INCENTIVIZING RENEWABLES AND REDUCING GRID IMBALANCES THROUGH MARKET INTERACTION

A FORECASTING AND CONTROL APPROACH

INCENTIVIZING RENEWABLES AND REDUCING GRID IMBALANCES THROUGH MARKET INTERACTION

A FORECASTING AND CONTROL APPROACH

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus Prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Monday, 28 September 2020 at 15:00

by

Jesus LAGO GARCIA

Master of Science in Microsystems Engineering,
University of Freiburg, Germany,
born in Vigo, Spain.

This dissertation has been approved by the promotor:

Prof. dr. ir. B. De Schutter

Composition of the doctoral committee:

Rector Magnificus,	chairman
Prof. dr. ir. B. De Schutter,	Delft University of Technology, promotor
Independent members:	
Prof. dr. ir. Z. Lukszo	Delft University of Technology
Prof. dr. P. Palensky	Delft University of Technology
Prof. dr. ir. J. M. A. Scherpen	University of Groningen
Prof. dr. M. Diehl	University of Freiburg
Prof. dr. C. Ocampo-Martinez	Polytechnic University of Catalonia
Other members:	
Dr. G. Suryanarayana	Flemish Institute for Technological Research

This research was supported by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 675318 (INCITE) and by the Flemish Institute for Technological Research (VITO).

Printed by: Jesus Lago
Email: jesuslagogarcia@gmail.com
ISBN: 978-94-6402-444-9

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

If you thought that science was certain — well, that is just an error on your part.

Richard P. Feynman

PREFACE

Doing research, and specially pursuing a PhD, might seem very daunting at first: too many unanswered questions, all the different paths that one might take, the solitude that sometimes comes with the job... As many other PhDs before me, these thoughts were fiddling with my mind even before I started this venture. Now, getting almost to the end of the journey, I realize that these thoughts have actually been useful companions that helped me to reach this point. The unknown gave me knowledge, wondering gave me wisdom, hesitating taught me to lose my fears, and solitude clarified my mind.

I would like to say that reaching this mindset was easy. The truth is that it was not. It was certainly only possible due to the numerous people that helped me and encouraged me along the way. First and foremost, this thesis would not have been possible without the endless support of Bart: without his supervision and continuous trust I would have never seen the end of this quest. I am particularly thankful for all the freedom that he granted me, it was key for my development into a truly independent researcher.

The same gratitude goes to Fjo and Gowri, their constant backing and guidance, and their freedom to let me choose my own research path. The great working environment at VITO and Energyville was also instrumental to reach this point. I am grateful to all my colleagues for creating such a wonderful atmosphere and working conditions. I am especially thankful to Georg and Hans for making the PhD trip easier to navigate, and to Brida, Chris, Gowri, Juliano, Davy, and Ana for their friendship.

All my other co-authors and research collaborators are also being held in equal high esteem. This PhD journey would have never been fruitful without the advice and co-operation of Ina De Jaeger, Tomas Pippia, Ksenia Poplavskaya, Grzegorz Marcjasz, Davy Geysen, Javier Arroyo, Dr. Nikolaos Sapountzoglou, Dr. Wiet Mazairac, Dr. Karel De Brabandere, Dr. Roel De Coninck, Dr. Peter Vrancx, Prof. Dirk Saelens, Prof. Michael Erhard, Prof. Rafał Weron, Prof. Laurens De Vries, Prof. Lieve Helsen, and Prof. Bertrand Raison. Thank you to all of you.

The INCITE project and all its members also deserve a special recognition. The feedback and courses provided during their biannual meetings were useful to redefine my research path and to improve my areas of expertise. The project was not only a hub for innovative thoughts and exchanges, but also a place where I gained great friends. Being part of the project has been a pleasure, but ending it with two good friends like Nikos and Tomas is priceless. The European Union also deserves all my gratitude, this venture would not have been possible without their funding through the INCITE Marie Skłodowska-Curie European Training Network.

I extend my appreciation to the companies 3E and Ecovat for hosting me for several months and for providing me with the tools to carry out my research in practical scenarios. I am specially thankful to Dr. Karel De Brabandere at 3E and to Mr. Aris De Groot at Ecovat for their insightful discussions and valuable feedback.

I would also like to thank my committee members, Prof. Moritz Diehl, Prof. Carlos Ocampo-Martinez, Prof. Zofia Lukszo, Prof. Peter Palensky, and Prof. Jacqueline Scherpen for their constructive comments and for their valuable time. They helped me greatly to improve this dissertation.

To all my friends, thank you for being there all the way and stand with my countless bad jokes and numerous factoids. Gizem and Zeki, my dearest friends in The Netherlands, I am specially grateful to have you in my life and being able to say that you are part of my family. Cem and Nidhi, despite the distance and the seldom chats, thank you for making me feel that nothing will ever change. Tomas, Carlos, Jesus, Manyu, Mattia, Momo, and Camilo, your friendships are one of the best things I get from my time at Delft. To Anton, Villar, Xoa, Ruiz, Pau and all my other friends in Spain, thank you for your valuable friendships in spite of the distance.

I would also like to use this opportunity to thank my parents, sisters, aunt, and grandparents. Their unconditional love and encouragement, even from Spain, have helped me reach the point where I am now and overcome many difficulties. Without their support I would have never seen the end of this voyage.

Last but not least, I would like to express my gratitude to my wife. Thank you for your endless sympathy and love, for your understanding during the last four years, for your valuable feedback and help in my research, for putting up with my academic complaints and worries, and for unconditionally being there whenever I needed.

Jesus Lago
Delft, September 2020

SUMMARY

As the penetration of *renewable energy sources (RESs)* increases, so does the dependence of electricity production on weather and, in turn, the uncertainty in electricity generation, the volatility in electricity prices, and the imbalances between production and consumption. In this context, while RES integration does complicate grid balance and increase price volatility, it also opens up opportunities for flexible market agents to reduce grid imbalances. In particular, by using the nature of the interactions between electricity markets and grid balance, market agents can reduce grid imbalances while increasing their profit. However, despite this obvious win-win situation, traditional research in this field has focused on balancing mechanisms that do not always exploit these relations between electricity markets and grid balance.

The aim of this thesis is to fill this scientific gap by exploiting the intrinsic relation between electricity markets and grid balancing. Particularly, the goal is to propose new modeling, forecasting, and control algorithms that increase the integration of RES and decrease the grid imbalances by using market interactions. The advantage of the proposed methods is that they allow more energy systems to participate in and contribute to grid balancing. The thesis comprises three parts: a) forecasting algorithms to reduce uncertainty; b) modeling and control of thermal seasonal storage to mitigate imbalances; c) new market mechanisms to ensure a wider participation in grid balancing.

As the uncertainty of RESs hinders their economic profits and makes the grid harder to balance, a first approach to exploit market interactions is to accurately predict future electricity prices and generation of renewables. Particularly, accurate and reliable forecasts lead to better decision making, higher economic profits, and lower uncertainty. This in turn translates into a grid that is easier to balance and larger economic incentives for integration of RESs. With this in mind, the first part of the thesis advances the field of forecasting methods by contributing to three research areas. First, motivated by the new EU market policies that aim at reaching a single and unified electricity market in Europe, we analyze the effect of market integration in electricity price dynamics and propose new forecasting models that exploit market integration to improve forecasting accuracy. Second, due to the advances of deep learning (DL) methods in several fields, we investigate the application of DL methods for electricity price forecasting and develop new DL forecasting techniques that achieve state-of-the-art results. Third, as forecasting short-term solar irradiance has become key for many applications, we propose a generalized short-term forecasting model that can forecast solar irradiance in any location without the need of ground measurements. The new method is paramount as solar generators are geographically dispersed and ground measurements are not always easy to obtain.

Improving the accuracy of forecasting techniques is an indirect approach to reduce the uncertainty in electricity trade and incentivize the integration of renewables. A more direct approach is to use energy storage systems to absorb the grid imbalances. In this

context, while long-term energy storage is arguably one of the most important elements to ensure the success of the energy transition, most of the existing technologies are only economically efficient for short-term and medium-term energy storage. Therefore, in the second part of this thesis, we investigate modeling and control techniques to ensure that seasonal storage systems maximize their profits while operating to reduce grid imbalances. First, as the existing models for thermal seasonal storage systems are too complex and cannot be efficiently integrated in control and optimization problems, we propose a new accurate model that can be integrated in real-time control and optimization applications. Second, we propose control algorithms for seasonal storage systems that, by explicitly exploiting the relation between imbalances and prices, reduce grid imbalances while maximizing profits. These algorithms are novel on their own as the control algorithms that currently exist for market interaction are limited to short-term horizons and are not suited for seasonal storage systems.

A more direct approach to incentivize the integration of renewables and keep the grid balanced is to explicitly modify the structure of electricity markets so that a larger number of energy systems have economic incentives to reduce grid imbalances. In particular, as traditional power plants are taken off the grid, it becomes clear that RES systems need to contribute to grid balancing if the grid is to remain stable. However, while some RES systems can potentially contribute to grid balancing, they are not being used for this purpose due to the current rules applied to system balancing. Examples of such systems include solar photovoltaic installations, storage systems such as seasonal storage, or even—in some countries—wind farms. With that motivation, in the third part of the thesis we investigate methods and new market structures that allow these systems to not only participate in balancing the grid, but to also have economic incentives to do so. In detail, we propose a new market framework for providing balancing services by trading with the imbalance settlement mechanism. The new framework incorporates newer systems into the portfolio of balancing providers and gives these systems economic incentives to balance the grid. As an additional advantage, it also incentivizes the use of long-term storage systems, which, as argued before, are key players in the energy transition.

SAMENVATTING

Naarmate de penetratie van hernieuwbare energiebronnen (HEB's) hoger wordt, neemt ook de weersafhankelijkheid van de elektriciteitsproductie toe, met een onzekerdere opwekking, volatielere elektriciteitsprijzen en een groter onevenwicht tussen productie en verbruik tot gevolg. In een dergelijke context opent de integratie van HEB's, hoewel die het handhaven van een netevenwicht complexer maakt en de prijsvolatiliteit doet toenemen, ook mogelijkheden voor flexibele marktspelers om onevenwichten in het net af te vlakken. Door met name in te spelen op de specifieke wisselwerking tussen elektriciteitsmarkten en netevenwicht kunnen marktspelers onevenwichten in het net verkleinen en tegelijk hun winst vergroten. Ondanks deze voor de hand liggende win-win concentreerde het traditionele onderzoek in dit domein zich echter veelal op balanceringsmechanismen die niet altijd gebruikmaken van die interactie tussen elektriciteitsmarkten en netevenwicht.

Het doel van dit proefschrift is om deze wetenschappelijke lacune op te vullen door in te spelen op de intrinsieke relatie tussen elektriciteitsmarkten en de balancering van het net. Het doel is vooral om nieuwe modellerings-, voorspellings- en controlealgoritmen voor te stellen die de integratie van HEB's verhogen en onevenwichten in het net verkleinen door uit te gaan van marktinteracties. Het voordeel van de voorgestelde methoden is dat ze meer energiesystemen toelaten om deel te nemen aan en bij te dragen tot de balancering van het net. Dit proefschrift bestaat uit drie delen: a) voorspellingsalgoritmen om de onzekerheid te verminderen; b) modellering en controle van thermische seizoenopslag om onevenwichten te beperken; c) nieuwe marktmechanismen om een bredere deelname aan de balancering van het net te waarborgen.

