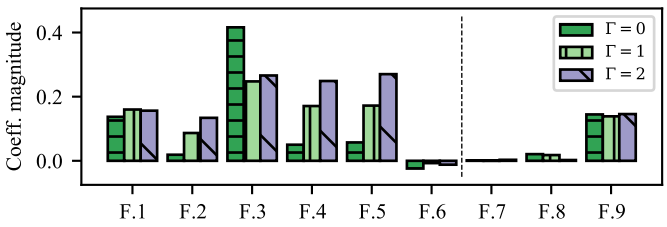


Figure 4.2: Point forecasting error metrics versus the number of missing features.

solar production forecasting. A few exceptions appear, although the differences are small — see top left plot for $\Gamma = 1$ (7% worse than `LAD`) and bottom left plot for $\Gamma = \{2, 3\}$ (5% worse than `LS`-ℓ_2).

Taking a closer look at the impute-then-regress benchmarks, we observe that `LS` and `LAD` exhibit a similar performance, in all applications. Note that for load forecasting (top right plot), although both `LS` and `LAD` perform on par with [118] in the nominal case, they suffer from bad conditioning, which leads to very large coefficients, and, in turn, to bad performance when features are missing (not shown in the plot). The regularized models `LS`-ℓ_1/ℓ_2, in general, improve the performance of the `LS` model — see, e.g., `LS`-ℓ_1/ℓ_2 for load (top right) and solar production (bottom left) forecasting. Lastly, `RF` exhibits the worst performance on average amongst the benchmarks, with the exception of the load forecasting case.



Features in $\mathcal{J}$		Features in $\mathcal{C}$	
F.1: Net Load Forecast	F.3: DA Price_24	F.6: Day	F.8: Month
F.2: System Margin	F.4: DA Price_144	F.7: Hour	F.9: Bias
	F.5: DA Price_168		

Figure 4.3: `FDRR`(Γ) coefficients for point forecasting of electricity prices.

To gain further insight, we focus on point forecasting of electricity prices and examine the effect of Γ. Fig. 4.3 presents the learned coefficients for $\Gamma = \{0, 1, 2\}$. Considering `FDRR`(0), i.e., `LAD`, the plot suggests that the price at lag 24 (DA Price_24 or F.3), i.e., the same hour of the previous day, is the most important feature, followed by the Net Load Forecast (F.1); therefore, if any of them is missing, the impact on performance is expected to be significant. On the other hand, the coefficients for prices at lag 144 (F.4), and lag 168 (F.5) are small, therefore their deletion has a smaller impact. Intuitively, F.3, F.4, and F.5 carry similar information pertaining to the autoregressive and seasonal nature of electricity prices. For $\Gamma = 0$, these three coefficients vary significantly, with a standard deviation of approximately 17%. For $\Gamma = 1$ we observe that the values of the coefficients come closer, and their standard deviation decreases to 3.5%, while for $\Gamma = 2$ their standard deviation further decreases to 0.09%. Effectively, `FDRR`(Γ) hedges against feature uncertainty by assigning similar coefficients to these features, which, in turn, mitigates the adverse effect of deleting F.3 from the test set. Moreover, we observe that the total weight of the coefficients increases with Γ to compensate for the larger number of features set to zero during training. Similar results are also observed for the other applications but omitted due to space limitations. For solar production, e.g., `FDRR`(1) hedges against the deletion of the surface solar radiation down forecast, which is arguably the most important feature.

We further examine performance for probabilistic forecasting and illustrate in Fig. 4.4 the average pinball loss for all applications. Note that we do not examine `RETRAIN` in this case, as applying it for each quantile becomes prohibitive. Indeed, the results closely resemble the ones presented in Fig. 4.2. The ranking of the models is generally maintained, with `FDRR` outperforming the benchmarks in all cases except for electricity price forecasting for $\Gamma = 1$ (7% worse than `QRF`), with an average pinball loss reduction of 5% for electricity price, 46% for load, 15% for wind production, and 21% for solar production forecasting. Moreover, as the number of missing features increases, the pinball loss increases in a qualitatively similar fashion to the respective error metrics for point forecasts. Lastly, as Γ increases, the values of the coefficients for all quantiles come closer — see, e.g., Fig. 4.5 for an illustration of probabilistic forecasting of electricity prices, for $\Gamma = \{0, 1, 2\}$.

Sensitivity Analysis

In this subsection, we perform sensitivity analysis with respect to the number of observations with missing features. Specifically, we sample a percentage of test observations that have missing features, we draw the number of missing features for each observation from a uniform distribution, and we subsequently sample the feature subset that is missing.

Table 4.2 presents the average point forecasting errors over 10 runs. The parentheses indicate the difference from the lowest nominal error, which is used to measure performance degradation. The best model is underlined in bold and the second best is in bold. As expected, `RETRAIN` leads to the smallest error when features are missing and is also the

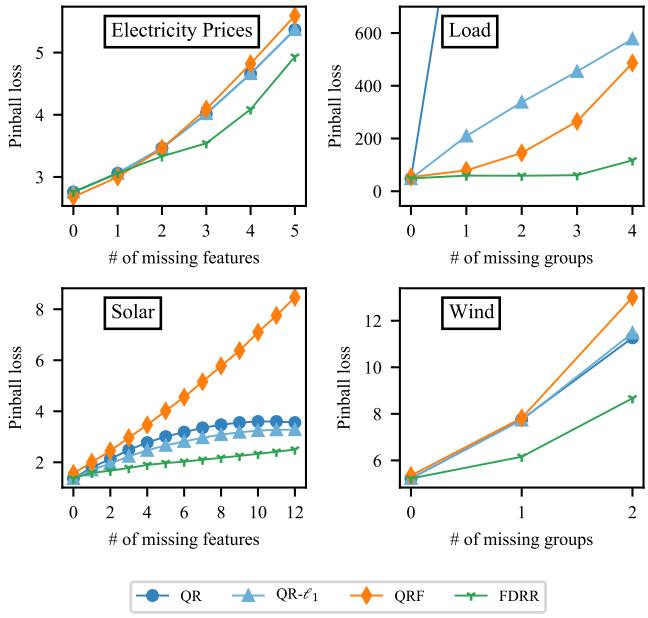


Figure 4.4: Pinball loss versus the number of missing features.

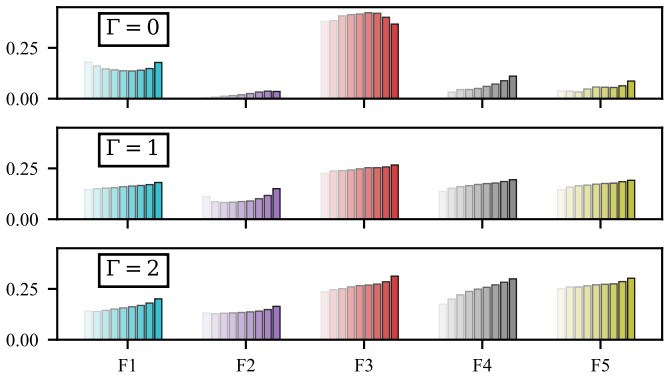


Figure 4.5: `FDRR`(Γ) coefficients for probabilistic forecasting of electricity prices. Higher transparency indicates lower quantiles (a 10% step is considered).

most consistent, i.e., it has the smallest degradation. `FDRR` typically ranks second both in terms of expected error and performance degradation, with generally small differences from `RETRAIN` (with the exception of solar production forecasting, where the performance degradation of `FDRR` is about twice higher compared to `RETRAIN`). Compared to impute-then-regress benchmarks, `FDRR` leads to both smaller error and smaller degradation in all cases except for the lower percentages in solar production forecasting, where it is worse than `LS-`ℓ_2 but only for up to 0.04%. Further, the relative improvement of `FDRR` over the benchmarks increases with the percentage of observations with missing features. Considering only impute-then-regress benchmarks, all models exhibit similar performance for electricity price and wind production forecasting, whereas `LS-`ℓ_1 and `LS-`ℓ_2 are significantly better than the rest for load and solar production forecasting.

We further investigate how FDRR performs with an approximation of the quadratic

loss function for solar production forecasting, which is the only application where `LS` ranks first without missing features. We use a piecewise linear function with 20 equally spaced breakpoints within $[-1, 1]$ — recall that the production is normalized between $[0, 1]$ — to approximate the quadratic loss and solve the robust problem using the affinely adjustable reformulation. Results are shown in the last row of Table 4.2 (`FDRR-PWL`). Without missing features, `FDRR-PWL` and `LS` have the same error, indicating that the piecewise linearization approximates the quadratic loss well. However, when features are missing, `FDRR-PWL` significantly outperforms `LS` (similarly to the way `FDRR` outperforms `LAD`). Furthermore, we note that for the lowest percentage (5%) of observations with missing features, where `LS` performs better than `LAD`, `FDRR-PWL` slightly outperforms `FDRR`.

4.5 Additional Numerical Experiments

In this section, we complement our work with additional numerical experiments from two relevant applications that are affected by missing data. Specifically, we consider the case of dealing with missing data in an integrated forecasting-optimization framework, using the problem of short-term trading of renewable energy production as a guiding example (in Subsection 4.5.1). Next, we examine the problem of forecasting wind power production in a very short-term, intra-hour horizon (in Subsection 4.5.2).

4.5.1 Feature-driven Trading of Renewable Energy Production

Problem Description, Experimental Setup, and Input Data

We consider a producer managing an aggregation of renewable plants participating in a day-ahead electricity market. The producer submits an energy offer for each clearing period and incurs a financial penalty if the realized production deviates from the submitted offer. We assume that the producer is a price-taker and the balancing market operates with a dual-price mechanism—a detailed description of this problem is provided in Chapter 2.4.2. Similar to the day-ahead energy forecasting problem, the forecast horizon is set at 12 to 36 hours ahead, and data are assumed to arrive in batches once per day.

Following our previous experimental setup, we first select a set of features that lead to a good trading performance in a feature-driven setting, i.e., the case where the producer leverages contextual information to directly forecast the trading decisions. Next, we train several benchmarks with the same set of features. Specifically, we evaluate the following approaches:

- `SAA`: the sample average approximation solution that does not consider any features—see Chapter 2.2.1 for details.
- `FeatD`: a feature-driven model that directly predicts the trading decisions with a linear decision rule approach [51].

Table 4.2: Point forecasting error versus percentage (%) of observations with missing features.

% of obs.		0 %	5 %	10 %	25%	50 %
El. Prices	LS	7.25 (0.46)	7.39 (0.60)	7.52 (0.73)	7.91 (1.12)	8.57 (1.78)
	LS-ℓ_2	7.71 (0.92)	7.83 (1.04)	7.95 (1.16)	8.29 (1.50)	8.87 (2.08)
	LS-ℓ_1	7.33 (0.54)	7.47 (0.68)	7.60 (0.81)	7.99 (1.19)	8.65 (1.86)
	LAD	**6.79 (0.00)**	6.95 (0.16)	7.10 (0.31)	7.56 (0.77)	8.33 (1.54)
	RF	**6.90 (0.10)**	7.07 (0.28)	7.23 (0.44)	7.73 (0.94)	8.58 (1.79)
	RETRAIN	**6.79 (0.00)**	**6.92 (0.12)**	**7.03 (0.24)**	**7.38 (0.59)**	**7.97 (1.18)**
	FDRR	**6.79 (0.00)**	**6.94 (0.15)**	**7.08 (0.28)**	**7.48 (0.69)**	**8.20 (1.41)**
Load	LS	5.22 (0.14)	13.65 (8.58)	22.35 (17.28)	46.87 (41.79)	89.07 (84.0)
	LS-ℓ_2	**5.07 (0.00)**	5.74 (0.67)	6.38 (1.31)	8.39 (3.32)	11.69 (6.62)
	LS-ℓ_1	**5.09 (0.02)**	5.60 (0.53)	6.10 (1.03)	7.58 (2.51)	10.03 (4.96)
	LAD	5.18 (0.10)	10.60 (5.53)	15.90 (10.83)	31.58 (26.51)	56.79 (51.72)
	RF	5.72 (0.65)	6.13 (1.06)	6.55 (1.48)	7.81 (2.74)	9.88 (4.81)
	RETRAIN	5.18 (0.10)	**5.27 (0.20)**	**5.38 (0.31)**	**5.66 (0.58)**	**6.13 (1.06)**
	FDRR	5.18 (0.10)	**5.28 (0.21)**	**5.39 (0.31)**	**5.69 (0.62)**	**6.18 (1.11)**
Wind	LS	13.90 (0.36)	14.29 (0.75)	14.65 (1.11)	15.85 (2.31)	17.78 (4.24)
	LS-ℓ_2	13.90 (0.36)	14.29 (0.75)	14.65 (1.11)	15.85 (2.31)	17.78 (4.24)
	LS-ℓ_1	13.95 (0.41)	14.32 (0.79)	14.67 (1.14)	15.83 (2.29)	17.71 (4.18)
	LAD	**13.55 (0.00)**	13.92 (0.39)	14.29 (0.75)	15.46 (1.92)	17.36 (3.82)
	RF	**13.56 (0.01)**	13.95 (0.41)	14.34 (0.80)	15.64 (2.11)	17.66 (4.12)
	RETRAIN	**13.55 (0.00)**	**13.84 (0.30)**	**14.06 (0.52)**	**14.78 (1.24)**	**16.09 (2.55)**
	FDRR	**13.55 (0.00)**	**13.85 (0.31)**	**14.07 (0.53)**	**14.80 (1.26)**	**16.15 (2.61)**
Solar	LS	**6.47 (0.00)**	6.79 (0.32)	7.10 (0.63)	8.04 (1.57)	9.65 (3.18)
	LS-ℓ_2	**6.47 (0.00)**	6.71 (0.23)	**6.92 (0.45)**	7.58 (1.11)	8.73 (2.26)
	LS-ℓ_1	**6.51 (0.04)**	6.74 (0.27)	6.95 (0.48)	7.58 (1.11)	8.70 (2.23)
	LAD	6.54 (0.07)	6.91 (0.44)	7.29 (0.82)	8.42 (1.95)	10.35 (3.88)
	RF	7.71 (1.24)	8.20 (1.72)	8.62 (2.15)	10.03 (3.56)	12.38 (5.91)
	RETRAIN	6.54 (0.07)	**6.62 (0.15)**	**6.71 (0.24)**	**6.94 (0.47)**	**7.37 (0.90)**
	FDRR	6.54 (0.07)	6.74 (0.27)	6.95 (0.48)	**7.53 (1.06)**	**8.51 (2.04)**
	FDRR-PWL	**6.47 (0.00)**	**6.69 (0.22)**	6.94 (0.47)	7.64 (1.17)	8.83 (2.36)

- `RF`: a Random Forest model that approximates the distribution of renewable production, following the method described in Chapter 2.3.
- `FDRR`(Γ): a robust version of `FeatD` with robustness budget Γ, trained with the affinely adjustable reformulation method.

Similarly to the previous experiments, for the models that cannot handle missing values directly, i.e., `FeatD` and `RF`, we follow the impute-then-regress approach with mean imputation. For `FDRR`(Γ), missing values are set to zero, and a different model is trained for each value of Γ. Note that when $\Gamma = 0$, `FDRR` and `FeatD` are equivalent; further, `SAA` is not affected by missing data as it does not include any features.

To evaluate trading performance, we estimate the mean imbalance cost (equivalently, trading cost) normalized per the nominal plant capacity, measured in EUR/MWh. Recall that under a dual-price market, the imbalance cost is always non-negative and a perfect foresight model leads to zero imbalance costs. The unit regulation costs p, q are also stochastic and need to be forecast. After preliminary experimentation, we found that the best trading performance is obtained by using the in-sample mean of the regulation costs as the out-of-sample forecast.

Regarding input data, we consider an aggregation of 3 WPPs and 1 PV plant, with a total capacity of 49 MW (16% PV share), respectively located in northern and southern France, same as in Chapter 2.5.1. The selected features include NWP model forecasts for each plant location, namely wind speed, wind direction, temperature, cloud coverage, and solar radiation forecasts, which leads to a total of 4 groups of NWP variables. For each group of variables, we further include quadratic and cubic terms of wind speed; we also include Fourier terms to model the diurnal patterns, which cannot be deleted. Each group of NWP variables is assumed to go missing independently.

Results

Fig. 4.6 presents the mean trading cost as a function of the number of missing feature groups. In the nominal case, i.e., without missing features, `RF` has the lowest mean imbalance cost

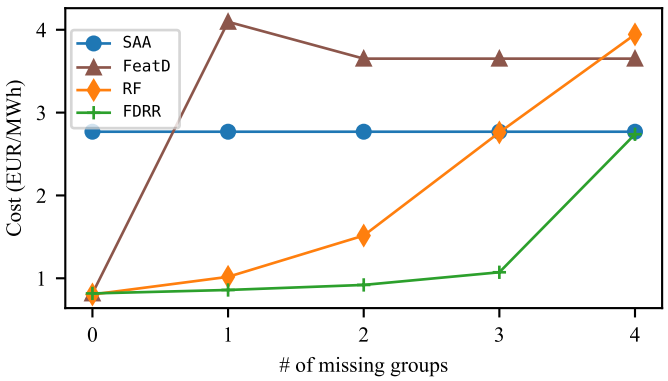


Figure 4.6: Trading cost (EUR/MWh) versus the number of missing features.

Table 4.3: Trading cost versus percentage (%) of observations with missing features.

% of obs.	0 %	5 %	10 %	25 %	50 %
SAA	2.77 (1.96)	2.77 (1.96)	2.77 (1.96)	2.77 (1.96)	2.77 (1.96)
FeatD	0.82 (0.01)	0.94 (0.13)	1.04 (0.24)	1.41 (0.60)	2.01 (1.20)
RF	**0.80 (0.00)**	**0.86 (0.06)**	**0.93 (0.12)**	**1.10 (0.30)**	1.39 (0.58)
RETRAIN	**0.82 (0.01)**	**0.84 (0.03)**	**0.86 (0.05)**	**0.92 (0.12)**	**1.03 (0.23)**
FDRR	**0.82 (0.01)**	0.88 (0.08)	0.94 (0.14)	1.12 (0.31)	**1.34 (0.54)**

(0.80 EUR/MWh), closely followed by `FeatD` and `FDRR` (0.82 EUR/MWh). As expected, `SAA` is the worst-performing model with a mean imbalance cost of 2.77 EUR/MWh. When features are missing, `FDRR` always leads to the lowest cost, with an average cost reduction of 29.28% compared to the second-best model in each case. Regarding the rest of the methods, we observe that `RF` is fairly robust to a small number of missing features, contrary to `FeatD` which significantly worsens when a single feature group is missing. Moreover, `SAA` improves upon `FeatD` and `RF` when the number of missing groups grows large; however, it remains worse than `FDRR` even when $\Gamma = 4$. Overall, the trading performance closely resembles the accuracy results obtained in the energy forecasting case study.

We further perform a sensitivity analysis with respect to the percentage of observations with missing data. Similarly to the energy forecasting case, we sample a percentage of test observations that have missing features, we draw the number of missing features for each observation from a uniform distribution, and we subsequently sample the feature subset that is missing.

Table 4.3 presents the average trading cost over 10 runs, with the parentheses indicating the difference from the lowest nominal error. The best model is underlined in bold and the second best is in bold. Interestingly, `RF` leads to the smallest trading cost for up to 25% of observations with missing features, while `FDRR` ranks second with small differences. When the percentage of observations with missing features reaches 50% `FDRR` starts to outperform `RF`. Conversely, `FeatD` performs notably worse compared to `FDRR` even for small percentages, while `SAA` always ranks worse with significantly higher trading cost in all examined cases.

4.5.2 Intra-hour Wind Production Forecasting with Missing Data

Problem Description, Experimental Setup, and Input Data

We consider the problem of generating point forecasts for the production of a wind power farm in a 30-minute ahead horizon, assuming that new data arrives every 30 minutes. The producer uses spatiotemporal production data, namely previous production values from their own as well as adjacent wind farms to improve forecasting performance. Typically, autoregressive models with regularization achieve state-of-the-art performance in this forecasting task [144].

Following our previous experimental setup, we first select a set of features that lead

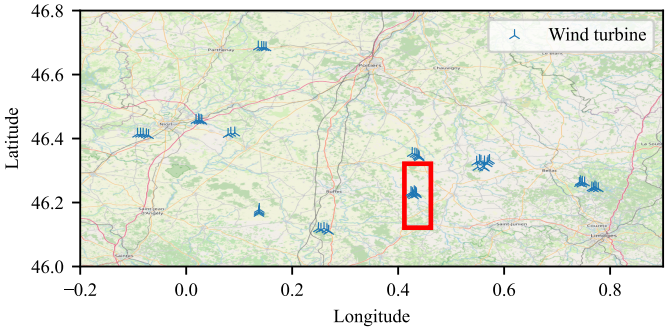


Figure 4.7: Map of the wind power turbines. The red square indicates the target wind farm.

to good forecasting performance. Next, we train several benchmarks with the same set of features. Specifically, we evaluate the following:

- `PERS`: a persistence forecasting model using the current value as forecast.
- `CLIM`: a climatology forecasting model using the in-sample mean as forecast.
- `LS`: an autoregressive LS model with production lags from all the farms.
- $\texttt{LS}_{\ell_1}$: an autoregressive LS model with additional ℓ_1 (lasso) regularization penalty.
- `LAD`: an autoregressive LAD model with production lags from all farms.
- `FDRR`(Γ): a robust regression with ℓ_1 loss, and robustness budget Γ.

For models that cannot handle missing values directly, i.e., `PERS`, `LS`-based, `LAD`-based, we follow the impute-then-regress approach with imputation by persistence. That is, if the current production value of a wind farm is missing, we replace it with its last known value. For `FDRR`(Γ), missing values are set to zero, and a different model is trained for each value of Γ. As before, when $\Gamma = 0$, `FDRR` and `LAD` are equivalent. Further, `CLIM` is not affected by missing data, as it is a constant value that depends on the training data set. To evaluate forecast performance we estimate the MAE (%) of normalized capacity.

For input data, we use power measurements from 60 wind power turbines located in mid-west France, with a nominal aggregated capacity of 120 MW, clustered in 13 wind farms. The selected features comprise the last two production lags from all the wind farms (26 in total)[6]. The available data sets span the period from December 2018 to September 2020 with a 30-min resolution. We use 5 months of data for training and tuning, with the last 5 months used for testing the performance. We evaluate performance on forecasting the production of a single wind farm—see Fig. 4.7 for details.

[6]We considered increasing the production lags but it did not lead to improved accuracy in the nominal case.

Generating Blocks of Missing Data

The examined forecasting application considers a constant stream of data, with new production measurements arriving every 30 minutes. Network latency, cyberattacks, or equipment failures can disrupt this stream of data, leading to missing input data that propagates through time [114]. This context significantly differs from the one considered in the previous two case studies, where data are assumed to arrive in batches once per day.

To generate blocks of missing data that propagate through time, we model the missingness mechanism as a Markov Chain and use a transition probability matrix. The matrix comprises two states, namely, the current observation is missing (State 1) or it is not missing (State 0), given by

$$P = \begin{bmatrix} P_{0,0} & P_{0,1} \\ P_{1,0} & P_{1,1} \end{bmatrix},$$

where $P_{i,j}$ indicates the transition probability from the i-th to the j-th state and the row-wise sum is equal to 1. That is, $P_{0,1}$ is the probability of the next value going missing when the current value is available. We assume that the starting state is $P_{0,0}$ and use historical data to estimate the transition probabilities $P_{1,1}$, $P_{1,0}$, while we vary $P_{0,1}$ to examine the sensitivity with respect to the probability of having a failure that leads to missing data.

Results

This section presents results for estimated values $P_{1,1} = 0.95$, $P_{1,0} = 0.05$, which translates to blocks of missing values with an average length of 10 hours. Fig. 4.8 presents the average MAE for point forecasting as a function of transition probability $P_{0,1}$, i.e., the probability of a failure that generates missing data occurring. For the nominal case (no missing data), the best performing models are `LAD` and `FDRR` with MAE 4.99%, followed by $\texttt{LS}_{\ell_1}$ and `LS`, with MAE 5.08% and 5.09% respectively. All of the models that consider features outperform both `PERS`, which has MAE 5.32%, and `CLIM`, which is significantly worse with MAE 24.92% and thus omitted from Fig. 4.8. When data are missing, i.e., as $P_{0,1}$ increases, Fig. 4.8 shows that `FDRR` leads to significantly better performance compared to the rest of the benchmarks.

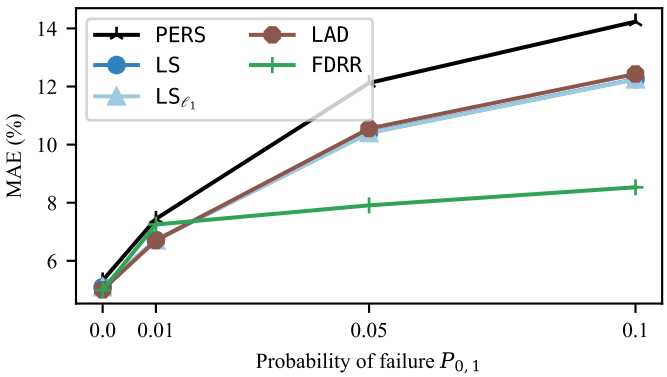


Figure 4.8: MAE (%) versus the transition probability ($P_{0,1}$).

The only exception is for $P_{0,1} = 0.01$, where `FDRR` is 8% worse than the best-performing benchmark, which is `LAD`. Conversely, for $P_{0,1} = 0.05$ and $P_{0,1} = 0.10$ `FDRR` is approximately 32% and 44% better than the best performing benchmark. Overall, the average improvement of `FDRR` against the impute-then-regress benchmarks is around 23% when data are missing.

4.6 Conclusions

This work provided a principled approach to enhance resilience against missing features in energy forecasting via robust optimization. We formulated a robust regression model that is optimally resilient against missing features at test time, considering both point and probabilistic forecasting, and we developed three solution methods for the resulting robust formulation, leading to LP problems. The numerical results indicated that the affinely adjustable reformulation method provides the best trade-off between accuracy and computational cost. In a comprehensive evaluation against several benchmarks coupled with imputation, the proposed approach improved point (probabilistic) forecasting performance in the presence of missing features by 2% (5%) for electricity price, 37% (46%) for load, 9% (15%) for wind production, and 5% (21%) for solar production. Moreover, the proposed approach performed comparable to retraining without the missing features, while avoiding a large number of additional models, and provided resilience in the adverse scenario where the most important feature is missing in an operational setting. A sensitivity analysis with respect to the number of observations with missing features further validated the practical applicability of the proposed approach. Additional case studies further validated the proposed approach. Considering the case of intra-hour wind power production forecasting, the proposed approach improved point forecasting performance by 23% against impute-then-regress benchmarks. Considering the case of a renewable producer directly forecasting the trading decisions for participating in a day-ahead electricity market, the proposed approach improved the average trading performance by approximately 6%. Overall, our results highlight the importance of moving beyond standard accuracy metrics to also consider resilience in adverse scenarios, prior to model deployment.

Future work can focus on extending this approach to non-linear models, such as neural networks. Jointly addressing resilience against missing features and corrupted data due to factors such as cyberattacks, also provides an interesting research direction.

Chapter 5

Data Pooling for Contextual Stochastic Optimization

Résumé en Français

Les méthodes basées sur les données sont très prometteuses pour faire face aux défis associés à la prise de décisions sous incertitude. Cependant, les systèmes complexes du monde réel doivent faire face à un grand nombre d'incertitudes et de problèmes correspondants, et chaque problème peut avoir des données limitées. La rareté des données pose un risque important qui entrave le déploiement de méthodes avancées basées sur les données, nécessitant de nouvelles méthodes qui peuvent pleinement exploiter les données d'instances de problèmes similaires, potentiellement sans lien entre elles. À cette fin, nous proposons deux méthodes pour regrouper les données lorsqu'il s'agit de problèmes d'optimisation stochastique multiples dépendants du contexte. La première consiste à regrouper naïvement des données et à former un modèle global pour dériver des prescriptions sur tous les problèmes, tandis que la seconde s'appuie sur la théorie du transport optimal pour estimer des distributions représentatives sur différents problèmes conditionnés par des informations contextuelles. Une contribution clé est le développement d'un algorithme prescriptif de mise en commun des données pour déterminer quand et combien de données mettre en commun. L'algorithme proposé exploite des outils d'apprentissage d'ensemble pour estimer le coût de décision hors échantillon attendu sans sacrifier les données de formation, et interpole efficacement entre une distribution locale et une distribution groupée. Pour la validation, nous examinons deux applications intégrales liées à l'intégration des sources d'énergie renouvelables dans les systèmes électriques : la prévision de la production d'électricité et la participation sur un marché de l'électricité day-ahead. Les résultats démontrent que la mise en commun des données améliore la prise de décision globale lorsque les données sont rares. Notamment, l'algorithme de mise en commun des données prescriptives proposé surpasse systématiquement les méthodes locales et les méthodes de mise en commun, avec une amélioration des performances attendue de plus de 2% par rapport à l'approche standard découplée.

The work in this chapter appears in [**J4**] that will be submitted soon.

5.1 Introduction

In real-world systems, such as modern power systems, decision-makers deal with a large number of uncertainties, which are also associated with some contextual information. In turn, these uncertainties can create thousands of potentially unrelated stochastic optimization problems. For instance, power producers manage portfolios of thousands of renewable energy sources, such as wind and solar power plants, whose production depends on the weather at each spatial location. Future power systems and smart grids that integrate a large number of heterogeneous assets, such as small-scale renewable energy sources, flexible loads, storage systems, and electric vehicles, further exacerbate this issue.