Aangezien de onzekerheid omtrent HEB's een obstakel is voor economische winst en het bereiken van een netevenwicht moeilijker maakt, bestaat een eerste benadering waarbij gebruik wordt gemaakt van marktinteracties erin de toekomstige elektriciteitsprijzen en de productie van hernieuwbare energie nauwkeurig te voorspellen. Nauwkeurige en betrouwbare voorspellingen resulteren immers in betere beslissingen, meer winst en minder onzekerheid. Dat vertaalt zich op zijn beurt in een net dat gemakkelijker in evenwicht kan worden gehouden en aantrekkelijkere economische stimulansen voor de integratie van HEB's. Met dit in het achterhoofd wordt in het eerste deel van het proefschrift ingezoomd op voorspellingsmethoden, die worden belicht in drie onderzoeksdomeinen. Ten eerste analyseren we, geprikkeld door het nieuwe marktbeleid van de EU dat gericht is op het tot stand brengen van één enkele en uniforme elektriciteitsmarkt in Europa, het effect van marktintegratie op de dynamiek van de elektriciteitsprijzen en stellen we nieuwe voorspellingsmodellen voor die uitgaan van marktintegratie om de nauwkeurigheid van de voorspellingen te verfijnen. Ten tweede, in het kielzog van de vooruitgang van deep learning (DL)-methoden in verschillende domeinen, onderzoeken we de toepassing van DL-methoden voor het voorspellen van elektriciteitsprijzen en ontwikkelen we nieuwe DL-voorspellingstechnieken die uiterst geavanceerde resultaten

opleveren. Ten derde, aangezien de voorspelling van de zonnestraling op korte termijn voor veel toepassingen van cruciaal belang is geworden, stellen we een algemeen model voor de voorspelling van de zonnestraling op korte termijn voor, dat de zonnestraling op elke locatie kan voorspellen zonder dat er grondmetingen nodig zijn. Deze nieuwe methode is van essentieel belang omdat zonnegeneratoren geografisch verspreid zijn en grondmetingen niet altijd gemakkelijk te bekomen zijn.

Het verbeteren van de nauwkeurigheid van voorspellingstechnieken is een indirecte benadering om de elektriciteitshandel minder onzeker te maken en de integratie van hernieuwbare energiebronnen te stimuleren. Een directere benadering is het toepassen van energieopslagsystemen om onevenwichten in het net op te vangen. In die context, hoewel energieopslag op lange termijn aantoonbaar een van de belangrijkste elementen is voor het welslagen van de energietransitie, zijn de meeste bestaande technologieën economisch gezien alleen efficiënt voor energieopslag op korte en middellange termijn. Daarom onderzoeken we in het tweede deel van dit proefschrift modellerings- en controletechnieken die ervoor zorgen dat seizoensopslagsystemen een maximale winst kunnen realiseren en onevenwichten in het net worden afgevlakt. Allereerst, omdat de bestaande thermische seizoensopslagmodellen te complex zijn en niet efficiënt kunnen worden overgezet op controle- en optimalisatieproblemen, stellen we een nieuw nauwkeurig model voor dat integreerbaar is in realtime controle- en optimalisatietoepassingen. Ten tweede stellen we controlealgoritmen voor seizoensopslagsystemen voor die, door expliciet gebruik te maken van de relatie tussen onevenwichten en prijzen, de onevenwichten in het net verkleinen en tegelijkertijd de winsten maximaliseren. Die algoritmen zijn op zichzelf al nieuw, aangezien de controlealgoritmen die momenteel beschikbaar zijn voor marktinteractie beperkt zijn tot de korte termijn en niet geschikt zijn voor seizoensopslagsystemen.

Een directere benadering om de integratie van hernieuwbare energiebronnen te stimuleren en het evenwicht in het net te bewaren, is om expliciet de structuur van de elektriciteitsmarkten te wijzigen, zodat meer energiesystemen economische stimulansen krijgen om de onevenwichten in het net op te vangen. In het bijzonder, naarmate de traditionele elektriciteitscentrales van het net worden losgekoppeld, wordt het duidelijk dat HEB-systemen moeten bijdragen aan de balancering van het net, wil het net stabiel blijven. Hoewel sommige HEB-systemen mogelijk kunnen bijdragen aan netbalancering, worden ze niet voor dit doel gebruikt vanwege de huidige regels die bij het balanceren van de systemen worden toegepast. Voorbeelden van dergelijke systemen zijn fotovoltaïsche zonne-energie-installaties, opslagsystemen zoals seizoensopslag, of zelfs - in sommige landen - windparken. Vanuit die motivatie onderzoeken we in het derde deel van dit proefschrift methoden en nieuwe marktstructuren die niet alleen deze systemen in staat stellen om bij te dragen aan de balancering van het net, maar ook economische prikkels geven om dat te doen. We tekenen bovendien een gedetailleerd nieuw marktkader uit voor het leveren van balanceringsdiensten via het verrekenen van onevenwichten. Het nieuwe kader neemt nieuwere systemen op in het aanbod van leveranciers van balanceringsdiensten en geeft deze systemen economische stimulansen om het net in evenwicht te brengen. Als bijkomend voordeel stimuleert het ook het gebruik van langetermijnopslagsystemen, die, zoals eerder aangevoerd, een sleutelrol spelen in de energietransitie.

CONTENTS

ABBREVIATIONS

AR	autoregressive
ARIMA	autoregressive integrated moving average
ARMA	AR with moving average terms
ARMAX	ARX with moving average terms
ARX	autoregressive with exogenous inputs
AWE	airborne wind energy
BFGS	Broyden–Fletcher–Goldfarb–Shanno
CNN	convolutional neural network
DL	deep learning
DM	Diebold-Mariano
DNI	direct normal irradiance
DNN	deep neural network
DR	dynamic regression
DSARIMA	double seasonal ARIMA
DSHW	double seasonal Holt-Winter
ECMWF	European Center for Medium-Range Weather Forecasts
EPEX	European Power Exchange
EXAA	Energy Exchange Austria
fARX	full-ARX
fARX-EN	fARX regularized with an elastic net
fARX-Lasso	fARX regularized with Lasso
GARCH	generalized autoregressive conditional heteroscedasticity
GBT	gradient boosting tree
GHI	global horizontal irradiance
GRU	gated recurrent unit
IHMARX	Hsieh-Manski ARX
ISP	imbalance settlement period
KNMI	Royal Netherlands Meteorological Institute (Koninklijk Nederlands Meteorologisch Instituut)
LSTM	long-short term memory
MA	moving average
MAE	mean absolute error
MAPE	mean absolute percentage error
MBE	mean bias error

MCMC	Markov chain Monte Carlo
MLP	multilayer perceptron
MPC	model predictive control
NWP	numerical weather prediction
OCP	optimal control problem
ODE	ordinary differential equation
PDE	partial differential equation
PSO	particle swarm optimization
RBF	radial basis function
ReLU	rectifier linear unit
RES	renewable energy source
RF	random forest
RL	reinforcement learning
RNN	recurrent neural network
rRMSE	relative root mean square error
SEVIRI	spinning enhanced visible and infrared imager
SICSS	surface insolation under clear and cloudy skies
sMAPE	symmetric mean absolute percentage error
SMBO	sequential model-based optimization
SNARX	smoothed nonparametric ARX
SOM	self-organizing maps
SOM-SVR	SVR with self-organizing maps
STESS	seasonal thermal energy storage systems
SVR	support vector regressor
tanh	hyperbolic tangent
TARX	threshold ARX
TBATS	exponential smoothing state space model with Box-Cox transformation, ARMA errors, trend and seasonal components
TES	thermal energy storage
TF	transfer function
TPE	tree-structured Parzen estimator
TSO	transmission system operator
WARIMA	wavelet-ARIMA
XGB	extreme gradient boosting

1

INTRODUCTION

They say it is the first step that costs the effort. I do not find it so.
I am sure I could write unlimited 'first chapters'. I have indeed written many.

John R. R. Tolkien

There is no question that climate change is happening; the only arguable point is what part humans are playing in it.

David Attenborough

The aim of this dissertation is to develop modeling, control, and forecasting approaches that can, by optimally trading in electricity markets, reduce grid imbalances while increasing the profits and integration of renewable energy sources. This introductory chapter starts by presenting the motivation behind this research, including the win-win relations between market actors and grid balancing, the importance of seasonal storage systems, and the necessity of accurate forecasting techniques. Then, it briefly outlines the main contributions of the dissertation, and concludes by explaining the structure and organization of the thesis.

1

1.1. MOTIVATION OF THE RESEARCH

As one of the many actions to reduce carbon emissions and mitigate the effects of climate change, the so-called energy transition [216] aims at dramatically increasing the share of *renewable energy sources (RESs)* in the energy mix. However, before achieving the desired goal of nearly 100% RES generation, there are several problems that need to be addressed. In particular, unlike most commodities, electricity networks require constant balancing between generation and consumption. However, as electricity supply and demand are uncertain, grid imbalances are unavoidable and the *transmission system operator (TSO)* has to correct them in real time. In this context, due to the weather dependence of RES generation, the increasing integration of RESs leads to more uncertain electricity generation and larger grid imbalances. Consequently, balancing the grid becomes more challenging and the TSO faces the risk of not being able to guarantee the grid safety.

Besides larger grid imbalances, another problem with the integration of RESs is that it leads to more complex and volatile electricity markets. Not only does this further complicate grid balancing, but it also limits the profits of RESs. In detail, when compared with other commodities, electricity trade displays some attributes that are quite uncommon, e.g. constant balance between production and consumption [211] or load and generation that are influenced by external weather conditions [243]. Due to these characteristics, the dynamics of electricity prices are in general very complex, e.g. highly volatile prices with sudden and unexpected price peaks [243]. In this context, as the penetration of RESs increases, so does the volatility in electricity prices [16, 32] and the complexity of price dynamics. Consequently, the behavior of market agents becomes more unpredictable, the imbalances become even larger, and the profits of RESs become more limited[1].

In short, as we approach the 100% RES generation target, the grid becomes harder to balance and control, electricity markets become more complex, and the profitability of RESs is hindered. In this context, while RES integration does complicate grid balance and increase price volatility, it also creates new market opportunities that can help to solve the mentioned problems. As an example, during periods of positive imbalances, i.e. when electricity generation is larger than consumption, prices are low; thus, market agents have economic incentives to wait for these periods to buy their energy. By doing so, they would not only reduce grid imbalances but they would also increase their own profits. Similarly, as prices are high during periods of negative imbalances (generation smaller than consumption), market agents have economic incentives to sell more energy and to reduce their consumption during those periods. As before, by doing so, they would reduce negative grid imbalances while increasing their own profits. Therefore, while the integration of RESs complicates the grid management and the electricity markets, it also creates win-win situations that allow flexible RESs to reduce grid imbalances and increase their profits.

In this framework, despite these seemingly obvious win-win opportunities, traditional research has usually considered mechanisms for grid balancing that disregard this

[1] As RESs complicate electricity trading, maximizing profits with such a technology is naturally more difficult than with traditional generators. In addition, since RESs also worsen the grid balance, existing policies of electricity markets limit the trading opportunities of RESs, e.g. in general RESs cannot participate in balancing markets.

relation between grid balance and electricity markets. Particularly, while existing balancing mechanisms provide economic incentives to regulate the grid, they usually do it through local market mechanisms based on demand response or through the use of balancing markets. Local market mechanisms are restricted to small consumers/producers and are not always economically optimal (as their prices are usually decoupled from the main wholesale markets). Moreover, balancing markets have very restrictive prequalification requirements and do not allow participation of several technologies, e.g. seasonal thermal storage or solar farms. Hence, new methods to exploit these win-win relations are needed so that the profit and incentives for RESs are maximized and the existing balancing mechanisms are improved.