In this context, decision-makers often encounter a "large-scale, small-data" regime. That is, while the aggregate volume of data across all problems is large, data at an individual (*local*) problem level might be scarce or contaminated, which hinders the deployment of data-driven methods. Fully utilizing the benefits of available data-driven methods, thus necessitates developing effective tools for pooling the available data from different problems.

5.1.1 Aim and Contribution

In this chapter, we propose two methods for data pooling to improve decision performance when dealing with multiple contextually-dependent stochastic optimization problems. The first involves naively pooling all data and training a global model to estimate a conditional distribution of uncertainty as a function of contextual information, which is subsequently used to derive prescriptions across all problems. The second approach is based on Optimal Transport (OT) [145], which is a mathematical framework that studies similarities of probability distributions. Specifically, we use OT to generate representative distributions across different problems conditioned on contextual information. To determine when and how much data to pool, we further develop a prescriptive data pooling algorithm that interpolates between a local and a pooled distribution. The proposed algorithm leverages techniques from ensemble learning, namely the Out-of-Bag (OOB) method [146], to provide an estimation of the expected out-of-sample decision cost without sacrificing training data and avoiding model retraining. We evaluate the effectiveness of the proposed data pooling methods in two critical applications related to the integration of renewable energy sources in power systems: power production forecasting and trading in a day-ahead electricity market. Our results show that data pooling leads to better decisions when data are scarce, with the proposed prescriptive data pooling algorithm consistently leading to better decisions, even as the number of local training observations increases.

5.1.2 Related Work

In recent years, there has been a growing interest in solving stochastic optimization problems where the uncertain parameters are associated with some contextual information. In

Chapter 2, we provide a comprehensive review of related work and discuss several methods that leverage data of joint observations of uncertainty and some associated contextual information. Relevant methods include estimating the probability distribution of uncertainty conditioned on contextual information [20] or directly forecasting the problem decisions [30]. Nonetheless, the majority of relevant work deals with a single problem and the setting of multiple contextually-dependent optimization problems simultaneously remains largely unexplored. In this regard, [147] examines data pooling for multiple stochastic optimization problems without contextual information and shows that it leads to better decisions owing to the so-called instability versus suboptimality trade-off. Intuitively, data pooling is most useful when data are scarce and the respective local solution, i.e., the solution that leverages only local problem data, is unstable. To determine when and how much data to pool across problems, [147] further develops an algorithm based on cross-validation that exploits the structure of the optimization problem, which, nonetheless, does not account for contextual information.

Conversely, in the area of time series forecasting, there is a growing interest in developing global forecasting models. The term *global* forecasting model refers to a single univariate model trained across a large number of time series, while a *local* forecasting model is a univariate model trained for a specific time series. Global forecasting models are considered an effective method of simultaneously reducing modeling effort and enabling cross-learning across tasks. For instance, [148] proposes a global deep learning model for probabilistic demand forecasting, while [149] shows that global models can perform on par with local models for time series forecasting, but may have a lower representational capacity for regression tasks. In power systems applications, global models have been used to forecast the uncertain renewable production of multiple plants or the individual consumption at a household level [150]. For instance, [151] examines centralized and federated learning frameworks to forecast the temperature of thermostatically controlled loads using domain-informed data augmentation. Conversely, [152] proposes a global model for load forecasting in the distribution grid and proposes a clustering-based localization method to improve performance under data heterogeneity. To cold-start the forecasting problem for a residential solar panel without historical data, [153] trains a generic cross-learning model across several series. Nonetheless, the problem of interpolating between a local and global model as a function of the volume of available data and the degree of data heterogeneity has not received much attention.

Our proposed approach to using OT for data pooling also shares similarities with the areas of ensemble learning, model aggregation, and forecast combination, with [154–156] being most closely related to our work. Particularly, [154, 155] leverage OT to combine experts' opinions (e.g., forecasts) of a reference probability distribution, via means of a weighted Wasserstein barycenter. This approach is further extended in [156] to the linear aggregation of point predictions for wind speed by aggregating forecasts in adjacent spatial

locations. Our work differs in several key aspects. First, rather than combining probabilistic forecasts, we are interested in approximating the conditional distribution of multiple, independent uncertain problem parameters. Second, we account for the downstream decision cost when aggregating individual models. Third, we do not have access to the true underlying distribution, i.e., the true conditional marginal, but rather leverage tools from bootstrapping and cross-validation [146, Ch. 8] to estimate the out-of-sample performance. Conversely, the above-mentioned works only consider in-sample performance which might not be a good indicator of out-of-sample performance, especially in a setting of scarce data.

5.1.3 Chapter Outline

The remainder of this chapter is organized as follows. Section 5.2 presents a short background on OT, while Section 5.3 introduces the main problem. Section 5.4 develops two data pooling methods and Section 5.5 develops the prescriptive algorithm that decides when and how much data to pool. Finally, Section 5.6 discusses the numerical results and Section 5.7 concludes and provides directions for future work.

5.2 Preliminaries on Optimal Transport

This section provides preliminaries on OT, namely, introduces the OT problem (in Subection 5.2.1) and the Wasserstein barycenter (in Subection 5.2.2).

5.2.1 Optimal Transport Problem

We consider a histogram $\mathbf{a} \in \Sigma_n$ of n values, where

$$\Sigma_n = \{\mathbf{a} \in \mathbb{R}_+^n \,|\, \mathbf{a}^\intercal \mathbf{1}_n = 1\}$$

is the standard $(n-1)$-dimensional probability simplex and $\mathbf{1}_n$ is an n-size vector of ones. The terms histogram and probability vector are used interchangeably throughout.

A discrete measure with weights $\mathbf{a}$ and locations $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n \in \Xi$ reads

$$\alpha = \sum_{i=1}^{n} \mathrm{a}_i \delta_{\boldsymbol{\xi}_i}, \tag{5.1}$$

where $\delta_{\boldsymbol{\xi}_i}$ is the Dirac delta distribution at position $\boldsymbol{\xi}_i$, intuitively a unit of mass that is concentrated at location $\boldsymbol{\xi}_i$. Such a measure is a probability measure if, additionally, $\mathbf{a} \in \Sigma_n$.

The OT problem seeks to find the best way to transport a given number of goods from a set of sources to a set of destinations, where the cost of transporting each unit of goods from each source to each destination is known. Formally, consider two discrete measures α, β of the form (5.1) with corresponding histograms $\mathbf{a} \in \Sigma_n$, $\mathbf{b} \in \Sigma_m$ and respective support locations $\boldsymbol{\xi}_i, i = 1, \ldots, n$, and $\boldsymbol{\xi}'_j, j = 1, \ldots, m$. Let $\mathbf{C} \in \mathbb{R}_+^{n \times m}$ be a known cost matrix,

where $c_{i,j}$ stores the cost of transporting a unit of goods from $\boldsymbol{\xi}_i$ to $\boldsymbol{\xi}'_j$. Further, let the polytope of admissible couplings between $\mathbf{a}, \mathbf{b}$ be

$$\mathbf{U}(\mathbf{a}, \mathbf{b}) \stackrel{\text{def}}{=} \{\boldsymbol{\Gamma} \in \mathbb{R}_+^{n \times m} \,|\, \boldsymbol{\Gamma}\mathbf{1}_m = \mathbf{a}, \boldsymbol{\Gamma}^\intercal \mathbf{1}_n = \mathbf{b}\}. \tag{5.2}$$

The OT problem between $\mathbf{a}, \mathbf{b}$ is given by

$$W(\mathbf{a}, \mathbf{b}) \stackrel{\text{def}}{=} \min_{\boldsymbol{\Gamma} \in \mathbf{U}(\mathbf{a}, \mathbf{b})} < \boldsymbol{\Gamma}, \mathbf{C} > . \tag{5.3}$$

where $< \boldsymbol{\Gamma}, \mathbf{C} >= \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_{i,j} c_{i,j}$. The decision matrix $\boldsymbol{\Gamma}$ is the so-called transportation plan, with $\gamma_{i,j}$ representing the probability mass transported from the i-th source to the j-th destination, with (5.2) ensuring that the total amount of mass moved satisfies both each source supply and each demand destination and the non-negativity constraints.

If we further assume that $c_{i,j} = \|\boldsymbol{\xi}_i - \boldsymbol{\xi}'_j\|^r$, for some $r \geq 1$, where $\|\cdot\|$ is an arbitrary norm, then the optimal value of (5.3) is equal to the r-Wasserstein distance between measures α, β, raised to the r-th power. The Wasserstein distance is a distance metric between probability distributions that measures the minimum cost of transforming one distribution into the other. The Wasserstein distance has many applications in different fields, such as computer vision, machine learning, and uncertainty quantification in mathematical programming.

The OT problem (5.3) is an LP problem, which can be solved using off-the-shelf solvers. If α, β are defined on the real line, then a closed-form solution also exists [155]. To deal with the computational challenges associated with large-scale problems that arise in machine learning applications, several specialized algorithms have also been developed, such as entropic regularization schemes [157, 158] — for a comprehensive overview of numerical methods for computational optimal transport, see [145].

5.2.2 Wasserstein Barycenter

We further consider S histograms $\{\mathbf{b}_s\}_{s=1}^S$, where $\mathbf{b}_s \in \Sigma_{n_s}$, and our goal is to estimate an "average histogram" over a grid of n fixed support locations. The Wasserstein barycenter [159], i.e., the generalized mean, is the histogram $\mathbf{a} \in \Sigma_n$ that minimizes the weighted sum of the Wasserstein distances from $\{\mathbf{b}_s\}_{s=1}^S$. The Wasserstein barycenter $\mathbf{q}^*$ is given by

$$\mathbf{q}^* = \arg\min_{\mathbf{q} \in \Sigma_n} \sum_{s=1}^{S} \lambda_s W(\mathbf{q}, \mathbf{p}_s), \tag{5.4}$$

and is parameterized by a probability vector of $\boldsymbol{\lambda} \in \Sigma_S$ of known weights, termed *barycentric coordinates*; a typical choice is to set $\lambda_s = \frac{n_s}{\sum_{s=1}^{S} n_s}$. Note that each Wasserstein distance itself denotes a minimization problem. Evidently, problem (5.4) is also an LP problem, although its size is much larger than the OT problem (5.3). The Wasserstein barycenter is a generalization of the Euclidean mean in higher dimensions and can be used to compute a representative distribution for a set of distributions, and has found many applications in clustering, classification, model aggregation [155], and variational data assimilation problems [160]. For measures defined on the real line, the Wasserstein barycenter can be estimated efficiently with a closed-form solution.

5.3 Problem Formulation

In this section, we first revisit the problem of contextual stochastic optimization (in Subsection 5.3.1). Then, we consider a setting of multiple problems each associated with some contextual information (in Subsection 5.3.2), and describe the standard solution approach (in Subsection 5.3.3).

5.3.1 Preliminaries on Contextual Stochastic Optimization

We consider a contextual stochastic optimization, or prescriptive analytics, problem given by

$$\min_{\mathbf{z}\in\mathcal{Z}} \mathbb{E}_{\mathbf{y}}[c(\mathbf{z};\mathbf{y})|\mathbf{x}=\mathbf{x}_0], \tag{5.5}$$

where $\mathbf{y}\in\mathcal{Y}$ denotes the uncertain problem parameters (e.g., renewable production), $\mathbf{x}\in\mathcal{X}$ denotes some associated contextual features (e.g., the weather), $\mathbf{x}_0$ denotes a realization of $\mathbf{x}$, $\mathbf{z}$ denotes the decision variables, $\mathcal{Z}$ denotes the set of feasible solutions, $c(\cdot)$ denotes a convex cost function, and the expectation is taken with respect to the conditional distribution of $\mathbf{y}$ given $\mathbf{x}=\mathbf{x}_0$.

We assume that the uncertain parameter $\mathbf{y}$ is a discrete random variable with finite support denoted by $\mathcal{Y}\overset{\text{def}}{=}\{\tilde{\mathbf{y}}_1,\ldots,\tilde{\mathbf{y}}_K\}$, where K is the number of support locations. For any $\mathbf{x}\in\mathcal{X}$, the true conditional distribution of $\mathbf{y}$ is given by a probability vector $\mathbf{p}(\mathbf{x})\in\Sigma_K$, where Σ_K is the $(K-1)$-dimensional probability simplex. The k-th component of $\mathbf{p}(\mathbf{x})$ is defined as $p_k(\mathbf{x})=\mathbb{P}(\mathbf{y}=\tilde{\mathbf{y}}_k|\mathbf{x})$, i.e., the probability of $\mathbf{y}=\tilde{\mathbf{y}}_k$ conditioned on contextual information $\mathbf{x}$. Thus, problem (5.5) can be equivalently written as

$$\min_{\mathbf{z}\in\mathcal{Z}} \mathbb{E}_{\mathbf{y}}[c(\mathbf{z};\mathbf{y})|\mathbf{x}=\mathbf{x}_0] = \min_{\mathbf{z}\in\mathcal{Z}} \sum_{k=1}^{K} p_k(\mathbf{x}_0)c(\mathbf{z};\tilde{\mathbf{y}}_k). \tag{5.6}$$

In practice, instead of the true probability vector $\mathbf{p}(\mathbf{x}_0)$, we have access to a training data set $\mathcal{D}=\{(\mathbf{y}_i,\mathbf{x}_i)\}_{i=1}^N$ of N observations, which we can use to approximate (5.6) — Chapter 2 reviews different data-driven methods to approximate (5.6). Here, we focus on the case where we use a function to estimate the true conditional distribution $\mathbf{p}(\mathbf{x})$ for all $\mathbf{x}\in\mathcal{X}$, that is, we employ a probabilistic forecasting model. Specifically, assume a hypothesis class $\mathcal{H}$ of functions $f:\mathcal{X}\to\Sigma_K$ that map contextual information $\mathbf{x}$ to the conditional distribution of uncertainty $\mathbf{y}$. Note that since $f(\mathbf{x})\in\Sigma_K$, the output of the learning model needs to satisfy a set of constraints. To keep the notation consistent, we refer to $\widehat{\mathbf{p}}:\mathcal{X}\to\Sigma_K$ as the model trained on available data and to $\widehat{\mathbf{p}}(\mathbf{x})\in\Sigma_K$ as the estimated conditional distribution (probability vector) for any $\mathbf{x}\in\mathcal{X}$.