1.2. RESEARCH GOALS AND MAIN CONTRIBUTIONS

THE aim of this PhD dissertation is to help to decrease the grid imbalances, to increase the profitability of renewable sources, and to ensure a larger integration of RESs by exploiting the relation between grid balancing and electricity markets. To do so, the thesis proposes new forecasting and control algorithms, dynamical models, and market policies that make use of these win-win opportunities. In detail, the thesis is divided in three parts with six different contributions:

- Part I - *Forecasting algorithms for electricity prices and renewable sources*

  In the first part of the thesis, we propose novel forecasting algorithms that help to reduce the uncertainty of RESs and to increase their profitability. In detail, as accurate forecasts lead to better decision making, higher economic profits, and lower uncertainty, a first approach to exploit market interaction is to accurately predict future electricity prices and generation of RESs. Particularly, besides larger profits, better forecasts grant a more predictably behavior of RESs, which in turn leads to a grid that is less uncertain and easier to balance. Based on that, Part I of the thesis focuses on advancing the field of forecasting technique via three distinct contributions:

  (a) **Forecasting algorithms for exploiting market integration:** As electricity markets around the world increase their level of integration, the dynamics of electricity prices are expected to depend on market integration. However, methods for forecasting electricity prices have traditionally disregarded this issue. To address this, we propose two forecasting techniques that exploit market interaction. The first technique is based on using input features representing market integration and the second one on modeling market integration in the output of the model. Using a case study, we show that the new approaches are key to improve the accuracy of forecasting models.

  (b) **Deep learning algorithms for electricity prices:** Motivated by the recent success of *deep learning (DL)* in several energy applications, we investigate its application in the context of electricity price forecasting. Particularly, we develop four forecasting models based on deep neural networks: one based on a feedforward network, two based on recurrent networks, and a fourth one based on a convolutional neural network. Then, using a case study, we show

that the new DL forecasting techniques lead to statistically significant improvements over the state-of-the-art methods from the literature.

(c) **A generalized model for solar irradiance forecasting:** As the amount of solar power in the energy mix increases, forecasting solar irradiance is becoming key in many applications. In this context, as solar generators are geographically dispersed and ground measurements are not easy to obtain, predicting solar irradiance without the need of local data is particularly important. To tackle that, we propose a novel time-series model that can, for the first time, forecast solar irradiance in any general location without the need of ground measurements. As we show, the method is not only location-independent, but it is also more accurate than existing forecasting techniques.

- Part II - *Modeling and control of seasonal storage to reduce grid imbalances*

  In this part of the thesis, we investigate approaches to reduce grid imbalances based on seasonal storage. Particularly, forecasting techniques are indirect methods to increase the profitability and integration of RESs and to improve the grid balance. A more direct approach is to use energy storage to provide energy flexibility and to absorb grid imbalances. In detail, by simply exploiting the win-win opportunities between market agents and grid balance, energy storage systems can use their flexibility to maximize their profits and to reduce grid imbalances. In this context, to absorb the seasonal fluctuations of RESs, long-term (seasonal) energy storage is arguably the most important storage technology. Based on that, Part II of the thesis improves the fields of seasonal storage and grid balancing with two distinct contributions:

  (d) **Modeling thermally stratified seasonal storage:** One of the most widely used seasonal storage technologies are thermally stratified storage tanks. Yet, despite their importance, existing models for such systems cannot be employed in control and optimization problems as they are too complex. To tackle that, we propose a new accurate model that can be integrated in real-time control and optimization applications. The main feature of the proposed model is that, unlike the existing models from the literature, it can model the buoyancy in the tank using a 1D smooth and continuous model.

  (e) **Control algorithms for seasonal storage:** Market trading with seasonal storage is key to reduce grid imbalances via profit maximization. Yet, existing control algorithms for seasonal storage do not consider market interaction. To address this, we propose control algorithms that exploit the relation between imbalances and prices in order to reduce grid imbalances and maximize profits.

- Part III - *New market framework for grid balancing support*

  In the third part of the thesis, we investigate an even more direct approach to incentivize the integration of renewables and to keep the grid balanced. In particular, we explore possible modifications in the structure of electricity markets so that

more energy systems contribute to grid balancing. In detail, as traditional power plants are taken off the grid, RESs need to participate in grid balancing to ensure grid stability. Yet, due to the current rules applied to grid balancing, RESs are generally[2] not used for this purpose. To tackle that, new market structures are needed so that RESs do not only participate in balancing the grid, but they also have economic incentives to do so. Based on that, Part III of the thesis contributes with a novel market framework:

(f) **New market framework for grid balancing support:** To address the described problem, we propose a new market framework for providing balancing services by trading with the imbalances. The new framework incorporates more systems (including RESs) into the portfolio of balancing providers, give these systems economic incentives to balance the grid, and reduces the cost of balancing the grid.

1.3. STRUCTURE OF THE THESIS

THE organization of the thesis is illustrated in Figure 1.1. The thesis has been written in three main parts, where each part starts with a background chapter that provides the foundations for the main contributing chapters. Part I of the thesis includes Chapters 2–5 and studies forecasting approaches that reduce RES uncertainty and increase the profitability of RESs. Part II comprises Chapters 6–8 and explores modeling and control algorithms for seasonal storage that reduce grid imbalances. Part III comprises Chapters 9–10 and analyzes new market structures to increase the number and types of systems providing balancing services. The contents of each contributing chapter can be summarized as follows:

CHAPTER 3

The chapter studies the problem of market integration and how that affects to forecasting accuracy. Motivated by the increasing integration among electricity markets, the chapter proposes a methodology to study the importance of market integration and derives new forecasting techniques that exploit this effect. The new approaches are shown to improve the accuracy of existing forecasting techniques.

The contents of this chapter have been published in:

[137] J. Lago, F. De Ridder, P. Vrancx and B. De Schutter. 'Forecasting day-ahead electricity prices in Europe: The importance of considering market integration'. *Applied Energy* 211, 890–903, 2018. DOI: 10.1016/j.apenergy.2017.11.098.

CHAPTER 4

Driven by the success of DL in other fields, this chapter studies the applicability of DL in the context of electricity price forecasting. In particular, four DL forecasting approaches are proposed and then compared with an extensive benchmark of methods from the literature. Based on the results of Chapter 3, the proposed DL forecasting techniques

[2] In some countries wind power is used for grid balancing.

1

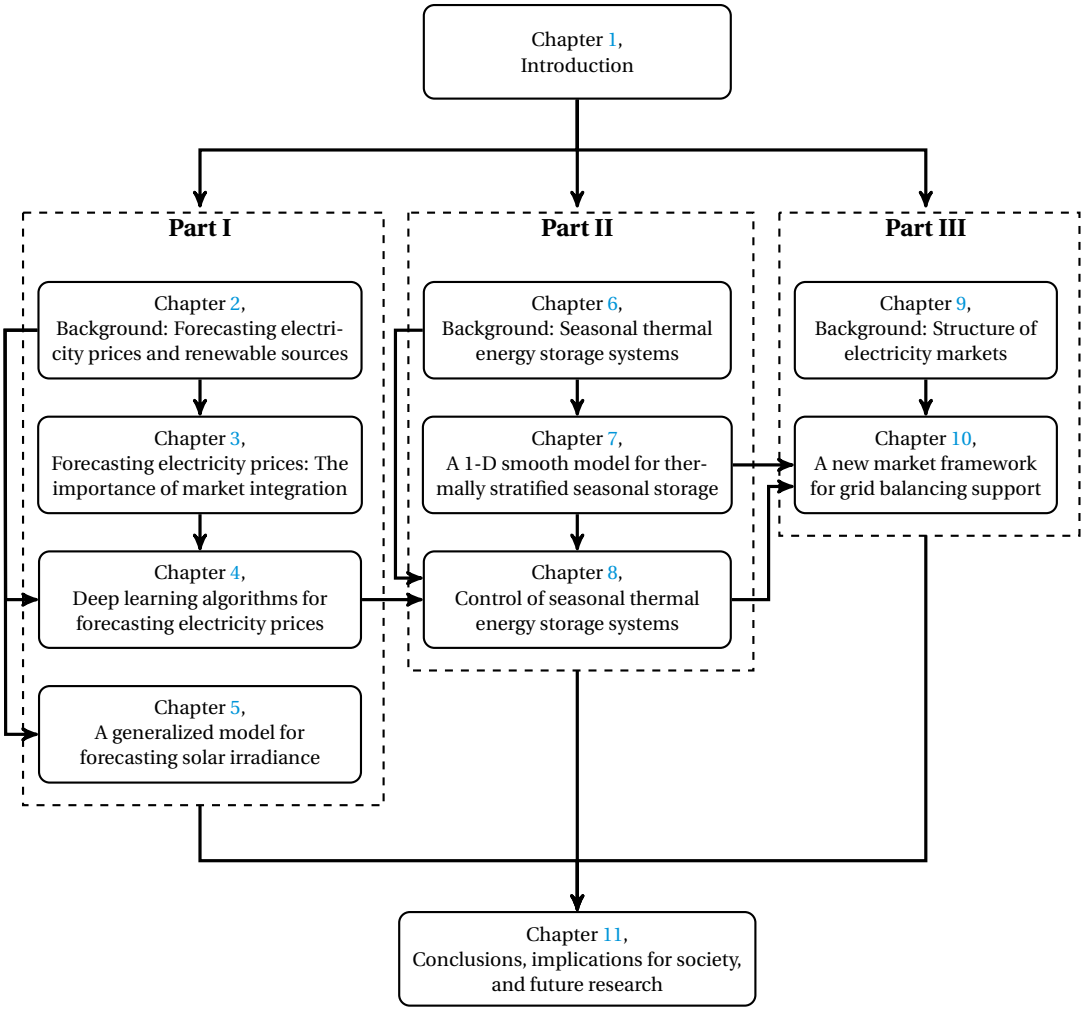


Figure 1.1: Structure of the thesis

are designed to exploit market integration. Using a case study, the proposed techniques are shown to forecast statistically significantly better than the literature methods. As a minor contribution, the chapter also provides the largest existing benchmark for electricity price forecasting.

Part of this chapter has been published in:

[135] J. Lago, F. De Ridder and B. De Schutter. 'Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms'. *Applied Energy* 221, 386–405, 2018. DOI: 10.1016/j.apenergy.2018.02.069.

CHAPTER 5

As solar generators are geographically dispersed and obtaining local data is not easy, this chapter proposes a generalized forecasting method for solar irradiance that can forecast in any location without the need of local data. Using a case study it is shown that, the method does not only avoid using local telemetry, but it is also more accurate than existing forecasting techniques.

The contents of this chapter have been published in:

[133] J. Lago, K. De Brabandere, F. De Ridder and B. De Schutter. 'Short-term forecasting of solar irradiance without local telemetry: A generalized model using satellite data'. *Solar Energy* 173, 566–577, 2018. DOI: 10.1016/j.solener.2018.07.050.

[132] J. Lago, K. De Brabandere, F. De Ridder and B. De Schutter. 'A generalized model for short-term forecasting of solar irradiance'. In: *Proceedings of the 2018 IEEE Conference on Decision and Control*, 3165–3170, 2018. DOI: 10.1109/cdc.2018.8618693.

CHAPTER 7

This chapter develops a new dynamical model for thermally stratified storage tanks. Particularly, despite being one of the most important seasonal storage systems, the existing models for thermally stratified storage tanks are too complex and cannot be used in control and optimization problems. This chapter tackles this issue by proposing a more efficient dynamical model that can be used in real-time applications and optimization problems

Part of this chapter has been published in:

[136] J. Lago, F. De Ridder, W. Mazairac and B. De Schutter. 'A 1-dimensional continuous and smooth model for thermally stratified storage tanks including mixing and buoyancy'. *Applied Energy* 248, 640–655, 2019. DOI: 10.1016/j.apenergy.2019.04.139.

CHAPTER 8

This chapter proposes control approaches for seasonal storage systems that are able to reduce grid imbalances. Particularly, exploiting the relation between prices and imbalances, the chapter proposes control algorithms that perform optimal energy trading and

keep the grid balanced. To do so, the algorithms make use of the model developed in Chapter 7 as well as the forecasting methods proposed in Chapter 4. Based on a case study, the control approaches are shown to reduce grid imbalances while increasing the profit of seasonal storage systems.

The contents of this chapter have been published in:

[142] J. Lago, G. Suryanarayana, E. Sogancioglu and B. De Schutter. 'Optimal control strategies for seasonal thermal energy storage systems with market interaction'. *IEEE Transactions on Control Systems Technology*, Early Access, 2020. DOI: 10.1109/TCST.2020.3016077.

[141] J. Lago, E. Sogancioglu, G. Suryanarayana, F. De Ridder and B. De Schutter. 'Building day-ahead bidding functions for seasonal storage systems: A reinforcement learning approach'. In: *Proceedings of the IFAC Workshop on Control of Smart Grid and Renewable Energy Systems*, 488–493, 2019. DOI: 10.1016/j.ifacol.2019.08.258.

CHAPTER 10

Motivated by the need of balancing the grid via RESs, this chapter proposes a new market framework that grants RESs access to grid balancing and provides RESs with economic incentives to do so. The core idea behind the new market framework is to trade electricity using the imbalance settlement mechanism. To test the framework, the model and control approaches proposes in Chapters 7 and 8 are employed. The new framework does not only increase the portfolio of balancing providers, but it also gives economic incentives to balance the grid and reduces the balancing costs.

Part of this chapter has been submitted for publication to Renewable & Sustainable Energy Reviews as:

[140] J. Lago, K. Poplavskaya, G. Suryanarayana and B. De Schutter. 'A market framework for grid balancing support through imbalances trading'. *Renewable and Sustainable Energy Reviews*, Under review.

CHAPTER 11

Finally, this chapter concludes the thesis with the main contributions, the implications that this research has for society, and the directions for future research.

1.4. ADDITIONAL WORK

As part of this PhD research, besides the publications [132, 133, 135–137, 140–142] that are used as basis for this thesis, we have authored other publications [59, 60, 138, 139, 159, 184, 185, 189, 190, 204, 205, 220, 221]. For the sake of simplicity, these publications are left out of the dissertation; however, it is worth summarizing their contributions here as they also provide solutions for the energy transition and the integration of RESs.

While the field of price forecasting has benefited from plenty of contributions in the last decades, it arguably lacks a rigorous approach when comparing and evaluating new

predictive algorithms. With that motivation, in [139] we have discussed and analyzed best practices for forecasting electricity prices, compared state-of-the-art algorithms in the same context, and proposed a large benchmark of data and models so that researches can use them to compare and evaluate new algorithms.

In general, the energy extracted by an *Airborne wind energy (AWE)* system depends on the wind velocity and direction. As these two properties vary within seconds, controllers need to generate flying trajectories online. This is a problem for existing *model predictive control (MPC)* approaches since to generate trajectories they need to solve complex nonlinear optimization problems. In [138], we have proposed a novel MPC algorithm, i.e. warping MPC, that solves this issue and is able to generate optimal trajectories in real-time at a negligible cost.

In [204] and [205] we have developed methods for detecting faults in low-voltage distribution grids. The proposed algorithms solve some of the drawbacks of the existing state-of-the-art methods, namely: (i) to be independent from the grid topology; (ii) to localize faults even with limited data; and (iii) to detect and localize high-impedance faults. Using a case study, the proposed methods are shown to outperform the state-of-the-art methods from the literature.

In [184, 185], we have proposed a *stochastic MPC (SMPC)* algorithm for the thermal control of buildings. Unlike the existing approaches from the literature, the proposed approach can control the building using realistic nonlinear dynamics and taking into account the stochasticity of the building disturbances. Since the heating and cooling necessities correspond to nearly 80 % of the total energy consumption in European households, the proposed controller helps to obtain buildings that are more energy efficient.

Urban building energy models are needed to assess the impact of energy policies and to quantify the building energy use at district and city level. However, due to the lack of data, existing buildings models are usually based on archetypes, and hence simulations of building energy use are highly uncertain. In [59, 60], we have proposed a probabilistic building characterization method that mitigates this issue by accurately modeling the variability of the existing residential building stock.

Forecasting the heat load demand at the district level is key for well-managed district heating networks. In particular, accurate forecasting is needed in order to have better decision making, increase the economic profits, and lower the uncertainty in heating networks. In this context, in [221] we have proposed two novel methods for heat load forecasting that improve upon the state-of-the-art methods from the literature.

To address some of the existing market inefficiencies, balancing markets in the European Union are undergoing substantial regulatory changes. In [189, 190], we have studied and analyzed the potential effects of these changes using an agent-based model. To the best of our knowledge, this is the first attempt to model a standalone balancing energy market with different pricing rules and the first model-based study of the upcoming balancing market changes.

In [220], we have proposed a data-driven methodology to identify the optimal placement of sensors in multi-zone buildings. The proposed methodology is based on statistical tests that study the independence of sensors. Its main advantages are that it does not need to rely on the underlying building model and that it is quick and expert-free. As we show in [220], the proposed approach is not only efficient and expert-free but it is

also as accurate as traditional methods.

In the context of electricity price forecasting, neural networks are among the most popular models. However, the structure of the neural networks proposed in the literature differs greatly, i.e. it is not always clear which structures are more suitable for forecasting electricity prices. In [159], we have tried to shed some light over this issue by studying the performance of the two most common structures: neural networks that focus on each hour of the day separately versus neural networks that model the vector of prices in a single model. Using several case studies, we answer this question and show that neural networks that model prices in a single model are significantly more accurate.

1.5. MATHEMATICAL NOTATION

In this dissertation, we use lowercase letters and symbols to represent scalar variables, e.g. x or θ, bold lowercase letters and symbols to represent vectors, e.g. $\mathbf{x}$ or $\boldsymbol{\theta}$, and bold capital letters for matrices, e.g. $\mathbf{X}$. However, to stay aligned with the standard notation in the literature, scalar values are sometimes also defined using capital letters: we use N for dataset sizes and the horizon of predictive controllers; I for solar irradiance; T for the temperature; $\dot{Q}$ for thermal power; and P, A, and V respectively for perimeter, area, and volume.

Sets are denoted by capital letters in their calligraphic forms, e.g. $\mathcal{X}$, or Greek capital symbols, e.g. Θ. Scalar functions are denoted by both lowercase and capital letters, e.g. $F(x)$ and $f(x)$, and vector functions by the corresponding bold letters, e.g. $\mathbf{F}(x)$ and $\mathbf{f}(x)$. The expected value $\mathbb{E}$ is defined using the capital letter E in its blackboard form.

We use $\mathbb{R}$ to denote the set of real numbers, $\mathbb{Z}$ for the set of integers, and $\mathbb{N}$ for the set of natural numbers including zero. Moreover, we define by $\mathbb{R}^n$ the set of real-valued vectors of dimension n, by $\mathbb{R}^{n\times m}$ the set of real-valued matrices with n rows and m columns, and we assume all vectors to be column vectors. To represent vectors and matrices in their elementwise forms we employ square brackets, e.g. $\mathbf{x} = [x_1, \dots, x_n]^\top$. However, for notational simplicity, concatenations of several vectors, e.g. $[\mathbf{x}^\top, \mathbf{y}^\top]^\top$, are shortened as $(\mathbf{x}, \mathbf{y})$.

Continuous-time indices are denoted by t and discrete ones by k, e.g. x_t and x_k respectively represent the variable x at time t and time step k. Indexed variables represent their indices using subscripts or superscripts, e.g. x_k, and multiple indices are separated by commas, e.g. $x_{k,i}$. In the case where the index represents a label and not variable we employ roman fonts, e.g. x_{in} where the subscript in stands for input.

Forecasted and estimated values are indicated with a hat (circumflex accent), e.g. $\hat{x}$, and fixed values are defined using a bar (macron accent), e.g. $\bar{x}$.

I

FORECASTING METHODS TO INCENTIVIZE RENEWABLE SOURCES

2

Background: Forecasting electricity prices and renewable sources

The most exciting phrase to hear in science, the one that heralds the most discoveries, is not "Eureka!" (I found it!) but "That's funny...".

Isaac Asimov

Prediction is very difficult, especially about the future.

Niels Bohr

This chapter provides background knowledge on the field of forecasting in the energy domain and on different algorithms and concepts used in the first part of the thesis. The chapter starts with a brief motivation on the research in forecasting methods together with a brief literature survey of the relevant topics, i.e. electricity price and irradiance forecasting, electricity market integration, feature selection, and deep learning. Next, different methodologies to evaluate forecasting algorithms are described, including statistical tests for time series forecasting. Then, the structure of the wholesale electricity markets is defined and the importance of the day-ahead market is explained. Finally, a brief introduction into the field of deep learning is provided together with a description of hyperparameter optimization and analysis.

2.1. INTRODUCTION

DUE to the intermittent and unpredictable nature of *renewable energy sources (RESs)*, the increasing integration of RESs into the electrical grid makes the grid harder to balance and to manage and hinders the economic profits of RESs [144, 236]. In this context, a possible solution to mitigate this issue is to employ accurate forecasts of electricity prices and RES generation since accurate forecasts lead to better decision making, higher economic profits, and lower uncertainty. In turn, these benefits translate into larger economic incentives for integration of RESs and a grid that is easier to balance. With that motivation, in this thesis we investigate new forecasting methods that can: (i) improve upon the state-of-the-art forecasting solutions; (ii) reduce the uncertainty in RES generation; (iii) increase the profits of RESs.

Due to the importance of electricity prices and solar irradiance forecasts, the focus of the thesis is on forecasting methods for these two variables. Particularly, as a result of the liberalization and deregulation of the electricity markets in the last two decades, the dynamics of the electricity trade have been completely reshaped. In this context, electricity has become a commodity that displays some characteristics that are uncommon to other markets, e.g. sudden and unexpected price peaks and seasonality of prices at three different levels (daily, weekly, and yearly) [243]. In recent years, with the increasing penetration of RESs, the described behavior has aggravated: as the penetration of RESs increases, so does the dependence of electricity production on weather conditions and, in turn, the volatility in electricity prices [16, 32, 88, 165]. Due to this effect, as the increasing integration of RESs increases the volatility of prices, the behavior of market agents becomes naturally more unpredictable, the imbalances between production and consumption increase, and the electrical grid becomes more unstable. To tackle these problems, electricity price forecasting has become a central point of research in the energy sector. In particular, by improving the forecasting accuracy, the negative effects of price uncertainty can be mitigated, the grid can be stabilized, and economic profits can be increased.

Similarly, although all different types of RES generation are difficult to forecast, solar energy is arguably one of the most challenging ones. Particularly, as solar energy is one of the most unpredictable renewable sources, the increasing use of solar power in recent years has led to an increasing interest in forecasting irradiance over short time horizons. In detail, short-term forecasts of solar irradiance are paramount for activation of energy reserves to manage the grid stability, operational planning, switching sources, programming backup, short-term power trading, peak load matching, scheduling of power systems, congestion management, and cost reduction [97, 196, 236].