To measure the prescriptive quality of a model $\widehat{\mathbf{p}}:\mathcal{X}\to\Sigma_K$, we further define a function that measures the excess cost incurred by using $\widehat{\mathbf{p}}$ to approximate a problem of the form of (5.6) compared to the perfect foresight solution. To streamline notation, for any $\mathbf{q}\in\Sigma_K$, we define $\mathbf{z}(\mathbf{q})=\arg\min_{\mathbf{z}\in\mathcal{Z}}\sum_{k=1}^K q_k c(\mathbf{z};\tilde{\mathbf{y}}_k)$. Let

$$D(\widehat{\mathbf{p}}(\mathbf{x}_0),\mathbf{y}_0\,|\,c,\mathcal{Z}) = c(\mathbf{z}(\widehat{\mathbf{p}}(\mathbf{x}_0));\mathbf{y}_0) - c(\mathbf{z}^*;\mathbf{y}_0), \tag{5.7}$$

denote the excess cost incurred using $\widehat{\mathbf{p}}$ estimated with respect to the cost function c and the feasible set $\mathcal{Z}$, where $\mathbf{z}^* = \arg\min_{\mathbf{z}\in\mathcal{Z}} c(\mathbf{z}; \mathbf{y}_0)$. Evidently, the prescriptive cost estimated from D is always non-negative.

Remark 5.1 *In the special case where $c(\mathbf{z}; \mathbf{y}) = \mathbf{y}^\intercal \mathbf{z}$, i.e., we deal with a linear objective function with unknown cost coefficients, then, for any $\mathbf{x} \in \mathcal{X}$,* (5.6) *becomes*

$$\min_{\mathbf{z}\in\mathcal{Z}} \sum_{k=1}^{K} p_k(\mathbf{x}_0)\mathbf{z}^\intercal \tilde{\mathbf{y}}_k = \min_{\mathbf{z}\in\mathcal{Z}} \mathbf{z}^\intercal \mathbb{E}[\mathbf{y}|\mathbf{x} = \mathbf{x}_0].$$

Thus, we can replace $\mathbf{p}(\mathbf{x}_0)$ with the conditional expectation of $\mathbf{y}$ given $\mathbf{x}$ using a deterministic forecasting model.

A variety of methods can be employed to generate probabilistic forecasts, including parametric models, non-parametric models [20], conformal prediction [161], or multi-label classification. The training process of the forecasting model can also incorporate the downstream optimization cost D — see, e.g., the methods proposed [26, 36, 162], and in Chapter 2.3.

Non-parametric machine learning methods For instance, consider the case of non-parametric machine learning models, such as neighbor-based or tree-based models. These models infer a function that assigns weights $\boldsymbol{\omega}(\mathbf{x}) \in \Sigma_N$ to training observations $\mathbf{y}_i$ based on contextual information $\mathbf{x}$. Then, the original problem (5.6) is approximated by

$$\min_{\mathbf{z}\in\mathcal{Z}} \sum_{i=1}^{N} \omega_i(\mathbf{x}_0) c(\mathbf{z}; \mathbf{y}_i). \tag{5.8}$$

A specific example that we revisit throughout this chapter is the case of an ensemble of T decision trees $\{\tau_1, \ldots, \tau_T\}$ grown with the random forest method [56], where $\tau_j : \mathcal{X} \to \{1, \ldots, L_j\}$ is a map that corresponds to a disjoint partition of $\mathcal{X}$ into L_j tree leaves and $\tau_j(\mathbf{x})$ is the leaf identity—see Chapter 2.3 for details. In this case, the respective weights are given by

$$\omega_i(\mathbf{x}_0) = \frac{1}{T} \sum_{j=1}^{T} \frac{\mathbb{I}\left[\tau_j(\mathbf{x}_i) = \tau_j(\mathbf{x}_0)\right]}{\sum_{i'=1}^{N} \mathbb{I}\left[\tau_j(\mathbf{x}_{i'}) = \tau_j(\mathbf{x}_0)\right]}, \tag{5.9}$$

where is $\mathbb{I}[\cdot]$ is the indicator function. Evidently, as $\mathbf{y}$ has finite support, we can count the number of times $\tilde{\mathbf{y}}_k$ appears in $\mathcal{D}$ and aggregate the respective weights $\omega_i(\mathbf{x}_0)$ to equivalently write (5.8) with a probability vector that weighs each support location. That is, the estimated probability of $\mathbf{y} = \tilde{\mathbf{y}}_k$ conditioned on $\mathbf{x} = \mathbf{x}_0$ is given by $\widehat{p}_k(\mathbf{x}) = \sum_{i=1}^{N} \mathbb{I}\left[\mathbf{y}_i = \tilde{\mathbf{y}}_k\right] \omega_i(\mathbf{x})$.

5.3.2 Dealing with Multiple, Contextually-Dependent Problems

In this setting, we are interested in solving a collection of S potentially independent stochastic optimization problems, where each uncertainty is associated with some contextual information, specified by

$$\frac{1}{S} \sum_{s=1}^{S} \min_{\mathbf{z}_s\in\mathcal{Z}_s} \mathbb{E}_{\mathbf{y}_s}[c_s(\mathbf{z}_s; \mathbf{y}_s)|\mathbf{x}_s = \mathbf{x}_{s,0}] = \frac{1}{S} \sum_{s=1}^{S} \min_{\mathbf{z}_s\in\mathcal{Z}_s} \sum_{k=1}^{K_s} p_{s,k}(\mathbf{x}_{s,0}) c_s(\mathbf{z}_s; \tilde{\mathbf{y}}_{s,k}), \tag{5.10}$$

where $\mathbf{y}_s$ represents the uncertain parameters, $\mathcal{Z}_s$ is the set of feasible solutions, $\mathbf{x}_{s,0}$ is a realization of the context $\mathbf{x}_s$, and $\mathbf{p}_s(\mathbf{x}_s) \in \Sigma_{K_s}$ denotes the true conditional distribution of $\mathbf{y}_s$ given $\mathbf{x}_s$. Throughout, subscript s is used to indicate that we are referring to the s-th subproblem[1].

In this work, we are interested in the case where the uncertainty $\mathbf{y}_s$ and the contextual information $\mathbf{x}_s$ represent the same variables across all problems; for example, they may represent a pair of renewable energy production and associated weather forecast observations. Thus, we assume that $\tilde{\mathbf{y}}_{s,k} = \tilde{\mathbf{y}}_k$, $K_s = K$, and $\mathcal{X}_s = \mathcal{X}$. To further simplify the notation, we assume, without loss of generality, that $c_s(\mathbf{z}; \mathbf{y}) = c(\mathbf{z}; \mathbf{y})$ and $\mathcal{Z}_s = \mathcal{Z}$. Thus, problem (5.10) can be equivalently written as

$$\frac{1}{S} \sum_{s=1}^{S} \min_{\mathbf{z}_s \in \mathcal{Z}} \sum_{k=1}^{K} p_{s,k}(\mathbf{x}_{s,0}) c(\mathbf{z}_s; \tilde{\mathbf{y}}_k). \tag{5.11}$$

Note that the true conditional distributions $\mathbf{p}_s(\mathbf{x})$ may differ across problems and are, naturally, unknown. Instead, for each subproblem s, we have access to a local training data set $\mathcal{D}_s = \{(\mathbf{y}_{s,i}, \mathbf{x}_{s,i})\}_{i=1}^{N_s}$ of N_s observations, with subscript s being used to highlight that training observations differ across problems; the same also holds true for the out-of-sample realizations $\mathbf{x}_{s,0}$. Similar to the case of the single problem, our goal is to use the available data sets to approximate (5.11).

5.3.3 The Standard Local Solution Approach

In the absence of coupling constraints or variables across the S subproblems in (5.11), the standard approach would be to decouple them and solve them separately using the local data sets. Consider a probabilistic forecasting model $\widehat{\mathbf{p}}_s : \mathcal{X} \to \Sigma_K$ trained on the local data set $\mathcal{D}_s$. The decoupled solution of (5.11) is then given by

$$\left\{ \min_{\mathbf{z} \in \mathcal{Z}} \sum_{k=1}^{K} \widehat{p}_{s,k}(\mathbf{x}_{s,0}) c(\mathbf{z}; \tilde{\mathbf{y}}_k) \right\}_{s=1,\ldots,S}, \tag{5.12}$$

where $\widehat{\mathbf{p}}_s(\mathbf{x}_{s,0}) \in \Sigma_K$ is an estimated probability vector. We consider (5.12) to be the standard benchmark of solving (5.11) and refer to it as the *local* approach, as it relies solely on the local data set $\mathcal{D}_s$ when solving the s-th subproblem.

However, if the local training data sets are scarce, the learned models may incur a high degree of misspecification and lead to poor out-of-sample performance. Therefore, we investigate whether pooling data across the S subproblems can be beneficial.

5.4 Data Pooling Methods

In this section, we describe different approaches to leverage data across problems to improve prescriptive performance. Specifically, we first describe a method based on naive data pooling (in Subsection 5.4.1), followed by an OT-based method (in Subsection 5.4.2).

[1]For simplicity, we assume that all problems are weighted equally in the objective.

5.4.1 Global Model with Naive Data Pooling

A straightforward approach for data pooling is to combine all local data sets $\{\mathcal{D}_s\}_{s=1}^S$ and train a single, centralized, global probabilistic forecasting model. Let $\mathcal{D}^{\texttt{pool}} = (\mathcal{D}_1, \ldots, \mathcal{D}_S)$ be the concatenation of all data sets, and let $\widehat{\mathbf{p}}^{\texttt{pool}} : \mathcal{X} \to \Sigma_K$ be a global model. Then, problem (5.11) can be approximated by solving S decoupled problems given by

$$\left\{ \min_{\mathbf{z} \in \mathcal{Z}} \sum_{k=1}^{K} \widehat{p}_k^{\texttt{pool}}(\mathbf{x}_{s,0}) c(\mathbf{z}; \tilde{\mathbf{y}}_k) \right\}_{s=1,\ldots,S}, \tag{5.13}$$

i.e., the decoupled problems are solved using a common, global forecasting model.

Revisiting the case of non-parametric machine learning algorithms (5.8), to apply the proposed global approach, we first train a single model using data set $\mathcal{D}^{\texttt{pool}}$. Then, for each problem, we estimate the respective weights $\boldsymbol{\omega}^{\texttt{pool}}(\mathbf{x}_{s,0}) \in \Sigma_{N^{\texttt{pool}}}$, where $N^{\texttt{pool}} = |\mathcal{D}^{\texttt{pool}}|$.

We refer to this approach as a global method with naive data pooling, following the global forecasting terminology [148]. In practice, this approach requires a centralized entity that collects all the data and trains the global model, which may create issues regarding data leakage and raise privacy concerns. Considering a federated learning framework where the global model is trained without sharing data across the S subproblems can ameliorate such privacy concerns.

Besides the concerns about privacy and data leakage, a potential shortcoming associated with the naive data pooling approach is model misspecification due to data heterogeneity. A global model may not generalize well to all subproblems due to differences in the underlying problem structure or the distribution of uncertain parameters. Specifically, concept drift, i.e., the case when the true joint distribution between $\mathbf{y}$ and $\mathbf{x}$ differs across the subproblems, poses a major challenge to training a global model. Therefore, alternative approaches that address these shortcomings may be necessary.

5.4.2 Optimal Transport-based Data Pooling

In this section, we propose an OT-based data pooling method that does not require centralized collection of data. Following the standard local approach described in Section 5.3.2, we assume S local models $\widehat{\mathbf{p}}_s : \mathcal{X} \to \Sigma_K$ that map contextual information to probability vectors $\widehat{\mathbf{p}}_s(\mathbf{x}) \in \Sigma_K$. Our goal is, for each $\mathbf{x} \in \mathcal{X}$, to combine knowledge across the S problems by estimating representative conditional distributions. Let $\mathbf{g} : \mathcal{X} \to \Sigma_K$ be defined as

$$\mathbf{g}(\mathbf{x}) = \arg\min_{\mathbf{q}} \sum_{s=1}^{S} \lambda_s W(\mathbf{q}, \mathbf{p}_s(\mathbf{x})). \tag{5.14}$$

In words, $\mathbf{g}$ is a composite function that, given some context $\mathbf{x}$, aggregates the S local models by evaluating the Wasserstein barycenter of their output, parameterized by coordinates $\boldsymbol{\lambda} \in \Sigma_S$. Then, problem (5.11) can be approximated by solving S decoupled problems given by

$$\left\{ \min_{z \in \mathcal{Z}} \sum_{k=1}^{K} g_k(\mathbf{x}_{s,0}) c(\mathbf{z}; \tilde{\mathbf{y}}_k) \right\}_{s=1,\ldots,S}. \tag{5.15}$$

As in the previous case, all problems leverage the same function (model) to derive conditional distributions. However, unlike the naive data pooling approach, we do not require centralized access to the local training data sets and do not affect model training. Rather, we only require access to the trained local models $\widehat{\mathbf{p}}_s$.

For the case of non-parametric machine learning algorithms (5.8), the OT-based data pooling involves, for each $\mathbf{x} \in \mathcal{X}$, first assigning weights $\boldsymbol{\omega}_s(\mathbf{x}) \in \Sigma_{N_s}$ to historical observations in $\mathcal{D}_s$, transforming them into probability vectors that weight each support location $\tilde{\mathbf{y}}_k$, and then estimating the respective Wasserstein barycenter.

There are some limitations associated with the Wasserstein barycenter approach that warrant discussion. Firstly, the estimation of the Wasserstein barycenter is generally computationally expensive. As the dimension of $\mathbf{x}$ is typically large compared to $\mathbf{y}$, we chose to estimate barycenters of conditional marginals, instead of barycenters of the joint distribution between $\mathbf{y}$ and $\mathbf{x}$, thus decoupling the estimation problem from the size of $\mathbf{x}$. Secondly, this approach assumes that all local models have the same level of expertise and that their respective data sets are equally informative. This assumption may not hold in practice, as some subproblems may have more informative data sets than others, and therefore the respective local models should be given more weight in the barycenter computation; this can be addressed by tuning the barycentric coordinates $\boldsymbol{\lambda}$.

5.5 Prescriptive Data Pooling

In the previous section, we discussed two approaches for pooling data across multiple subproblems: naive data pooling and the Wasserstein barycenter. Here, we propose a prescriptive data pooling algorithm that interpolates between the local and the global approaches based on the expected out-of-sample decision cost of the downstream optimization problem.

First, we introduce a method to estimate the expected prescriptive cost using the OOB method, which sets the foundation for our method (in Subsection 5.5.1). Next, we present our prescriptive data pooling algorithm (in Subsection 5.5.2).

5.5.1 OOB Estimation of the Prescriptive Cost

This section describes how to estimate the out-of-sample prescriptive cost of a trained model building on the OOB error method, which is a technique used in ensemble learning to estimate model performance. The reason for building our proposed approach on the OOB method is twofold. First, it allows us to jointly train and test a model, which is considerably less computationally costly than cross-validation. Second, it leverages the full training data set and does not require a separate validation set, making it advantageous when training data are scarce.

We consider an ensemble model of weak base learners trained using bootstrap aggregation (*bagging*), e.g., a random forest model. That is, during the training process, each base

learner is trained on a new data set created by subsampling with replacement (bootstrapping) from the original training data set. The predictions of the models inferred by the base learners are then aggregated via, e.g., averaging— see [146, Ch. 8] for details. By evaluating predictions on observations not used in the training of a specific model, bagging allows for evaluating the so-called OOB error, which provides an estimate of the out-of-sample prediction error. As the number of training observations increases, the OOB error converges to the leave-one-out cross-validation error [163].

We now describe a novel approach to evaluating the expected out-of-sample decision cost, by adapting the OOB method to a prescriptive context. For simplicity, we consider the case of a single model and drop subscript s. Consider a problem of the form of (5.6) approximated using an ensemble model $\widehat{\mathbf{p}} : \mathcal{X} \to \Sigma_K$ composed of weak base learners, trained either to minimize prediction error or the downstream decision cost. The process is described as follows. For $i = 1, \ldots, N$, we find all the models inferred by the base learners for which the i-th observation was not used for training. These models can be considered a new ensemble model, which we use to derive a conditional distribution $\widehat{\mathbf{p}}_i^{\texttt{OOB}}(\mathbf{x}_i)$. The OOB estimate of the prescriptive cost is the average difference between the incurred decision cost and the perfect foresight solution, given by

$$\frac{1}{N} \sum_{i=1}^{N} D(\widehat{\mathbf{p}}_i^{\texttt{OOB}}(\mathbf{x}_i), \mathbf{y}_i \,|\, c, \mathcal{Z}). \tag{5.16}$$

Notably, the key distinction from the OOB estimation of the prediction error method is that the OOB estimation of the prescriptive cost solves a weighted SAA of a stochastic optimization problem for each OOB observation and measures the incurred decision cost. In contrast, the standard OOB method involves averaging the base learner predictions and measuring the prediction error [2]. The prescriptive OOB method has also potential applications in searching for model hyperparameters that lead to the smallest decision cost, similar to [38].

We next describe in detail how to estimate $\widehat{\mathbf{p}}_i^{\texttt{OOB}}(\mathbf{x})$ for the specific case when $\widehat{\mathbf{p}}$ is a random forest model. Consider a random forest composed of T trees $\{\tau_1, \ldots, \tau_T\}$ that outputs weights $\boldsymbol{\omega}(\mathbf{x}) \in \Sigma_N$ of the form (5.9) for any $\mathbf{x} \in \mathcal{X}$, where τ_j is trained using a bootstrapped version of $\mathcal{D}$. For the i-th training observation, let $\mathcal{T} \subseteq [T]$ be the subset of trees that did not use the i-th observation for training. Further, let $\mathcal{D}' = \mathcal{D} \setminus \{(\mathbf{y}_i, \mathbf{x}_i)\}$ be a surrogate data set that excludes the i-th training observation from the original data set $\mathcal{D}$. For each observation in $\mathcal{D}'$, indexed by subscript ℓ, we estimate weights

$$\omega_l^{\texttt{OOB}}(\mathbf{x}_i) = \frac{1}{|\mathcal{T}|} \sum_{j \in \mathcal{T}} \frac{\mathbb{I}\left[\tau_j(\mathbf{x}_l) = \tau_j(\mathbf{x}_i)\right]}{\sum_{l'=1}^{N-1} \mathbb{I}\left[\tau_j(\mathbf{x}_{l'}) = \tau_j(\mathbf{x}_i)\right]},$$

which are of the form of (5.9) but only consider a subset of trees. Note that the i-th observation is removed from the original data set to avoid potential bias. Finally, for $k = 1, \ldots, K$, we estimate $\widehat{p}_{i,k}^{\texttt{OOB}}(\mathbf{x}_i) = \sum_{l=1}^{N-1} \mathbb{I}\left[\mathbf{y}_l = \tilde{\mathbf{y}}_k\right] \omega_l^{\texttt{OOB}}(\mathbf{x}_i)$.

[2] For simplicity, we assume a regression setting where the target variable is real-valued.

5.5.2 Prescriptive Barycentric Interpolation

Algorithm 5.1 `PrescrInterp`

Input: training data sets $\{\mathcal{D}_s\}_{s=1}^S$, local models $\{\widehat{\mathbf{p}}_s\}_{s=1}^S$, anchor probability vector $\mathbf{p}^{\texttt{anch}}(\mathbf{x})$

Output: hyperparameters $\{\alpha_s\}_{s=1}^S$

1: fix a grid of values, e.g., $\mathcal{A} = \{0.0, 0.1, \ldots, 1.0\}$

2: **for** $s = 1, \ldots, S$ **do**

3: **for** $\alpha \in \mathcal{A}, i = 1, \ldots, N_s$ **do**

4: find $\widehat{\mathbf{p}}_{s,i}^{\texttt{OOB}}(\mathbf{x}_{s,i})$ {OOB histogram}

5: $\mathbf{q}_{s,i,\alpha}^{\texttt{OOB}} = \arg\min_{\mathbf{q}} \alpha W(\mathbf{q}, \widehat{\mathbf{p}}_{s,i,k}^{\texttt{OOB}}(\mathbf{x}_{s,i})) + (1-\alpha) W(\mathbf{q}, \mathbf{p}^{\texttt{anch}}(\mathbf{x}_{s,i}))$ {barycentric interpolation}

6: **end for**

7: find $\alpha_s^* = \arg\min_{\alpha \in \mathcal{A}} \frac{1}{N_s} \sum_{i=1}^{N_s} D(\mathbf{q}_{s,i,\alpha}^{\texttt{OOB}}, \mathbf{y}_{s,i})$ {minimizes the OOB prescriptive cost}

8: **end for**

return $\{\alpha_s^*\}_{s=1}^S$

In this section, we propose an optimization-based approach that combines the prescriptive OOB method and data pooling for a collection of S problems with contextual information.

We assume access to local data sets $\mathcal{D}_s$ and models $\widehat{\mathbf{p}}_s$, as well as an anchor distribution $\mathbf{p}^{\texttt{anch}}(\mathbf{x})$ estimated from a data pooling procedure, e.g., the output of a global model with naive data pooling or an aggregation of $\widehat{\mathbf{p}}_s$ of the form of (5.14). Note that it is also possible to consider distributions that are not data-driven, e.g., a distribution provided by a domain expert given the context. Our goal is to determine when and how much data to pool in order to minimize the expected out-of-sample prescriptive cost. To achieve this, once again, we utilize the notion of the Wasserstein barycenter to interpolate between a local and an anchor distribution, allowing for a flexible combination of information from local data sets and the aggregated anchor distribution.

Let $\alpha \in [0, 1]$ be a hyperparameter that controls the amount of data pooling. The optimization-based interpolation algorithm is detailed in Algorithm 5.1. The algorithm begins by fixing a grid point of values for hyperparameter α. For each subproblem s, the algorithm iterates over the values of α and training observations $i = 1, \ldots, N_s$, and estimates a conditional distribution using the prescriptive OOB method. The algorithm then interpolates between the locally estimated and anchor distributions by estimating a barycenter whose coordinates are given by α — see Step 5 in Algorithm 5.1. For clarity, $\widehat{\mathbf{p}}_{s,i}^{\texttt{OOB}}(\mathbf{x}_{s,i})$ is the OOB probability vector given $\mathbf{x} = \mathbf{x}_{s,i}$, estimated from a subset of base learners from the ensemble model $\widehat{\mathbf{p}}_s$ which did not use the i-th observation for training (hence the superscript `OOB`). Further, $\mathbf{q}_{s,i,\alpha}^{\texttt{OOB}}$ is the α-weighted average distribution, in the sense of the Wasserstein distance, between $\widehat{\mathbf{p}}_{s,i}^{\texttt{OOB}}(\mathbf{x}_{s,i})$ and $\mathbf{p}^{\texttt{anchor}}(\mathbf{x}_{s,i})$. Evidently, $\alpha = 1$ retrieves the local solution, while $\alpha = 0$ maximizes the amount of data pooling. Finally, the

algorithm finds the value of α that minimizes the OOB estimate of the prescriptive cost.

For an out-of-sample realization of uncertainty, $\mathbf{x}_{s,0}$, we first estimate the α-weighted Wasserstein barycenter of the local and the anchor models, and then solve the respective problem. Note that a different hyperparameter α is selected for each problem. This way, problems with high-quality local data sets and, by extension, high-quality forecasting models will converge to the local approach faster, while the rest of the problems may still benefit from data pooling.

Alternatively, we propose interpolating between the local and the anchor distribution by minimizing the ℓ_2 distance, instead of the Wasserstein distance, by replacing Step 5 of Algorithm 5.1 with

$$\mathbf{q}^{\texttt{OOB}}_{s,i,\alpha} = \alpha \widehat{\mathbf{p}}^{\texttt{OOB}}_{s,i,k}(\mathbf{x}_{s,i}) + (1-\alpha)\mathbf{p}^{\texttt{anch}}(\mathbf{x}_{s,i}),$$

effectively creating a convex combination between the local and the anchor distribution. The barycentric interpolation using the ℓ_2 distance has the benefit of reducing the computational cost both for the offline training phase and the model deployment. Nonetheless, it creates a mixture of distributions that do not maintain the geometric structure and is less interpretable.

5.6 Numerical Experiments

In this section, we empirically validate data pooling for power system applications. First, we present various motivating applications (in Subsection 5.6.1). Then, we discuss our experimental setup and input data (in Subsection 5.6.2), and present the numerical results (in Subsections 5.6.3 and 5.6.4).

5.6.1 Motivating Power System Applications

In this section, we present the two motivating examples related to the integration of stochastic renewable energy sources in power systems and electricity markets, which we study in the numerical experiments.

Prediction In the prediction setting, our goal is to forecast the power production of a number of wind turbines in the day-ahead horizon given a set of weather forecasts derived from an NWP model. The cost function is given by

$$c(z;y) = (z-y)^2,$$

i.e., the standard MSE loss, and the feasible set is given by $\mathcal{Z} = \{z \,|\, 0 \leq z \leq 1\}$, i.e., forecast values are normalized by the nominal capacity.

Trading In the trading setting, we consider a renewable producer participating as a price-taker in a day-ahead market subject to imbalance penalties, assuming a dual-price balancing

mechanism. As before, the producer derives trading decisions given a set of weather forecasts from an NWP model. As previously discussed in Chapter 2.4.2, this problem is an instance of the well-known newsvendor problem [51], which aims to find the optimal replenishment quantity for a perishable product. The cost function is given by

$$c(z; y) = \max(\frac{\tau}{1-\tau}(y - z), (z - y)),$$

which is known as the pinball loss, with $0 < \tau < 1$ being the critical fractile. The full-information solution is the τ-th quantile of the distribution of y, i.e., the wind production. Similarly to the prediction problem, the feasible set is given by $\mathcal{Z} = \{z \,|\, 0 \leq z \leq 1\}$.

5.6.2 Experimental Setup and Input Data

In the numerical experiments, we investigate the performance of different data pooling methods in terms of the expected cost. Specifically, we examine the impact of the number of training observations, N_s, which are either fixed or randomly distributed across problems, and the effect of the number of problems S. The following methods are compared:

- `Local`: a standard approach where each subproblem is solved independently without any data pooling.
- `Pool-Naive`: naive data pooling with a global model.
- `Pool-OT`: OT-based data pooling.
- `Interp`: barycentric interpolation between `Local` and `Pool-OT` using the proposed prescriptive data pooling algorithm.

In all cases, we use random forest models to estimate the conditional distributions. For the prediction problem, we train a random forest model with 100 trees for each subproblem using default hyperparameters. For the trading problem, we use random forests trained to minimize the prescriptive cost criterion described in Chapter 2.3 and the same hyperparameters. For `Pool-Naive`, we consider the same model and hyperparameters as `Local` but trained on the concatenated data sets. For `Pool-OT`, we use the 1-Wasserstein metric to compute the barycenters and set $\lambda_s = \frac{n_s}{\sum_{s=1}^{S} n_s}$.

For input data, we use power measurements from $S = 50$ wind turbines located in mid-west France, with an aggregated nominal capacity of 100 MW. The available data sets span the period from January 2019 to April 2020 with an hourly resolution. We use the data from 2019 to sample training data sets and the remaining 5 months for testing. For both applications considered, i.e., prediction and trading in a day-ahead market, we consider a horizon of 12 to 36 hours ahead.

For contextual information, we use wind speed forecasts from an NWP model, issued daily at 00:00 UTC with a spatial resolution of $0.1^o \times 0.1^o$ and a forecast horizon of 96 hours ahead. For the s-th subproblem, $\mathbf{x}_s$ comprises the NWP model forecasts from the

closest grid point in terms of Euclidean distance. Wind production series are normalized and assumed to take values on the fixed grid $\{0.00, 0.01, \ldots, 1.00\}$.

5.6.3 Prediction Results

First, we consider a scenario where the number of training observations N_s is fixed across all problems and investigate the performance of the different methods as a function of N_s, as well as the number of problems (i.e., number of wind turbines) S. To obtain our results, we first sample S wind turbines and N_s training observations for each turbine, train both the local and global models, estimate the Wasserstein barycenters for each $\mathbf{x}$, and run Algorithm 5.1 for the combination method. We then evaluate the performance of each method on the test set. The process is repeated 10 times.

Fig. 5.1 presents the average MSE over the S subproblems and all the iterations. Overall, the results suggest that data pooling can be beneficial when data are scarce, but as the amount of data increases, the local approach, `Local`, becomes more reliable and outperforms data pooling for all values of S. This result is intuitive and corroborates the findings of previous works — see, e.g., [147]. Specifically, `Pool-Naive` is always the best-performing method for $N_s = 50$, followed by closely `Pool-OT`, while `Pool-OT` converges to slightly better performance for larger values of N_s. Further, we observe that the number of problems S has an effect on the performance of both `Pool-Naive` and `Pool-OT`, as both perform better for larger values of S, with the effect being more pronounced for smaller values of N_s, which indicates that data pooling benefits from the presence of multiple problems.

Note that Fig. 5.1 may also indicate that increasing the number of problems S is associated with improved performance for `Local`, which, however, is by design independent of S. Indeed, examining the performance for each local problem confirms that this effect is spurious and is attributed to the variability of the experiments.

Notably, the prescriptive data pooling performs consistently well, outperforming both `Local` and `Pool-OT`. When N_s is moderate to small, `Interp` is considerably better than `Local`, while when N_s is larger, `Interp` converges to similar or better performance than `Local`. This result indicates that the prescriptive data pooling algorithm does a good job of identifying how much data to pool given the size of the training sample, and that a small degree of data pooling offers benefits even for larger training samples.