In this context, as introductory material, this chapter provides background knowledge on the field of forecasting on the energy domain and on other concepts explored in this part of the thesis. In detail, the remainder of the chapter is organized as follows: Section 2.2 performs a brief literature review of the relevant research topics. Then, Section 2.3 introduces the methods and metrics to evaluate forecasting techniques. Section 2.4 provides a summary of the structure of electricity markets. Finally, Section 2.5 introduces the field of deep learning and Section 2.6 the field of hyperparameter optimization.

2.2. LITERATURE SURVEY

IN this section, we present a brief literature review of the five topics that are relevant for this part of the thesis: electricity price forecasting, electricity market integration, feature selection for electricity prices, solar irradiance forecasting, and deep learning for energy applications.

2.2.1. ELECTRICITY PRICE FORECASTING

The price forecasting literature is typically divided into five areas: (i) multi-agent or game theory models simulating the operation of market agents, (ii) fundamental methods employing physical and economic factors, (iii) reduced-form models using statistical properties of electricity trade, (iv) statistical models comprising time series and econometric models, and (v) artificial intelligence methods [243]. For forecasting day-ahead prices, or in general any other type of electricity spot prices, statistical and artificial intelligence methods have showed to yield the best results [243] and are thus the main focus of this review.

Typical statistical methods are: *autoregressive (AR)* models [244], AR models with exogenous inputs [244], AR models with moving average terms [55, 252], dynamic regression models [171], transfer functions [171], double seasonal Holt-Winter models [56], threshold AR models [244], generalized AR conditional heteroscedasticity models [67], or semi/non-parametric models [244]. Besides pure statistical models, several hybrid methods have also been proposed, e.g. wavelet-based models [52, 225, 252]. Statistical models are usually linear forecasters, and as such, they are successful in the areas where the frequency of the data is low, e.g. for weekly patterns. However, for hourly values, the nonlinear behavior of the data might be too complicated to predict [5].

As a result, motivated by the need for forecasters that are able to predict the nonlinear behavior of hourly prices, several artificial intelligence methods have been proposed. Among these methods, artificial neural networks [39, 224, 239, 245], support vector regressors [76], radial basis function networks [151], and fuzzy networks [6] are among the most commonly used. Despite the large number of studies, the results comparing the accuracy of the mentioned models have however produced unclear conclusions [39]. In general, the effectiveness of each model seems to depend on the market under study and on the period considered.

2.2.2. MARKET INTEGRATION IN ELECTRICITY PRICES

In the last decades, the EU has passed several laws trying to achieve a single and integrated European electricity market [116, 187]. At the moment, even though a single market is far from existing, there is evidence suggesting that the level of integration across the different regional markets has been increasing over time [34]. In particular, evidence suggests that in the case of Belgium and France, the spot prices share strong common dynamics [164]. While some researchers have evaluated the level of integration of the European markets [34, 164, 256], and others have proposed statistical models to evaluate the probability of spike transmissions across EU markets [152], the literature regarding market integration to improve forecasting accuracy is rather scarce. To the best of our knowledge, only two other works have taken into account some sort of market integration, namely [258] and [175].

In particular, [258] analyzes the effect of using the day-ahead prices of the *energy exchange Austria (EXAA)* on a given day to forecast the prices of other European markets on the same day. Using the fact that for the EXAA market the clearing prices are released before the closure of other European markets, [258] models the price dynamics of several European markets and considers the EXAA prices of the same day as part of these models. It is shown that, for certain European markets, using the available prices from the EXAA improves the forecasting accuracy in a statistically significant manner.

Similarly, [175] considers external price forecasts from other European markets as exogenous inputs of an artificial neural network to predict Italian day-ahead prices. Then, [175] shows that using the given forecasts the mean absolute percentage error of the neural network is reduced.

2.2.3. FEATURE SELECTION FOR ELECTRICITY PRICES

Feature selection is defined as the process to select, for a given model, the subset of important and relevant input variables, i.e. features. Typically, three families of methods to perform feature selection exist: filter, wrapper, and embedded methods [95]. Filter methods apply some statistical measure to assess the importance of features [38]. Their main disadvantage is that, as the specific model performance is not evaluated and the relations between features are not considered, they may select redundant information or avoid selecting some important features. Their main advantage is that, as a model does not have to be estimated, they are very fast. By contrast, wrapper methods perform a search across several feature sets, evaluating the performance of a given set by first estimating the prediction model and then using the predictive accuracy of the model as the performance measure of the set [38]. Their main advantage is that they consider a more realistic evaluation of the performance and interrelations of the features; their drawback is a long computation time. Finally, embedded methods, e.g. regularization [86], learn the feature selection at the same time the model is estimated. Their advantage is that, despite being less computationally expensive than wrapper methods, they still consider the underlying model. However, as a drawback, they are specific to a learning algorithm, and thus, they cannot always be applied.

Approaches for feature selection in the electricity price forecasting literature vary according to the prediction model used. For time series methods using only prices, e.g. ARIMA, autocorrelation plots [52] or the Akaike information criterion [41] have been commonly used. In the case of forecasters with explanatory variables, e.g. neural networks, most researchers have used trial and error or filter methods based on linear analysis techniques: statistical sensitivity analysis [56, 224], correlation analysis [198], or principal component analysis [106]. Since prices display nonlinear dynamics, the mentioned techniques might be limited [9]; to address this, nonlinear filter methods such as the relief algorithm [7] or techniques based on mutual information [9, 83, 123] have been proposed. More recently, a hybrid nonlinear filter-wrapper method, which uses mutual information and information content as a first filter step and a real-coded genetic algorithm as a second wrapper step, has been proposed [2].

2.2.4. SOLAR IRRADIANCE FORECASTING

The forecasting of solar irradiance can be typically divided between methods for *global horizontal irradiance (GHI)* and methods for *direct normal irradiance (DNI)* [146], with the latter being a component of the GHI (together with the diffuse solar irradiance). In this thesis, we focus on forecasting GHI as it is the variable that determines the power of photovoltaic panels. For the case of GHI, forecasting techniques can be categorized into two subfields according to the input data and the forecast horizon [65, 236]:

1. Time series models based on satellite images, measurements on the ground level, or sky images. These methods are usually suitable for short-term forecasts up to 4-6 hours. Within this field, the literature can be further divided into three groups.
   (a) Classical statistical models like ARMA models [3], ARIMA models [196], the CARDS model [109], or the Lasso model [251].
   (b) Artificial intelligence models such as neural networks models [145, 163], support vector machines [145], decision trees-based models [162], or Gaussian models [145].
   (c) Cloud-moving vector models that use satellite images [155].
2. Numerical weather prediction models that simulate weather conditions. These methods are suitable for longer forecast horizons, i.e. horizons beyond 4-6 hours, time scales where they outperform the statistical models [179]. As the research of this thesis focuses on short-term forecasts, a complete review of numerical weather prediction methods is not provided. Instead, we refer to [65] for further details.

While the division in accuracy between numerical weather prediction and time series models is given by the predictive horizon, establishing comparisons between time series models is more complex. In particular, while some authors have reported the superiority of statistical models over artificial intelligence methods [196], others have obtained opposite results [209].

2.2.5. DEEP LEARNING FOR ENERGY APPLICATIONS

In the last decade, the field of neural networks has experienced several innovations that have lead to what is known as *deep learning (DL)* (see Section 2.5). In particular, one of the traditional issues of neural networks had always been the large computational cost of training large models. However, that changed completely when [102] showed that a deep belief network could be trained efficiently using an algorithm called greedy layer-wise pretraining. As related developments followed, researchers started to be able to efficiently train complex neural networks whose depth was not just limited to a single hidden layer (as in the traditional *multilayer perceptron (MLP)*. As these new structures systemically showed better results and generalization capabilities, the field was renamed as deep learning to stress the importance of the depth in the achieved improvements [86].

While this success of DL models initiated in computer science applications, e.g. image recognition [131], speech recognition [103], or machine translation [15], the benefits

of DL have also spread in the last years to several energy-related applications [50, 75, 79, 130, 240, 241]. Among these areas, wind power forecasting is arguably the field that has benefited the most: [241] shows how, using a deep belief network and quantile regression, probabilistic forecasting of wind speed can be improved. Similar to [241], [79] proposes a deep feature selection algorithm that, in combination with a multi-model framework, improves the wind speed forecasting accuracy by 30%. In the same area of research, [240] proposes an ensemble of convolutional neural networks to obtain more accurate probability forecasts of wind power.

In addition to wind power applications, DL has also shown success in other energy-related fields. In the context of load forecasting, [75] proposes a deep autoencoder in combination with a gradient boosting tree model and shows how they forecast building cooling load more accurately than alternative techniques; within the same research paper, a deep neural network to accurately forecast building cooling load is also proposed. For a different application, [130] proposes a DL model to detect islanding and to distinguish this effect from grid disturbances; based on the obtained simulation results, [130] indicates that the DL model can detect islanding with a very high accuracy. In addition, [50] proposes a DL strategy for time series forecasting and shows how it can be used successfully to forecast electricity consumption in households.

2.3. EVALUATION OF FORECASTING ALGORITHMS

IN order to evaluate forecasting algorithms, their performance is usually analyzed by means of accuracy metrics. Moreover, in some applications, the use of statistical testing is also employed to ensure that differences in accuracy are statistically significant. In this section, we describe the standard evaluation procedures typically used in the context of forecasting electricity prices and solar irradiance.

2.3.1. ELECTRICITY PRICE FORECASTING

In the field of electricity price forecasting, the most common method to evaluate forecasts is to compare its performance with established methods by means of accuracy metrics. While not as common, the use of statistical testing is also often employed.

ACCURACY METRICS

The most widely used metrics to measure accuracy are the *mean absolute error (MAE)*, the *mean absolute percentage error (MAPE)*, and the *symmetric mean absolute percentage error (sMAPE)* [158]:

$$\text{MAE} = \frac{1}{N}\sum_{k=1}^{N}|y_k - \hat{y}_k|, \tag{2.1}$$

$$\text{MAPE} = \frac{1}{N}\sum_{k=1}^{N}\frac{|y_k - \hat{y}_k|}{|y_k|}, \tag{2.2}$$

$$\text{sMAPE} = \frac{1}{N}\sum_{k=1}^{N}2\frac{|y_k - \hat{y}_k|}{|y_k| + |\hat{y}_k|}, \tag{2.3}$$

where y_k and $\hat{y}_k$ respectively represent the real and forecasted values at times k, and N is the size of the out-of-sample, i.e. test, dataset. Since absolute errors are hard to compare between different datasets, the MAE is not always very informative. Moreover, since MAPE values become very large with prices close to zero (regardless of the actual absolute errors), the MAPE is usually dominated by the periods of low prices and is also not very informative. For these reasons, for evaluating electricity price forecasting models in this thesis, the sMAPE metric is employed.

DIEBOLD-MARIANO TEST

The metrics defined above only provide an assessment of which model has a better accuracy. While the accuracy of a model can be higher, the difference in performance might be not significant enough to establish that the model is really better. To assess the statistical significance in the difference of predictive accuracy performance, a commonly used tool is the *Diebold-Mariano (DM)* test [66].