Next, we repeat the previous experiment but randomly sample the number of training observations, N_s, for each subproblem from a normal distribution with a of mean 100 and standard deviation of 25. Table (5.1) summarizes the expected improvement in terms of MSE and the standard error of each method. Both `Pool-Naive` and `Pool-OT` perform, on average, better than `Local`, although, in several cases, the result is not statistically significant. Conversely, `Interp` leads to a considerable improvement over `Local` of approximately 4.71% over all the values of S. This further highlights the fact that a convex combination of the local and pooled methods can outperform both approaches. Moreover, the prescriptive data

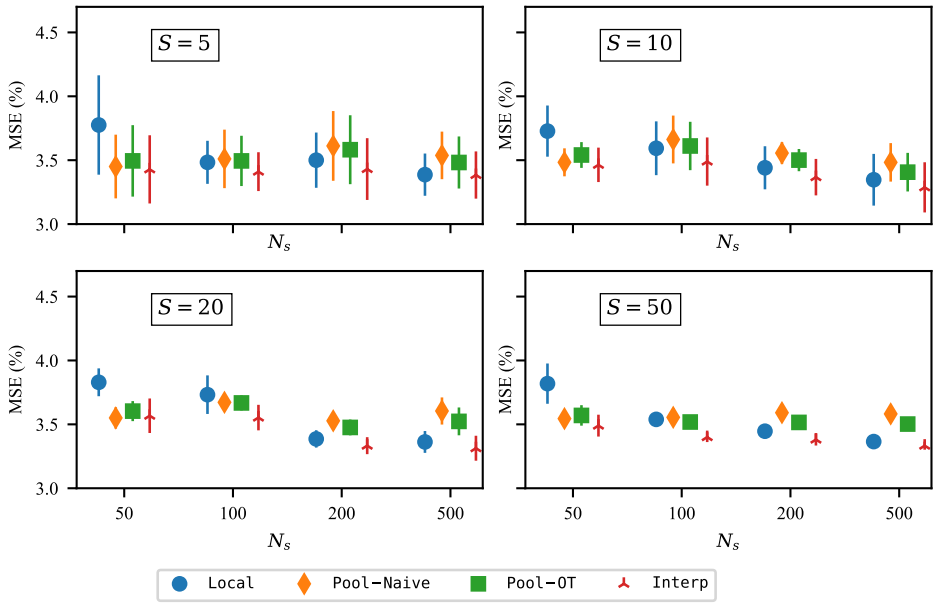


Figure 5.1: Average MSE versus sample size N_s (same for all subproblems). Error bars show ± 1 standard error.

Table 5.1: Average percentage (%) of MSE improvement over `Local`. Parentheses show the standard error.

	`Pool-Naive`	`Pool-OT`	`Interp`
$S = 5$	0.31 (1.87)	1.56 (1.73)	3.79 (1.27)
$S = 10$	3.55 (1.77)	3.48 (1.74)	5.99 (1.23)
$S = 20$	0.40 (0.44)	0.99 (0.43)	4.03 (0.34)
$S = 50$	1.40 (0.32)	2.26 (0.31)	5.02 (0.16)

pooling algorithm performs consistently well even when sample sizes vary across problems.

5.6.4 Trading Results

In this subsection, we consider the setting of trading renewable production in a day-ahead market. We present results for a fixed value of $\tau = 0.80$, i.e., the optimal trading offer equals the 80-th quantile of the wind production distribution, and measure performance in terms of expected pinball loss.

Similarly to the prediction problem, we first consider the scenario where the number of training observations N_s is fixed across all problems and investigate the performance of the different methods as a function of N_s, as well as the number of problems S, repeating the process 10 times. Fig. 5.2 presents the average pinball loss over the S subproblems and all the iterations. Overall, the results closely resemble the ones presented in the prediction problem. Specifically, data pooling outperforms the local approach when data are scarce,

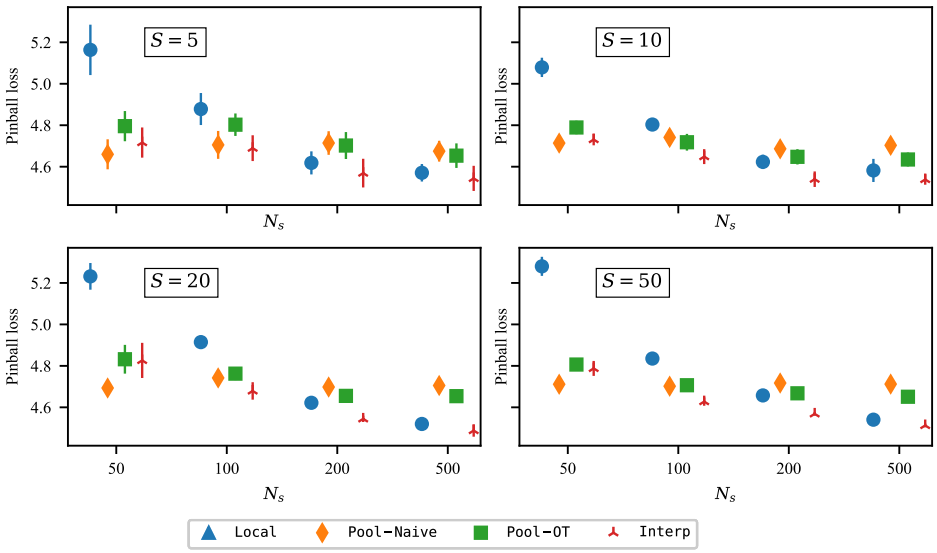


Figure 5.2: Average pinball loss for $\tau = 0.80$ versus sample size N_s (same for all subproblems). Error bars show ± 1 standard deviation.

Table 5.2: Average percentage (%) of pinball loss improvement over `Local`. Parentheses show the standard error.

	`Pool-Naive`	`Pool-OT`	`Interp`
$S = 5$	2.46 (1.11)	1.96 (0.88)	3.40 (0.57)
$S = 10$	4.31 (0.80)	3.18 (0.60)	4.89 (0.30)
$S = 20$	4.56 (0.67)	4.09 (0.66)	5.49 (0.54)
$S = 50$	4.69 (0.29)	4.55 (0.24)	5.71 (0.30)

the local approach converges to better performance as N_s increases, and the prescriptive data pooling algorithm combines the best of both worlds. The effect of S is also similar to the prediction problem.

We further examine the performance of the methods when the number of training observations for each problem is randomly sampled from a normal distribution, similar to the previous experiment. Table 5.2 summarizes the expected improvement in terms of pinball loss and the associated standard error of each method. Contrary to the previous problem examined, both data pooling methods, namely, `Pool-Naive` and `Pool-OT`, significantly outperform the local approach. Specifically, the expected improvement over `Local` is 4.01% for `Pool-Naive` and 3.45% for `Pool-OT`, respectively. Conversely, `Interp` leads to an expected improvement of 4.87% and is consistently the best-performing method for all values of S, which closely resembles the results presented in Table 5.1.

5.7 Conclusions

In this chapter, we investigated data pooling methods to address data scarcity when dealing with multiple contextually-dependent problems. Two approaches were examined, namely training a global model with naive data pooling and an OT-based method for combining estimated conditional distributions. We further developed a prescriptive data pooling algorithm that interpolates between a local and a pooled distribution based on the expected decision cost of the downstream optimization problem. For validation, we examined two pivotal applications related to the integration of renewable energy sources in power systems, namely renewable production forecasting and trading in a day-ahead market. Our empirical results illustrated that data pooling improves overall performance when data are scarce and, perhaps more importantly, our prescriptive data pooling algorithm correctly identifies when and how much data to pool, leading to consistently better performance than standalone and pooled methods.

Future work could focus on the case of both scarce and contaminated data, and developing data pooling methods that are robust to local outliers. Moreover, it is worth exploring adding entropic regularization to the estimation of the Wasserstein barycenter, which induces a smoothing effect and, potentially, could improve performance for smaller sample sizes.

Chapter 6

Conclusions and Future Directions

Résumé en Français

Les méthodes basées sur les données sont très prometteuses en tant que catalyseurs clés de la transition vers un réseau électrique décarboné et durable avec une part importante de sources d'énergie renouvelables. Dans cette thèse, nous avons exploré diverses directions de recherche pour développer des méthodes basées sur les données qui améliorent la prise de décision dans les systèmes électriques, en nous concentrant principalement sur un calendrier opérationnel. En particulier, cette thèse a contribué à permettre de meilleures décisions grâce à la prévision et à l'optimisation intégrées, à favoriser l'adoption d'outils d'analyse avancés grâce à des méthodes intrinsèquement interprétables et à permettre la résilience aux défis liés aux données, tels que les données manquantes dans un environnement opérationnel ou les données de formation rares. Dans l'ensemble, les méthodes et les résultats présentés soulignent l'importance d'aller au-delà des mesures de précision statistique et de se concentrer sur l'obtention d'une valeur de prévision plus élevée, ainsi que sur l'anticipation des défis potentiels liés au déploiement de méthodes basées sur les données dans des contextes réels, tels que les données manquantes, pour assurer la cohérence de leur production. Les méthodes et algorithmes proposés peuvent également être étendus dans plusieurs directions intéressantes. Par exemple, le cadre intégré de prévision et d'optimisation développé dans le Chapitre 2 pourrait être étendu au cas où le problème d'optimisation en aval change, par exemple avec l'ajout de nouvelles contraintes, tandis que la méthodologie du Chapitre 3 pourrait être étendu à d'autres tâches opérationnelles critiques telles que le flux d'énergie optimal soumis à des contraintes de sécurité. De plus, le cadre de régression robuste proposé au Chapitre 4 pourrait être étendu aux modèles non linéaires, tels que les modèles de réseaux neuronaux, tandis que la méthodologie de regroupement de données du Chapitre 5 pourrait être étendue à un cadre d'estimation décentralisé pour améliorer les problèmes de confidentialité des données. Dans l'ensemble, cette thèse contribue à améliorer l'efficacité et la fiabilité des systèmes électriques modernes en développant des méthodes avancées basées sur les données ainsi qu'en relevant les défis associés à leur déploiement dans un environnement opérationnel réel.

To mitigate the adverse effects of climate change, the electricity power sector is rapidly transitioning towards decarbonization through the integration of renewable energy sources, such as wind and solar. Nonetheless, the highly variable and uncertain nature of weather-dependent renewable energy sources poses major challenges in the current mode of operation. In this context, advanced data-driven methods, leveraging tools from machine learning, operations research, and data science, hold significant promise as key enablers in the transition towards a decarbonized and sustainable electricity grid.

In this thesis, we explored various research directions to develop data-driven methods that lead to better decisions and address challenges related to their deployment in real-world power systems, focusing on a short-term operational time frame. We took a holistic approach by examining the model chain that goes from data to uncertainty modeling and then to decisions, and contributed towards improving different aspects of data-driven decision-making processes. Specifically, this thesis contributed to enabling better decisions through integrated forecasting and optimization, fostering the adoption of advanced analytics tools through intrinsically interpretable methods, and enabling resilience to data-related challenges, such as missing data in an operational setting or scarce training data.

In Chapter 2, we investigated the interaction between forecasting and optimization, which are two integral components of data-driven decision-making. To maximize the value of forecasts and enable better decisions, we developed an integrated method that embeds the downstream decision problem within the forecasting model. Additionally, we proposed various metrics to evaluate the impact of data on decision performance. Through comprehensive numerical experiments concerning participation in electricity markets, we demonstrated that the proposed approach performs similarly or better than the current state-of-the-art methods, while also reducing the associated modeling effort. A key takeaway from Chapter 2 is that moving beyond standard statistical accuracy and embedding knowledge about the downstream optimization problem within forecasting models can significantly improve decisions and mitigate uncertainty.

Chapter 3 explored the use of machine learning to accelerate traditional power system workflows and further improve decision-making processes. We proposed an interpretable learning method to directly forecast the solutions of a constrained optimization problem with feasibility guarantees, using the DC-OPF problem as a guiding example. Comprehensive numerical experiments demonstrated that our proposed method performs comparably to state-of-the-art black-box methods while also offering interpretable insights. Overall, Chapter 3 highlights that interpretability does not have to come at the expense of performance and illustrates the importance of developing methods that can provide transparency to facilitate decision-making in complex and critical domains such as power systems.

The implicit assumption underpinning most data-driven methods is that data will always be available when needed in an operational setting, such as when a model is deployed in production. In Chapter 4, we addressed the challenge of missing data in an opera-

tional setting that can compromise model performance, and developed a practical approach to enable model resilience, using a forecasting task as a guiding example. Importantly, the proposed method is agnostic to the mechanism that generates missing data, hedging against the worst-case scenario, and maintains practicality compared to ad hoc solutions. A series of numerical experiments highlighted its efficacy in enabling resilience and consistent performance. A key takeaway from Chapter 4 is that model performance is contingent on data availability and quality. Hence, fully leveraging available data-driven methods requires ensuring reliable and consistent performance under challenges that frequently arise in real-world applications, such as missing data.

Chapter 5 further contributed to enabling resilience against data-related issues by addressing the challenge of scarce training data in complex, real-world systems, such as power systems, where a large number of decision problems are solved under uncertainty. In this context, the aggregate volume of data is typically large, but data on an individual problem level can be scarce, creating a "large-scale, small-data" regime that hinders the deployment of advanced methods, such as those presented in Chapter 2. To address this, we formulated various methods for pooling data across problems and further developed an optimization-based algorithm to tune the amount of data pooling. Through numerical experiments, we illustrated that data pooling enhances performance in the case of scarce data, and the proposed algorithm can be beneficial even as the amount of local data increases. A key takeaway from Chapter 5 is that effectively utilizing advanced data-driven methods requires novel tools that can exploit available data from various sources.

This thesis has explored several promising research directions and there are still several interesting challenges and avenues for future work.

For the integrated forecasting and optimization framework developed in Chapter 2, future work could focus on extending the proposed framework to an online setting that readily adapts to shifts in the underlying distributions of uncertainty, as well as changes in the downstream optimization problem, such as the addition of new constraints. Another interesting challenge is reducing the computational cost of the integrated tree-based method through improved splitting heuristics. Furthermore, future work could consider cases where the prescribed decisions also affect the uncertain parameters, such as the case of a price-maker participating in wholesale electricity markets.

In Chapter 3, future work could explore the application of the proposed methodology to other critical tasks, such as Security Constrained DC-OPF or AC-OPF problems. From a methodology standpoint, the proposed framework to encode domain knowledge could also be adapted to other generic algorithms.

Regarding the robust regression framework developed in Chapter 4, future work could focus on extending the method to nonlinear models, such as neural network models, and jointly considering resilience against missing and corrupted data, from factors such as cyberattacks. Lastly, to minimize data leakage and ameliorate privacy concerns, future work

could adapt the data pooling methods developed in Chapter 5 in a decentralized estimation framework.

This thesis contributes to a diverse range of aspects of analytics tools for power systems. Our overarching goal is for the methods and tools developed in this thesis to complement each other, enabling better utilization of analytics tools for power systems. In future power systems that integrate a large number of heterogeneous assets, we envision a setting where streams of data such as updated weather predictions, production measurements, and market information are translated into value-maximizing decisions in a reliable, fast, and understandable manner. We believe that the methods and tools developed in this thesis will contribute towards this vision, improving the efficiency and reliability of power systems, and ultimately leading to a more sustainable future.

Bibliography

[1] Council of European Union, "On the promotion of the use of energy from renewable sources," 2018. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2018.328.01.0082.01.ENG&toc=OJ:L:2018:328:TOC

[2] IEA, "Renewables 2022," IEA, Paris, France, Tech. Rep., 2012. [Online]. Available: https://www.iea.org/reports/renewables-2022

[3] S. G. C. Group, "Smart grid reference architecture," CEN-CENELEC-ETSI, Tech. Rep., 2012.

[4] L. Baardman, R. Cristian, G. Perakis, D. Singhvi, O. Skali Lami, and L. Thayaparan, "The role of optimization in some recent advances in data-driven decision-making," *Mathematical Programming*, pp. 1–35, 2022.

[5] European Union, "Digitalisation of the energy system," 2022.

[6] D. Rolnick, P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. S. Ross, N. Milojevic-Dupont, N. Jaques, A. Waldman-Brown *et al.*, "Tackling climate change with machine learning," *ACM Computing Surveys (CSUR)*, vol. 55, no. 2, pp. 1–96, 2022.

[7] K. Lepenioti, A. Bousdekis, D. Apostolou, and G. Mentzas, "Prescriptive analytics: Literature review and research challenges," *International Journal of Information Management*, vol. 50, pp. 57–70, 2020.

[8] K. S. Perera, Z. Aung, and W. L. Woon, "Machine learning techniques for supporting renewable energy generation and integration: A survey," in *Data Analytics for Renewable Energy Integration*, W. L. Woon, Z. Aung, and S. Madnick, Eds. Cham: Springer International Publishing, 2014, pp. 81–96.

[9] T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, and H. Zareipour, "Energy forecasting: A review and outlook," *IEEE Open Access Journal of Power and Energy*, 2020.

[10] R. Sioshansi and A. J. Conejo, *Optimization in Engineering: Models and Algorithms*. Springer, 2019.

[11] J. M. Morales, A. J. Conejo, H. Madsen, P. Pinson, and M. Zugno, *Integrating renewables in electricity markets: operational problems*. Springer Science & Business Media, 2013, vol. 205.

[12] J. R. Birge and F. Louveaux, *Introduction to stochastic programming.* Springer Science & Business Media, 2011.

[13] L. A. Roald, D. Pozo, A. Papavasiliou, D. K. Molzahn, J. Kazempour, and A. Conejo, "Power systems optimization under uncertainty: A review of methods and applications," *Electric Power Systems Research*, vol. 214, p. 108725, 2023.

[14] H. Zareipour, C. A. Canizares, and K. Bhattacharya, "Economic impact of electricity market price forecasting errors: A demand-side analysis," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 254–262, 2009.

[15] G. G. Loke, Q. Tang, and Y. Xiao, "Decision-Driven Regularization: A Blended Model for Predict-then-Optimize," Rochester, NY, Feb. 2022. [Online]. Available: https://papers.ssrn.com/abstract=3623006

[16] S. Camal, A. Michiorri, and G. Kariniotakis, "Optimal offer of automatic frequency restoration reserve from a combined pv/wind virtual power plant," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6155–6170, 2018.

[17] S. Chatzivasileiadis, A. Venzke, J. Stiasny, and G. Misyris, "Machine learning in power systems: Is it time to trust it?" *IEEE Power and Energy Magazine*, vol. 20, no. 3, pp. 32–41, 2022.

[18] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, "Data management challenges in production machine learning," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1723–1726.

[19] J.-F. Toubeau, J. Bottieau, F. Vallée, and Z. De Grève, "Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1203–1215, 2018.

[20] D. Bertsimas and N. Kallus, "From predictive to prescriptive analytics," *Management Science*, vol. 66, no. 3, pp. 1025–1044, 2020.

[21] A. Georghiou, D. Kuhn, and W. Wiesemann, "The decision rule approach to optimization under uncertainty: methodology and applications," *Computational Management Science*, vol. 16, no. 4, pp. 545–576, 2019.

[22] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory.* SIAM, 2014.

[23] M. Qi and Z.-J. Shen, "Integrating prediction/estimation and optimization with applications in operations management," in *Tutorials in Operations Research: Emerging and Impactful Topics in Operations.* INFORMS, 2022, pp. 36–58.

[24] Y. Bengio, "Using a financial training criterion rather than a prediction criterion," *International journal of neural systems*, vol. 8, no. 04, pp. 433–443, 1997.

[25] B. Amos and J. Z. Kolter, "OptNet: Differentiable optimization as a layer in neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70, 06–11 Aug 2017, pp. 136–145.

[26] P. L. Donti, B. Amos, and J. Z. Kolter, "Task-based end-to-end model learning in stochastic optimization," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5490–5500.

[27] A. N. Elmachtoub and P. Grigas, "Smart "predict, then optimize"," *Management Science*, vol. 68, no. 1, pp. 9–26, 2022.

[28] A. Elmachtoub, J. C. N. Liang, and R. McNellis, "Decision trees for decision-making under the predict-then-optimize framework," in *International Conference on Machine Learning*, 2020, pp. 2858–2867.

[29] M. A. Muñoz, S. Pineda, and J. M. Morales, "A bilevel framework for decision-making under uncertainty with contextual information," *Omega*, vol. 108, p. 102575, 2022.

[30] G.-Y. Ban and C. Rudin, "The big data newsvendor: Practical insights from machine learning," *Operations Research*, vol. 67, no. 1, pp. 90–108, 2019.

[31] D. Bertsimas and B. Van Parys, "Bootstrap robust prescriptive analytics," *Mathematical Programming*, pp. 1–40, 2021.

[32] D. Bertsimas and C. McCord, "From predictions to prescriptions in multistage optimization problems," *arXiv:1904.11637*, 2019.

[33] D. Bertsimas, C. McCord, and B. Sturt, "Dynamic optimization with side information," *European Journal of Operational Research*, vol. 304, no. 2, pp. 634–651, 2023.

[34] R. Kannan, G. Bayraksan, and J. R. Luedtke, "Data-driven sample average approximation with covariate information," *arXiv:2207.13554*, 2022.

[35] A. Esteban-Pérez and J. M. Morales, "Distributionally robust stochastic programs with side information based on trimmings," *Mathematical Programming*, vol. 195, no. 1-2, pp. 1069–1105, 2022.

[36] N. Kallus and X. Mao, "Stochastic optimization forests," *Management Science*, vol. 69, no. 4, pp. 1975–1994, 2023.

[37] N. Mundru, "Predictive and prescriptive methods in operations research and machine learning: an optimization approach," Ph.D. dissertation, Massachusetts Institute of Technology, 2019.

[38] A. Corredera and C. Ruiz, "Prescriptive selection of machine learning hyperparameters with applications in power markets: Retailer's optimal trading," *European Journal of Operational Research*, vol. 306, no. 1, pp. 370–388, 2023.

[39] Y. Wang and L. Wu, "Improving economic values of day-ahead load forecasts to real-time power system operations," *IET Generation, Transmission & Distribution*, vol. 11, no. 17, pp. 4238–4247, 2017.

[40] X. Chen, Y. Yang, Y. Liu, and L. Wu, "Feature-driven economic improvement for network-constrained unit commitment: A closed-loop predict-and-optimize framework," *IEEE Transactions on Power Systems*, vol. 37, no. 4, pp. 3104–3118, 2022.

[41] X. Chen, Y. Liu, and L. Wu, "Towards improving operation economics: A bilevel mip-

based closed-loop predict-and-optimize framework for prescribing unit commitment," *arXiv:2208.13065*, 2023.

[42] O. Yurdakul, F. Qiu, and S. Albayrak, "Predictive prescription of unit commitment decisions under net load uncertainty," in *2023 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2023, pp. 1–5.

[43] J. Han, L. Yan, and Z. Li, "A task-based day-ahead load forecasting model for stochastic economic dispatch," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5294–5304, 2021.

[44] L. Sang, Y. Xu, H. Long, Q. Hu, and H. Sun, "Electricity price prediction for energy storage system arbitrage: A decision-focused approach," *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 2822–2832, 2022.

[45] A. Esteban-Pérez and J. M. Morales, "Distributionally robust optimal power flow with contextual information," *European Journal of Operational Research*, vol. 306, no. 3, pp. 1047–1058, 2023.

[46] G. Li and H.-D. Chiang, "Toward cost-oriented forecasting of wind power generation," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2508–2517, 2016.

[47] J. Zhang, Y. Wang, and G. Hug, "Cost-oriented load forecasting," *Electric Power Systems Research*, vol. 205, p. 107723, 2022.

[48] Z. Liang, R. Mieth, and Y. Dvorkin, "Operation-adversarial scenario generation," *Electric Power Systems Research*, vol. 212, p. 108451, 2022.

[49] J. Morales, M. Muñoz, and S. Pineda, "Prescribing net demand for two-stage electricity generation scheduling," *Operations Research Perspectives*, vol. 10, p. 100268, 2023.

[50] A. Stratigakos, A. Michiorri, and G. Kariniotakis, "A value-oriented price forecasting approach to optimize trading of renewable generation," in *2021 IEEE Madrid PowerTech*, 2021, pp. 1–6.

[51] M. Muñoz, J. M. Morales, and S. Pineda, "Feature-driven improvement of renewable energy forecasting and trading," *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 3753–3763, 2020.

[52] M. A. Muñoz, P. Pinson, and J. Kazempour, "Online decision making for trading wind energy," *arXiv:2209.02009*, 2023.

[53] T. Carriere and G. Kariniotakis, "An integrated approach for value-oriented energy forecasting and data-driven decision-making application to renewable energy trading," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6933–6944, 2019.

[54] G. Henri and N. Lu, "A supervised machine learning approach to control energy storage devices," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 5910–5919, 2019.

[55] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[56] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[57] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[58] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," *Advances in neural information processing systems 26*, 2013.

[59] P. Pinson, C. Chevallier, and G. N. Kariniotakis, "Trading wind generation from short-term probabilistic forecasts of wind power," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1148–1156, 2007.

[60] C. J. Dent, J. W. Bialek, and B. F. Hobbs, "Opportunity cost bidding by wind generators in forward markets: Analytical results," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1600–1608, 2011.

[61] E. Y. Bitar, R. Rajagopal, P. P. Khargonekar, K. Poolla, and P. Varaiya, "Bringing wind energy to market," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1225–1235, 2012.

[62] J. M. Morales, A. J. Conejo, and J. Pérez-Ruiz, "Short-term trading for a wind power producer," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 554–564, 2010.

[63] E. Du, N. Zhang, C. Kang, B. Kroposki, H. Huang, M. Miao, and Q. Xia, "Managing wind power uncertainty through strategic reserve purchasing," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2547–2559, 2016.

[64] M. Zugno, T. Jónsson, and P. Pinson, "Trading wind energy on the basis of probabilistic forecasts both of wind generation and of market quantities," *Wind Energy*, vol. 16, no. 6, pp. 909–926, 2013.

[65] J. Browell, "Risk constrained trading strategies for stochastic generation with a single-price balancing market," *Energies*, vol. 11, no. 6, p. 1345, 2018.

[66] H. Ding, P. Pinson, Z. Hu, and Y. Song, "Optimal offering and operating strategies for wind-storage systems with linear decision rules," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4755–4764, 2016.

[67] D. Bertsimas and D. den Hertog, *Robust and adaptive optimization*. Dynamic Ideas LLC, 2020, vol. 958.

[68] M. Beykirch, T. Janke, and F. Steinke, "Bidding and scheduling in energy markets: Which probabilistic forecast do we need?" in *2022 17th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, 2022, pp. 1–6.

[69] ENTSO-E, "Transparency platform." [Online]. Available: https://transparency.entsoe.eu/

[70] N. Meinshausen and G. Ridgeway, "Quantile regression forests." *Journal of Machine Learning Research*, vol. 7, no. 6, 2006.

[71] K. Bellinguer, V. Mahler, S. Camal, and G. Kariniotakis, "Probabilistic forecasting of regional wind power generation for the eem20 competition: a physics-oriented machine

learning approach," in *2020 17th International Conference on the European Energy Market (EEM)*, 2020, pp. 1–6.

[72] T. Jónsson, P. Pinson, H. A. Nielsen, and H. Madsen, "Exponential smoothing approaches for prediction in real-time electricity markets," *Energies*, vol. 7, no. 6, pp. 3710–3732, 2014.

[73] P. Pinson, H. Madsen, H. A. Nielsen, G. Papaefthymiou, and B. Klöckl, "From probabilistic forecasts to statistical scenarios of short-term wind power production," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 12, no. 1, pp. 51–62, 2009.

[74] B. Stott, J. Jardim, and O. Alsac, "DC power flow revisited," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290–1300, 2009.

[75] ENTSO-E, "Options for the design of european electricity markets in 2030." [Online]. Available: https://eepublicdownloads.entsoe.eu/clean-documents/Publications/Market%20Committee%20publications/210331_Market_design%202030.pdf

[76] L. Duchesne, E. Karangelos, and L. Wehenkel, "Recent developments in machine learning for energy systems reliability management," *Proceedings of the IEEE*, vol. 108, no. 9, pp. 1656–1676, 2020.

[77] M. E. Kaminski, "The right to explanation, explained," *Berkeley Technology Law Journal*, vol. 34, no. 1, pp. 189–218, 2019.

[78] X. Pan, T. Zhao, and M. Chen, "DeepOPF: Deep neural network for DC optimal power flow," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2019, pp. 1–6.

[79] X. Pan, T. Zhao, M. Chen, and S. Zhang, "DeepOPF: A deep neural network approach for security-constrained DC optimal power flow," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 1725–1735, 2021.

[80] A. Venzke, G. Qu, S. Low, and S. Chatzivasileiadis, "Learning optimal power flow: Worst-case guarantees for neural networks," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2020, pp. 1–7.

[81] R. Nellikkath and S. Chatzivasileiadis, "Physics-informed neural networks for minimising worst-case violations in DC optimal power flow," in *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2021, pp. 419–424.

[82] T. Zhao, X. Pan, M. Chen, and S. H. Low, "Ensuring DNN solution feasibility for optimization problems with convex constraints and its application to DC optimal power flow problems," *arXiv:2112.08091*, 2021.

[83] J. W. Dunn, "Optimal trees for prediction and prescription," Ph.D. dissertation, Massachusetts Institute of Technology, 2018.

[84] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella, "An artificial intelligence

framework for online transient stability assessment of power systems," *IEEE Transactions on Power Systems*, vol. 4, no. 2, pp. 789–800, 1989.

[85] J. L. Cremer, I. Konstantelos, and G. Strbac, "From optimization-based machine learning to interpretable security rules for operation," *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 3826–3836, 2019.

[86] Y. Ng, S. Misra, L. A. Roald, and S. Backhaus, "Statistical learning for DC optimal power flow," in *2018 Power Systems Computation Conference (PSCC)*, 2018, pp. 1–7.

[87] S. Misra, L. Roald, and Y. Ng, "Learning for constrained optimization: Identifying optimal active constraint sets," *INFORMS Journal on Computing*, vol. 34, no. 1, p. 463–480, jan 2022.

[88] D. Deka and S. Misra, "Learning for DC-OPF: Classifying active sets using neural nets," in *2019 IEEE Milan PowerTech*, 2019, pp. 1–6.