Given a time series vector $\mathbf{y} = [y_1, \dots, y_N]^\top$ to be forecasted, two forecast vectors $\hat{\mathbf{y}}_1$ and $\hat{\mathbf{y}}_2$ of $\mathbf{y}$, and the associated forecasting error vectors $\boldsymbol{\varepsilon}^{\hat{\mathbf{y}}_1} = [\varepsilon_1^{\hat{\mathbf{y}}_1}, \dots, \varepsilon_N^{\hat{\mathbf{y}}_1}]^\top$ and $\boldsymbol{\varepsilon}^{\hat{\mathbf{y}}_2} = [\varepsilon_1^{\hat{\mathbf{y}}_2}, \dots, \varepsilon_N^{\hat{\mathbf{y}}_2}]^\top$, the DM test evaluates whether there is a significant difference in performance accuracy based on an error loss function $L(\cdot)$. In particular, the DM test builds a loss differential function as:

$$d_k^{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2} = L(\varepsilon_k^{\hat{\mathbf{y}}_1}) - L(\varepsilon_k^{\hat{\mathbf{y}}_2}). \tag{2.4}$$

Then, it statistically tests the null hypothesis H_0 of both forecasts having equal accuracy, i.e. equal expected loss, against the alternative hypothesis H_1 of the forecasts having different accuracy, i.e.:

$$\begin{array}{c}\text{Two-sided}\\ \text{DM test}\end{array} \quad \begin{cases} H_0 : \mathbb{E}[d_k^{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2}] = 0, \\ H_1 : \mathbb{E}[d_k^{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2}] \neq 0. \end{cases} \tag{2.5}$$

To infer the p-value[1], the test approximates the expected value $\mathbb{E}[d_k^{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2}]$ as the average of the sequence of loss differentials:

$$\mathbb{E}[d_k^{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2}] \approx \bar{d}^{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2} = \frac{1}{N} \sum_{k=1}^{N} d_k^{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2}, \tag{2.6}$$

and then performs an asymptotic z-test[2]. For the details on the computation we refer to [66].

Similar to the standard two-sided DM test, a one-sided DM test can be built using the same statistic but testing the null hypothesis that the accuracy of $\hat{\mathbf{y}}_1$ is equal or worse than the accuracy of $\hat{\mathbf{y}}_2$ versus the alternative hypothesis of the accuracy of $\hat{\mathbf{y}}_1$ being better:

$$\begin{array}{c}\text{One-sided}\\ \text{DM test}\end{array} \quad \begin{cases} H_0 : \mathbb{E}[d_k^{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2}] \geq 0, \\ H_1 : \mathbb{E}[d_k^{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2}] < 0. \end{cases} \tag{2.7}$$

[1] The p-value represents the probability of observing a sequence of loss differentials $\{d_k^{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2}\}_{k=1}^N$ at least as extreme as the one observed, assuming that the null hypothesis is correct.

[2] A z-test is a statistical test that, under the null hypothesis H_0, approximates the distribution of the test statistic by a normal distribution.

While the loss function L can be freely chosen, it has to ensure that the resulting loss differential is covariance stationary. A loss function that is typically used is:

$$L(\varepsilon_k^{M_i}) = |\varepsilon_k^{M_i}|^p, \tag{2.8}$$

where usually $p \in \{1,2\}$.

2.3.2. SOLAR IRRADIANCE FORECASTING

In the context of solar irradiance forecasting, the most common method to evaluate forecasts is to analyze their accuracy by means of three different metrics: the *relative root mean square error (rRMSE)*, the *mean bias error (MBE)*, and the forecasting skill s [160]. Unlike with prices, the use of statistical testing is rarely employed.

The first metric that is often used is the rRMSE, which provides an assessment of the average spread of the forecasting errors. In particular, given a vector $\mathbf{y} = [y_1, \dots, y_N]^\top$ and its forecast $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_N]^\top$, the rRMSE is computed as:

$$\text{rRMSE} = \frac{\sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2}}{\frac{1}{N}\sum_{k=1}^{N} y_k} \cdot 100\%. \tag{2.9}$$

Besides analyzing the average spread of errors, forecasts of solar irradiance are also often analyzed by means of the MBE, a measure of the overall bias of the model. Using the same definitions as before, the MBE is computed as:

$$\frac{1}{N}\sum_{k=1}^{N} y_k - \hat{y}_k. \tag{2.10}$$

While both metrics can properly assess and compare models using the same dataset, they are hard to interpret when it comes to make comparisons across multiple locations, climate, and time of the year [160]. A metric that tries to solve this issue is the forecasting skill s [160]; particularly, s defines first a metric v that accounts for the variability of the solar irradiance, i.e. accounts for the specific variability due to location, climate, and time. Next, it defines a second metric u that accounts for the uncertainty, i.e. errors, of the forecasting model. Finally, the forecasting skill s is defined as:

$$s = 1 - \frac{u}{v}. \tag{2.11}$$

It is important to note that s is a normalized metric w.r.t. to a simple persistence model (see Section 5.3.2) that permits the comparison of models across different conditions. A normal forecaster is characterized by $s \in [0,1]$ with higher values indicating better forecasting; particularly, $s = 1$ indicates that the solar irradiance is perfectly forecasted, and $s = 0$ that the model is not better than a simple persistence model (by definition of u and v a persistence model will always have $s = 0$). While negative values of s are also possible, they represent forecasters that are worse than the simple persistence model.

2.4. ELECTRICITY MARKETS

As introduced in the first chapter, due to their importance in preserving a balanced grid, wholesale markets are of primary importance for the research of this thesis. In this section, we briefly describe their structure and we motivate the focus of the research in the day-ahead market.

2.4.1. STRUCTURE OF WHOLESALE ELECTRICITY MARKETS

To help obtain a balanced grid, European wholesale electricity markets have a very specific structure. Particularly, participants are obliged to submit their projected generation and consumption schedules ahead of time and they are financially responsible for deviations from these schedules. In order to avoid deviations from their notified schedules, market participants can trade in different markets that are mainly distinguished by the time of execution. Particularly, as market participants obtain more accurate information about their actual generation and consumption, they can adjust their schedules by trading in markets with execution times closer to real time. In this context, besides bilateral trading, European actors have several organized marketplaces at their disposal:

- Forward market: Electricity is traded weeks or months in advance.
- Day-ahead market: Electricity is traded up to one day in advance.
- Intraday market: Electricity is traded one day ahead of delivery to one hour or some minutes before delivery time. In addition, electricity is traded continuously, in hourly or quarterly auctions, or a mix thereof [71].

In theory, by having all these markets with different gate closure times, market participants are provided with several opportunities to correct their deviations.

2.4.2. DAY-AHEAD MARKET

In this part of the thesis, the primary focus is on the day-ahead market as it is arguably the most important trading hub for RESs. In particular, due the uncertainty of RES generation, most of the RESs are traded in the spot market, i.e. the day-ahead and intraday markets. Moreover, due to continuous nature of the intraday market, it is much less liquid[3] than the day-ahead market. As a result, the amount of renewable energy traded in the day-ahead is much larger, and research on the day-ahead market is arguably a more important topic.

In the most general format of the day-ahead market, producers and consumers have to submit bids for the 24 hours of day d before some deadline on day $d-1$ (in most European markets, this deadline occurs at 11:00 am or 12:00 pm, i.e. noon). Except for some markets, these bids are defined per hour, i.e. every market player has to submit 24 bids. After the deadline, the market operator takes into account all the bids and computes the market clearing price for each of the 24 hours. Then, consumer bids that are larger than or equal to the market clearing price and producer bids that are lower than or

[3] In several EU markets, the volume traded in the intraday market is more than an order of magnitude lower than the volume in the day-ahead market.

equal to the market clearing price are approved. Figure 2.1 provides the schematic representation of the day-ahead market during day $d-1$ and the posterior delivery of energy at some hour h during day d. As can be seen, a useful forecaster for the day-ahead market should thus be able to predict the set of 24 market clearing prices of day d based on the information available before the deadline of day $d-1$.

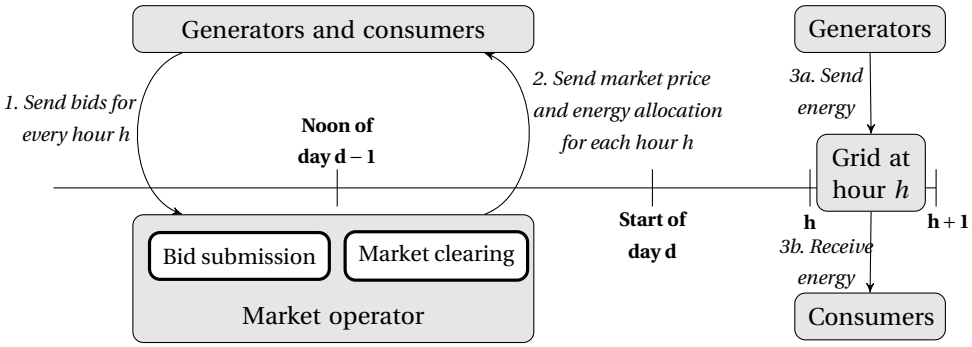


Figure 2.1: Representation of the time frame and decision making in the day-ahead market during day $d-1$, and posterior deliver of energy at some hour h at day d.

2.5. DEEP LEARNING

BECAUSE of the success of DL in different energy applications, in this thesis several DL algorithms are used to improve the state-of-the-art in electricity price and solar irradiance forecasting. To facilitate the understanding of the proposed algorithms, this section provides a brief introduction of DL.

2.5.1. DEFINITION

During the last decade, the field of neural networks has gone through some major innovations that have lead to what nowadays is known as deep learning [86]. Specifically, the term deep refers to the fact that, thanks to the developments of recent years, we can now train different neural network configurations whose depth is not just limited to a single hidden layer (as in the traditional MLP), and which have systemically showed better generalization capabilities [86].

While there are different DL architectures, e.g. convolutional networks or recurrent networks, most of the work in this thesis is based on the standard *deep neural network (DNN)*.

2.5.2. DNN

The basic DL model is the DNN [86], the natural extension of the traditional MLP that uses multiple hidden layers. Defining by $\mathbf{x} = [x_1, \ldots, x_n]^\top \in \mathbb{R}^n$ the input of the network, by $\mathbf{y} = [y_1, y_2, \ldots, y_m]^\top \in \mathbb{R}^m$ the output of the network, by n_k the number of neurons of the k^{th} hidden layer, and by $\mathbf{z}_k = [z_{k,1}, \ldots, z_{k,n_k}]^\top$ the state vector in the k^{th} hidden layer, a general DNN with two hidden layers can be represented as in Figure 2.2.

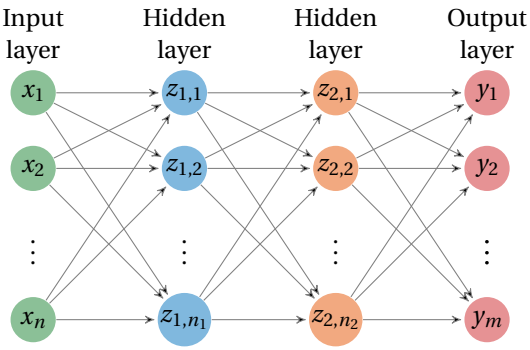


Figure 2.2: Example of a DNN.

MODEL PARAMETERS

In the representation above, the parameters of the model are defined by the vector of weights $\mathbf{w}$ establishing the mapping connections between the different neurons of the network [86]. In particular, in the most general case, the parameters could be defined as follows:

- $\mathbf{w}_{\mathrm{i},i}$: the vector of weights between the input $\mathbf{x}$ and the neuron i of the first hidden layer.
- $\mathbf{w}_{k,i}$: the vector of weights from the $(k-1)^{\text{th}}$ hidden layer to neuron i in the k^{th} hidden layer.
- $\mathbf{w}_{\mathrm{o},i}$: the weights between the last hidden layer and output y_i.
- $\mathbf{b}_k = [b_{k,1}, \ldots, b_{k,n_k}]^\top$: the vector of bias weights of the k^{th} hidden layer.
- $\mathbf{b}_\mathrm{o} = [b_{\mathrm{o},1} \ldots, b_{\mathrm{o},m}]^\top$: the vector of bias weights of the output layer.