[89] Y. Chen and B. Zhang, "Learning to solve network flow problems via neural decoding," *arXiv:2002.04091*, 2020.

[90] A. Robson, M. Jamei, C. Ududec, and L. Mones, "Learning an optimally reduced formulation of OPF through meta-optimization," *arXiv:1911.06784*, 2019.

[91] L. Chen, M. Sim, X. Zhang, and M. Zhou, "Robust explainable prescriptive analytics," *Available at SSRN 4106222*, 2022.

[92] I. Pappas, D. Kenefake, B. Burnak, S. Avraamidou, H. S. Ganesh, J. Katz, N. A. Diangelakis, and E. N. Pistikopoulos, "Multiparametric programming in process systems engineering: Recent developments and path forward," *Frontiers in Chemical Engineering*, vol. 2, p. 620168, 2021.

[93] B. Eldridge, R. P. O'Neill, and A. R. Castillo, "Marginal loss calculations for the dcopf," Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2016.

[94] F. Zhou, J. Anderson, and S. H. Low, "The optimal power flow operator: Theory and computation," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 2, pp. 1010–1022, 2020.

[95] E. T. Maddalena, R. K. H. Galvão, and R. J. M. Afonso, "Robust region elimination for piecewise affine control laws," *Automatica*, vol. 99, pp. 333–337, 2019.

[96] S. K. Murthy, S. Kasif, S. Salzberg, and R. Beigel, "Oc1: A randomized algorithm for building oblique decision trees," in *Proceedings of AAAI*, vol. 93, 1993, pp. 322–327.

[97] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[98] D. Bertsimas, I. Dunning, and M. Lubin, "Reformulation versus cutting-planes for robust optimization: A computational study," *Computational Management Science*, vol. 13, pp. 195–217, 2016.

[99] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco,

R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang *et al.*, "The power grid library for benchmarking ac optimal power flow algorithms," *arXiv:1908.02788*, 2019.

[100] https://git.persee.mines-paristech.fr/akylas.stratigakos/interpretable-learning-dcopf.

[101] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: https://www.gurobi.com

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

[103] J. Nowotarski and R. Weron, "Recent advances in electricity price forecasting: A review of probabilistic forecasting," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1548–1568, 2018.

[104] Y. Zhang, J. Wang, and X. Wang, "Review on probabilistic forecasting of wind power generation," *Renewable and Sustainable Energy Reviews*, vol. 32, pp. 255–270, 2014.

[105] D. W. Van der Meer, J. Widén, and J. Munkhammar, "Review on probabilistic forecasting of photovoltaic power production and electricity consumption," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1484–1512, 2018.

[106] M. Bohlke-Schneider, S. Kapoor, and T. Januschowski, "Resilient neural forecasting systems," in *Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning*, 2020, pp. 1–5.

[107] European Commission, "A review of the entso-e transparency platform," 2017. [Online]. Available: https://energy.ec.europa.eu/system/files/2018-05/review_of_the_entso_e_plattform_0.pdf

[108] European Centre for Medium-Range Weather Forecasts, "2016 Survey: MARS," 2016. [Online]. Available: https://confluence.ecmwf.int/display/UDOC/2016+Survey%3A+MARS

[109] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.

[110] I. R. White, P. Royston, and A. M. Wood, "Multiple imputation using chained equations: issues and guidance for practice," *Statistics in medicine*, vol. 30, no. 4, pp. 377–399, 2011.

[111] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[112] A. Stratigakos, D. van der Meer, S. Camal, and G. Kariniotakis, "End-to-end learning for hierarchical forecasting of renewable energy production with missing values," in *2022 17th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, 2022, pp. 1–6.

[113] D. Bertsimas, A. Delarue, and J. Pauphilet, "Beyond impute-then-regress: Adapting prediction to missing data," *arXiv:2104.03158*, 2021.

[114] R. Tawn, J. Browell, and I. Dinwoodie, "Missing data in wind farm time series: Prop-

erties and effect on forecasts," *Electric Power Systems Research*, vol. 189, p. 106640, 2020.

[115] H. Wen, P. Pinson, J. Gu, and Z. Jin, "Wind energy forecasting with missing values within a fully conditional specification framework," *International Journal of Forecasting*, 2023.

[116] A. Gerossier, R. Girard, A. Bocquet, and G. Kariniotakis, "Robust day-ahead forecasting of household electricity demand and operational challenges," *Energies*, vol. 11, no. 12, p. 3503, 2018.

[117] Q. Li, Y. Xu, B. Chew, H. Ding, and L. Zhao, "An integrated missing-data tolerant model for probabilistic pv power generation forecasting," *IEEE Transactions on Power Systems*, vol. 37, no. 6, pp. 4447–4459, 2022.

[118] J. Luo, T. Hong, and S.-C. Fang, "Benchmarking robustness of load forecasting models under data integrity attacks," *International Journal of Forecasting*, vol. 34, no. 1, pp. 89–104, 2018.

[119] ——, "Robust regression models for load forecasting," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5397–5404, 2018.

[120] Y. Zhang, F. Lin, and K. Wang, "Robustness of short-term wind power forecasting against false data injection attacks," *Energies*, vol. 13, no. 15, p. 3780, 2020.

[121] Y. Liang, D. He, and D. Chen, "Poisoning attack on load forecasting," in *2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, 2019, pp. 1230–1235.

[122] J. Luo, T. Hong, and M. Yue, "Real-time anomaly detection for very short-term load forecasting," *Journal of Modern Power Systems and Clean Energy*, vol. 6, no. 2, pp. 235–243, 2018.

[123] M. Yue, T. Hong, and J. Wang, "Descriptive analytics-based anomaly detection for cybersecure load forecasting," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 5964–5974, 2019.

[124] M. Cui, J. Wang, and M. Yue, "Machine learning-based anomaly detection for load forecasting under cyberattacks," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5724–5734, 2019.

[125] Y. Chen, Y. Tan, and B. Zhang, "Exploiting vulnerabilities of load forecasting through adversarial attacks," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, 2019, pp. 1–11.

[126] Y. Zhou, Z. Ding, Q. Wen, and Y. Wang, "Robust load forecasting towards adversarial attacks via bayesian learning," *IEEE Transactions on Power Systems*, vol. 38, no. 2, pp. 1445–1459, 2023.

[127] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[128] H. Xu, C. Caramanis, and S. Mannor, "Robust regression and lasso," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3561–3574, 2010.

[129] L. El Ghaoui and H. Lebret, "Robust solutions to least-squares problems with uncertain data," *SIAM Journal on matrix analysis and applications*, vol. 18, no. 4, pp. 1035–1064, 1997.

[130] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM review*, vol. 53, no. 3, pp. 464–501, 2011.

[131] C. Caramanis, S. Mannor, and H. Xu, "Robust optimization in machine learning," in *Optimization for machine learning*, S. Sra, S. Nowozin, and S. J. Wright, Eds. MIT Press, 2012, pp. 369–402.

[132] D. Bertsimas, J. Dunn, C. Pawlowski, and Y. D. Zhuo, "Robust classification," *INFORMS Journal on Optimization*, vol. 1, no. 1, pp. 2–34, 2019.

[133] D. Bertsimas, X. Boix, K. V. Carballo, and D. d. Hertog, "A robust optimization approach to deep learning," *arXiv:2112.09279*, 2021.

[134] A. Globerson and S. Roweis, "Nightmare at test time: robust learning by feature deletion," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 353–360.

[135] B. L. Gorissen and D. Den Hertog, "Robust counterparts of inequalities containing sums of maxima of linear functions," *European Journal of Operational Research*, vol. 227, no. 1, pp. 30–43, 2013.

[136] R. Koenker and K. F. Hallock, "Quantile regression," *Journal of economic perspectives*, vol. 15, no. 4, pp. 143–156, 2001.

[137] T. Hong, "Short term electric load forecasting," Ph.D. dissertation, North Carolina State University, 2010.

[138] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization*. Athena Scientific Belmont, MA, 1997, vol. 6.

[139] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[140] D. Bertsimas and N. Koduri, "Data-driven optimization: A reproducing kernel hilbert space approach," *Operations Research*, vol. 70, no. 1, pp. 454–471, 2022.

[141] T. Januschowski, Y. Wang, K. Torkkola, T. Erkkilä, H. Hasson, and J. Gasthaus, "Forecasting with trees," *International Journal of Forecasting*, vol. 38, no. 4, pp. 1473–1481, 2022, special Issue: M5 competition.

[142] T. Hong, P. Pinson, and S. Fan, "Global energy forecasting competition 2012," *International Journal of Forecasting*, vol. 30, no. 2, pp. 357–363, 2014.

[143] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond," *International Journal of Forecasting*, vol. 32, no. 3, pp. 896–913, 2016.

[144] L. Cavalcante, R. J. Bessa, M. Reis, and J. Browell, "Lasso vector autoregression

structures for very short-term wind power forecasting," *Wind Energy*, vol. 20, no. 4, pp. 657–675, 2017.

[145] G. Peyré and M. Cuturi, "Computational optimal transport," *Foundations and Trends in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[146] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media, 2009.

[147] V. Gupta and N. Kallus, "Data pooling in stochastic optimization," *Management Science*, vol. 68, no. 3, pp. 1595–1615, 2022.

[148] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[149] P. Montero-Manso and R. J. Hyndman, "Principles and algorithms for forecasting groups of time series: Locality and globality," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1632–1653, 2021.

[150] H. Kazmi, Í. Munné-Collado, F. Mehmood, T. A. Syed, and J. Driesen, "Towards data-driven energy communities: A review of open-source datasets, models and tools," *Renewable and Sustainable Energy Reviews*, vol. 148, p. 111290, 2021.

[151] A. Balint, H. Raja, J. Driesen, and H. Kazmi, "Using domain-augmented federated learning to model thermostatically controlled loads," *IEEE Transactions on Smart Grid*, pp. 1–1, 2023.

[152] M. Grabner, Y. Wang, Q. Wen, B. Blažič, and V. Štruc, "A global modeling framework for load forecasting in distribution networks," *IEEE Transactions on Smart Grid*, pp. 1–1, 2023.

[153] J. Bottieau, Z. De Grève, T. Piraux, A. Dubois, F. Vallée, and J.-F. Toubeau, "A cross-learning approach for cold-start forecasting of residential photovoltaic generation," *Electric Power Systems Research*, vol. 212, p. 108415, 2022.

[154] G. Papayiannis and A. Yannacopoulos, "A learning algorithm based on experts' opinions," *Available at SSRN 2605905*, 2015.

[155] G. I. Papayiannis and A. N. Yannacopoulos, "A learning algorithm for source aggregation," *Mathematical Methods in the Applied Sciences*, vol. 41, no. 3, pp. 1033–1039, 2018.

[156] G. Papayiannis, G. Galanis, and A. Yannacopoulos, "Model aggregation using optimal transport and applications in wind speed forecasting," *Environmetrics*, vol. 29, no. 8, p. e2531, 2018.

[157] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.

[158] M. Cuturi and G. Peyré, "A smoothed dual approach for variational wasserstein problems," *SIAM Journal on Imaging Sciences*, vol. 9, no. 1, pp. 320–343, 2016.

[159] M. Agueh and G. Carlier, "Barycenters in the wasserstein space," *SIAM Journal on Mathematical Analysis*, vol. 43, no. 2, pp. 904–924, 2011.

[160] N. Feyeux, A. Vidard, and M. Nodet, "Optimal transport for variational data assimilation," *Nonlinear Processes in Geophysics*, vol. 25, no. 1, pp. 55–66, 2018.

[161] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," *CoRR*, vol. abs/2107.07511, 2021.

[162] P. Grigas, M. Qi, and Z.-J. M. Shen, "Integrated conditional estimation-optimization," *arXiv:2110.12351*, 2021.

[163] L. Breiman, "Out-of-bag estimation." Technical report, Statistics Department, University of California Berkeley, 1996.

RÉSUMÉ

Pour atténuer les effets néfastes du changement climatique, le secteur de l'électricité passe rapidement à la décarbonation grâce à l'intégration de sources d'énergie renouvelables, telles que l'éolien et le solaire. Dans ce contexte, les méthodes avancées basées sur les données, tirant parti des outils de l'apprentissage automatique et de la recherche opérationnelle, sont très prometteuses en tant que catalyseurs clés pour faire face à l'incertitude et à la variabilité des sources d'énergie renouvelables dépendantes des conditions météorologiques. Dans cette thèse, nous adoptons une approche holistique en examinant la chaîne de modèles qui va des données à la modélisation de l'incertitude, puis aux décisions et développons des méthodes basées sur les données qui permettent une prise de décision améliorée et résiliente dans les systèmes électriques modernes. Pour maximiser la valeur des prévisions, nous développons une méthode qui intègre la prévision et l'optimisation et proposons un cadre pour évaluer l'impact des données sur les décisions. Pour favoriser l'adoption de méthodes avancées basées sur les données et accélérer les flux de travail traditionnels, nous développons une méthode interprétable pour prévoir les solutions aux problèmes d'optimisation sous contraintes. Pour renforcer la résilience des modèles face aux données problématiques nous proposons une approche qui permet de gérer les données manquantes dans un cadre opérationnel. Nous proposons également une méthode basée sur l'optimisation pour regrouper les données sur un certain nombre de problèmes indépendants, améliorant ainsi les performances globales et la robustesse des décisions. Les méthodes proposées sont validées dans diverses expériences liées à l'exploitation du système électrique et à la participation aux marchés de l'électricité.

MOTS CLÉS

Prévision énergétique, science de la donnée, apprentissage automatique, optimisation, système électrique, analyse prescriptive, sources d'énergie renouvelables.

ABSTRACT

To mitigate the adverse effects of climate change, the power sector is rapidly transitioning towards decarbonization through the integration of renewable energy sources, such as wind and solar. In this context, advanced data-driven methods, leveraging tools from machine learning and operations research, hold significant promise as key enablers to deal with the uncertainty and variability of weather-dependent renewable energy sources. In this thesis, we take a holistic approach by examining the model chain that goes from data to uncertainty modeling and then to decisions and develop data-driven methods that enable improved and resilient decision-making in modern power systems. To maximize forecast value, we develop a method that integrates forecasting and optimization and propose a framework to evaluate the impact of data on decisions. To foster the adoption of advanced data-driven methods and speed up traditional workflows, we develop an interpretable method to forecast the solutions to constrained optimization problems. To enhance resilience against data-related challenges, we propose a principled approach to handle missing data in an operational setting and develop an optimization-based method to pool data across a number of independent problems, thereby improving the overall performance and robustness of decisions. The proposed methods are validated in various experiments related to power system operations and participation in electricity markets. Overall, the methods and tools developed in this thesis contribute to the transition towards a decarbonized and sustainable electricity grid.

KEYWORDS

Energy forecasting, data science, machine learning, optimization, power system, prescriptive analytics, renewable energy sources.