In this context, the vector $\mathbf{w}$ is defined as the concatenation of the individual vectors defined above, i.e. $\mathbf{w} = (\mathbf{w}_{\mathrm{i},1}, \mathbf{w}_{\mathrm{i},2}, \ldots, \mathbf{w}_{\mathrm{i},n_1}, \mathbf{w}_{1,1}, \ldots, \mathbf{w}_{1,n_2}, \ldots, \mathbf{w}_{\mathrm{o},1}, \ldots, \mathbf{w}_{\mathrm{o},m}, \mathbf{b}_1, \ldots, \mathbf{b}_\mathrm{o})$.

MODEL EQUATIONS

Based on the above definitions, the equations of a general DNN with l hidden layers can be defined by

$$z_{1,i} = f_{1,i}\left(\mathbf{w}_{\mathrm{i},i}^\top \cdot \mathbf{x} + b_{1,i}\right), \qquad \text{for } i = 1, \ldots n_1, \tag{2.12a}$$

$$z_{k,i} = f_{k,i}\left(\mathbf{w}_{k,i}^\top \cdot \mathbf{z}_{k-1} + b_{k,i}\right), \text{for } i = 1, \ldots n_k, \quad \text{for } k = 2, \ldots l, \tag{2.12b}$$

$$y_i = f_{\mathrm{o},i}\left(\mathbf{w}_{\mathrm{o},i}^\top \cdot \mathbf{z}_l + b_{\mathrm{o},i}\right), \qquad \text{for } i = 1, \ldots m, \tag{2.12c}$$

where $f_{k,i}$ represents the activation function of neuron i in of the k^{th} hidden layer and $f_{\mathrm{o},i}$ the activation function of neuron i in the output layer. Typical activation functions are the sigmoid function, the hyperbolic tangent function, or the rectified linear unit [86].

TRAINING

The process of estimating the model weights $\mathbf{w}$ is usually called training. In particular, given a training set $\mathcal{S}_T = \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^N$ with N data points, the network training is done by solving a general optimization problem with the following structure:

$$\underset{\mathbf{w}}{\text{minimize}} \quad \sum_{k=1}^{N} g_k\Big(\mathbf{y}_k, F(\mathbf{x}_k, \mathbf{w})\Big), \tag{2.13}$$

where $F : \mathbb{R}^n \to \mathbb{R}^m$ is the neural network map, and g_k is the problem-specific cost function, e.g. the Euclidean norm or the average cross-entropy. Traditional methods to solve (2.13) include gradient descent or the Levenberg–Marquardt algorithm [243]. However, although these methods work well for small sized-networks, they display computational and scalability issues for DNNs. In particular, better alternatives for DNNs are stochastic gradient descent and all its variants [86].

It is important to note that (2.13) is an approximation of the real problem we wish to minimize. Particularly, in an ideal situation, we would minimize the cost function w.r.t. to the underlying data distribution; however, as the distribution is unknown, the problem has to be approximated by minimizing the cost function over the finite training set. This is especially relevant for neural networks, where a model could be overfitted and have a good performance in the training set, but perform badly in the test set, i.e. a set with a different data distribution. To avoid this situation, the network is usually trained in combination with regularization techniques, e.g. early-stopping, and using out-of-sample data to evaluate the performance [86].

NETWORK HYPERPARAMETERS

In addition to the weights, the network has several parameters that need to be selected before the training process. Typical parameters include the number of neurons of the hidden layers, the number of hidden layers, the type of activation functions, or the learning rate of the stochastic gradient descent method. To distinguish them from the main parameters, i.e. the network weights, they are referred to as the network hyperparameters.

2.5.3. RECURRENT NEURAL NETWORK (RNN)

While DNN structures are successful in many applications, they might fail to capture the nature of time series data. In particular, given a sequence of model inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$ representing successive time steps, DNNs assume that each input is independent and that an output $\mathbf{y}_k$ only depends on the corresponding input $\mathbf{x}_k$. However, since time series data are usually correlated, this assumption does not always hold true. Instead, a model that is arguably more suited for time series data is the RNN [86], which models the output $\mathbf{y}_k$ as a function of a time input sequence $\mathbf{x}_1, \ldots, \mathbf{x}_k$ instead of a single input $\mathbf{x}_k$. To model this sequential dependence, RNNs build additional mappings between the neurons of the same hidden layer to hold relevant information from past inputs. An example of a RNN is given in Figure 2.3.

A great disadvantage of RNNs is that, although in theory they are able to model any sequential dependence, in practice they are incapable of learning long-term dependencies due to the vanishing gradient problem [23]. More specifically, given a training

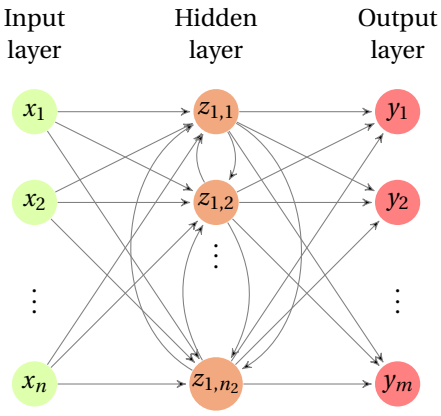


Figure 2.3: Example of a recurrent neural network.

sequence $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{N}$, traditional RNNs have the problem that, due to the recurrence in the structure, the network gradient with respect to an input $(\mathbf{x}_k, \mathbf{y}_k)$ depends on the multiplication of the gradients w.r.t. the previous inputs. As a consequence, as the length of the training sequence increases, the gradient contribution later training pairs either becomes 0 or grows unbounded. In the first case, only earlier inputs of the training sequence are effectively used, and thus, the training becomes slow and hard. In the second case, the training runs into numerical issues. In both scenarios, the end result are RNNs that are unable to hand long-term dependencies.

2.5.4. LONG-SHORT TERM MEMORY (LSTM)

LSTM networks [104] are a type of recurrent networks that avoid the vanishing gradient problem. Whereas in a standard RNN each neuron is represented by a simple neural unit, i.e. a single nonlinear mapping, an LSTM consists of a cell with four neural units. The key idea is that, by using four units per neuron, an LSTM is able to model a memory cell state $\mathbf{c}$ with a selective forget-remember behavior. In more detail, as depicted in Figure 2.4, each LSTM cell consists of three gates: an input gate I, an output gate O, and a forget gate F. Together with a hyperbolic tangent function, these gates represent the four neural units. Then, at a time step k, the cell is characterized by its hidden state $\mathbf{z}_k$, its cell state $\mathbf{c}_k$ and the input state $\mathbf{x}_k$; moreover, the output of the cell is represented by the hidden state $\mathbf{z}_k$. In Figure 2.4, the blue ellipses represent vectorial element-wise operations, the $\odot$ symbol the Hadamard or element-wise product, and the $\oplus$ symbol a concatenation of vectors.

The working principle of the cell is as follows. At any time step k, the neuron regards $\mathbf{z}_{k-1}$ and $\mathbf{x}_k$ as decision variables. Based on them, it uses the neural units to build $\mathbf{f}_k$, $\mathbf{i}_k$ and $\mathbf{o}_k$, three vectors of real numbers between 0 and 1 that select which information from $\mathbf{x}_k$, $\mathbf{c}_{k-1}$, and $\mathbf{z}_{k-1}$ is used to build $\mathbf{c}_k$ and $\mathbf{z}_k$. Defining the parameters of an LSTM cell as the matrices $\mathbf{W}_\mathrm{F}, \mathbf{W}_\mathrm{I}, \mathbf{W}_\mathrm{O}, \mathbf{W}_\mathrm{c}$ and vectors $\mathbf{b}_\mathrm{F}, \mathbf{b}_\mathrm{I}, \mathbf{b}_\mathrm{O}, \mathbf{b}_\mathrm{c}$, the neuronal mapping consists of four steps:

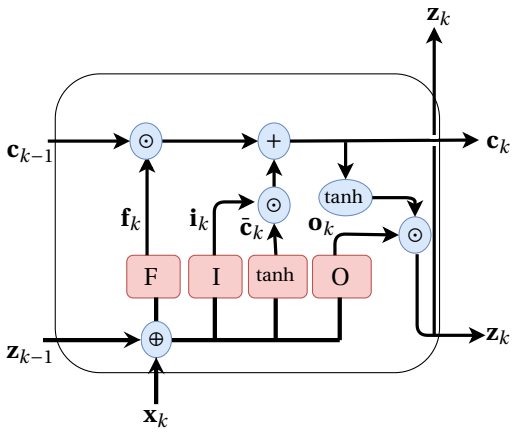


Figure 2.4: Basic LSTM Cell. The $\odot$ symbol represents the Hadamard or element-wise product and the $\oplus$ symbol a concatenation of vectors.

1. The forget gate decides which information from the old cell state $\mathbf{c}_{k-1}$ is kept in $\mathbf{c}_k$ by building the decision vector

$$\mathbf{f}_k = \sigma\left(\mathbf{W}_\mathrm{F}\begin{bmatrix}\mathbf{z}_{k-1}\\ \mathbf{x}_k\end{bmatrix} + \mathbf{b}_\mathrm{F}\right), \tag{2.14a}$$

   with σ representing the sigmoid function.

2. Next, the input gate and the tanh unit select which new information is added. Particularly, the tanh unit creates a vector $\bar{\mathbf{c}}_k$ with the relevant new information:

$$\bar{\mathbf{c}}_k = \tanh\left(\mathbf{W}_\mathrm{c}\begin{bmatrix}\mathbf{z}_{k-1}\\ \mathbf{x}_k\end{bmatrix} + \mathbf{b}_\mathrm{c}\right). \tag{2.14b}$$

   Then, the input gate builds the vector

$$\mathbf{i}_k = \sigma\left(\mathbf{W}_\mathrm{I}\begin{bmatrix}\mathbf{z}_{k-1}\\ \mathbf{x}_k\end{bmatrix} + \mathbf{b}_\mathrm{I}\right), \tag{2.14c}$$

   which selects which of the new information in $\bar{\mathbf{c}}_k$ is kept in $\mathbf{c}_k$:

3. Using $\mathbf{f}_k$ and $\mathbf{i}_k$, the new cell state is built:

$$\mathbf{c}_k = \mathbf{f}_k \odot \mathbf{c}_{k-1} + \mathbf{i}_k \odot \bar{\mathbf{c}}_k, \tag{2.14d}$$

4. Finally, the output gate builds the last decision vector:

$$\mathbf{o}_k = \sigma\left(\mathbf{W}_\mathrm{O}\begin{bmatrix}\mathbf{z}_{k-1}\\ \mathbf{x}_k\end{bmatrix} + \mathbf{b}_\mathrm{o}\right), \tag{2.14e}$$

   which decides which information of $\mathbf{c}_k$ is used for the new hidden state $\mathbf{z}_k$:

$$\mathbf{z}_k = \mathbf{o}_k \odot \tanh(\mathbf{c}_k). \tag{2.14f}$$

2.5.5. GATED RECURRENT UNIT (GRU)

GRU networks [47] are RNNs that use a cell structure very similar to the LSTM case. However, in contrast with an LSTM cell, a GRU cell does not distinguish between the memory cell $\mathbf{c}$ and the hidden state $\mathbf{z}$; instead, it uses a single state variable $\mathbf{z}$. In addition, while an LSTM cell uses a three-gates structure, the GRU cell only requires two: an update gate U and a reset gate R. A representation of an GRU cell is given in Figure 2.5; as before, the red boxes represent the three neural units, the blue ellipses vectorial element-wise operations, the $\odot$ symbol the element-wise product, and the $\oplus$ symbol a concatenation of vectors.

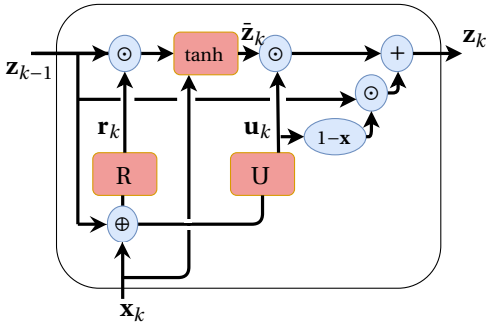


Figure 2.5: Basic GRU Cell. The $\odot$ symbol represents the Hadamard or element-wise product and the $\oplus$ symbol a concatenation of vectors.

The working principle of this cell resembles the one of an LSTM. Particularly, defining the parameters of an GRU cell as the matrices $\mathbf{W}_\mathrm{U}, \mathbf{W}_\mathrm{R}, \mathbf{W}_\mathrm{c}$ and vectors $\mathbf{b}_\mathrm{U}, \mathbf{b}_\mathrm{R}, \mathbf{b}_\mathrm{c}$, the neuronal mapping consists of three steps:

1. The reset gate builds the decision vector

$$\mathbf{r}_k = \sigma\left(\mathbf{W}_\mathrm{R}\begin{bmatrix}\mathbf{z}_{k-1}\\ \mathbf{x}_k\end{bmatrix} + \mathbf{b}_\mathrm{R}\right), \tag{2.15a}$$

   which selects which information from $\mathbf{z}_{k-1}$ is kept in the vector of new information $\bar{\mathbf{z}}_k$.

2. Simultaneously, the tanh unit builds

$$\bar{\mathbf{z}}_k = \tanh\left(\mathbf{W}_\mathrm{c}\begin{bmatrix}\mathbf{r}_k \odot \mathbf{z}_{k-1}\\ \mathbf{x}_k\end{bmatrix} + \mathbf{b}_\mathrm{c}\right), \tag{2.15b}$$

   which contains the relevant new information in $\mathbf{x}_k$ and in the reset gate selection $\mathbf{r}_k \odot \mathbf{z}_{k-1}$.

3. Finally, the update gate builds

$$\mathbf{u}_k = \sigma\left(\mathbf{W}_\mathrm{U}\begin{bmatrix}\mathbf{z}_{k-1}\\ \mathbf{x}_k\end{bmatrix} + \mathbf{b}_\mathrm{U}\right), \tag{2.15c}$$

a decision vector that models the new state $\mathbf{z}_k$ as a trade-off between the old state $\mathbf{z}_{k-1}$ and the new relevant information $\bar{\mathbf{z}}_k$:

$$\mathbf{z}_k = \mathbf{u}_k \odot \bar{\mathbf{z}}_k + (1 - \mathbf{u}_k) \odot \mathbf{z}_{k-1}. \tag{2.15d}$$

Compared with an LSTM network, the GRU has a simpler structure and it is easier and faster to train. In addition, for some applications, it has been empirically shown that it can outperform the LSTM counterpart [47].

2.5.6. CONVOLUTIONAL NEURAL NETWORKS

Another prominent family of DL structures are *convolutional neural networks (CNNs)*. The core idea of a CNN is to analyze an array of data by performing local operations in different areas of the array and outputting the result of these operations to a new layer. As a result, unlike other network structures, a CNN does not have all the layers fully connected.

CNNs are modeled using three building blocks: a convolution operation followed by a nonlinear function, a pooling operation, and a fully connected layer. Given an array of data, the convolution operation slides a filter across the data array and computes an element-wise cross product between the filter and the areas where the filter goes over. Then, for each of the convolved values, a nonlinear map is used and a new array of data is outputted. As different filters capture different properties, CNNs typically use different filters over the input data to output several data arrays. These output arrays are called feature maps, and each one of them represents a distinctive characteristic of the original data array.

As a second building block, the CNN performs a pooling operation. The basic idea is to reduce the size of the feature maps by reducing large areas into single values. Typical operations are the maximum pooling (maximum value of an area) or the average pooling (average value in the area).

Finally, after the CNN subsequently performs several convolutions and pooling operations (with one layer per operation), the values of the final set of feature maps are used as inputs for a fully connected layer. This layer can be used as the output of the network or be followed by more fully connected layers.

It is important to note that, while the above description considered a single input data array, a general CNN can have many; particularly, each of these input arrays is called a channel. A typical 3-channel example is an RGB image, where the blue, red and green layers represent the 3 input channels.

An example of a CNN structure is given in Figure 2.6. In the example, the CNN considers three channels represented by 50×50 data arrays. Then, in a first layer, it computes 24 features maps using 8 different filters per channel. Next, it reduces the size of the maps to 11×11 arrays via a pooling operation. Then, it performs a second convolution and pooling operations that lead to 72 feature maps of size 6×6. Finally, the network becomes fully connected using a DNN with two hidden layers.

It is important to note that, while most of the applications of CNNs have been with images, they have also been applied to other type of data, e.g. time series forecasting [28].

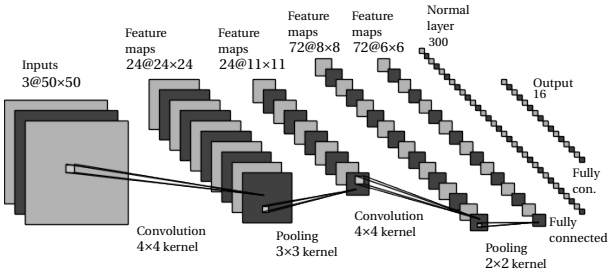


Figure 2.6: Example of a CNN network.

2.6. HYPERPARAMETER SELECTION

To obtain highly accurate forecasting models, a step that is often needed is hyperparameter optimization. In detail, besides the regular parameters of a model, e.g. the weights in a neural network, forecasting models often have other parameters that need to be selected before the training process, e.g. the number of neurons in a neural network or the lag order in an autoregressive model. To distinguish them from the main parameters, they are referred to as the network hyperparameters.

2.6.1. HYPERPARAMETER OPTIMIZATION

In this thesis, to objectively analyze and compare the accuracy of each model, we optimize all hyperparameters using the same algorithm. In particular, we consider a Bayesian optimization algorithm that has been widely used in the machine learning community: the tree-structured Parzen estimator [25], an optimization algorithm within the family of *sequential model-based optimization (SMBO)* methods [110].

The basic principle of an SMBO method is to optimize a black-box function, e.g. the performance of a neural network as a function of the hyperparameters, by iteratively estimating an approximation of the function and exploring the function space using the local minimum of the approximation. At any given iteration i, the algorithm evaluates the black-box function at a new point $\boldsymbol{\theta}_i$. Next, it estimates an approximation $M(\cdot)$ of the black-box function by fitting the previously sampled points to the obtained function evaluations. Then, it selects the next sample point $\boldsymbol{\theta}_{i+1}$ by numerically optimizing $M(\cdot)$ and starts the next iteration. Finally, after a maximum number of iterations n_{iter} have been performed, the algorithm selects the best configuration. Algorithm 1 represents an example of a SMBO for hyperparameter selection.

2.6.2. HYPERPARAMETER ANALYSIS

An optional step after hyperparameter optimization is to perform an analysis of the hyperparameter importance. In particular, although the optimal hyperparameter configuration has been already obtained, it is unknown how much each hyperparameter contributes to the overall performance. Investigating this is specially relevant in order to avoid unnecessary model complexities; e.g. while the optimal number of neurons of a neural network might be large, reducing the number of neurons might barely affect the performance.

Algorithm 1 Hyperparameter Optimization

procedure SMBO($n_{\text{iter}}, \boldsymbol{\theta}_0$)
  $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_0$
  $\mathcal{H} \leftarrow \emptyset$
  for $i = 1, \ldots, n_{\text{iter}}$ **do**
    $\eta_i \leftarrow$ TrainNetwork($\boldsymbol{\theta}_i$)
    $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\eta_i, \boldsymbol{\theta}_i)\}$
    if $i < n_{\text{iter}}$ **then**
      $M_i(\cdot) \leftarrow$ EstimateModel($\mathcal{H}$)
      $\boldsymbol{\theta}_i \leftarrow \text{argmax}_{\boldsymbol{\theta}} M_i(\boldsymbol{\theta})$
    end if
  end for
  $\boldsymbol{\theta}^* \leftarrow$ BestHyperparameters($\mathcal{H}$)
  return $\boldsymbol{\theta}^*$
end procedure

An approach to carry out such an analysis is proposed in [111], where a method based on random forests and functional ANOVA is introduced. In particular, [111] considers the generic case of having z hyperparameters with domains $\Theta_1, \ldots, \Theta_z$, and defines the following concepts:

- Hyperparameter set $\mathcal{Z} = \{1, \ldots, z\}$.
- Hyperparameter space $\Theta : \Theta_1 \times \ldots \times \Theta_z$.
- Hyperparameter instantiation $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_z]^\top$.
- Hyperparameter subset $\mathcal{U} = \{u_1, \ldots, u_q\} \subseteq \mathcal{Z}$ and associated partial hyperparameter instantiation $\boldsymbol{\theta}_{\mathcal{U}} = [\theta_{u_1}, \ldots, \theta_{u_q}]^\top$.

Then, given a set $\mathcal{H} = \{(\boldsymbol{\theta}_k, \eta_k)\}_{k=1}^{n_{\text{iter}}}$ of hyperparameter realizations, the proposed method fits a random forest model $M_{\text{RF}}(\cdot)$ to build a predictor of the performance η as a function of the hyperparameter vector $\boldsymbol{\theta}$. For each hyperparameter subset $\mathcal{U} \subseteq \mathcal{Z}$, the method then uses $M_{\text{RF}}(\cdot)$ to define a marginal performance predictor $\hat{m}_{\mathcal{U}}(\cdot)$ as a forecaster of the performance of any partial hyperparameter instantiation $\boldsymbol{\theta}_{\mathcal{U}}$. In particular, $\hat{m}_{\mathcal{U}}(\boldsymbol{\theta}_{\mathcal{U}})$ provides an estimation of the average performance across the hyperparameter space $\mathcal{Z} \setminus \mathcal{U}$ when the hyperparameters of $\mathcal{U}$ are fixed at $\boldsymbol{\theta}_{\mathcal{U}}$.

Finally, using the marginal performance predictors $\hat{m}_{\mathcal{U}}(\cdot)$, the algorithm carries out a functional ANOVA analysis to estimate the importance of each hyperparameter. Particularly, denoting the total variance across the performance by V, the algorithm partitions V as a sum of individual variance contributions of all possible subsets $\mathcal{U} \subseteq \mathcal{Z}$ to V:

$$V = \sum_{\mathcal{U} \subseteq \mathcal{Z}} V_{\mathcal{U}}, \tag{2.16}$$

where $V_{\mathcal{U}}$ is the contribution of subset $\mathcal{U}$ to the total variance. Then, the importance $F_{\mathcal{U}}$ of each subset $\mathcal{U}$ is computed based on the subset contribution to the total performance

variance:

$$F_{\mathcal{U}} = \frac{V_{\mathcal{U}}}{V}. \tag{2.17}$$

For the particular case of the hyperparameter importance, the algorithm just evaluates $F_{\mathcal{U}}$ for each subset $\mathcal{U} = \{i\}$ composed of a single hyperparameter. As in [111], we refer to the variance contributions $F_{\mathcal{U}}$ of single hyperparameters as main effects and to the rest as interaction effects.

It is important to note that, in addition to the importance $F_{\mathcal{U}}$, the algorithm also provides, for each partial hyperparameter instantiation $\boldsymbol{\theta}_{\mathcal{U}}$, the prediction of the marginal performance $\hat{m}_{\mathcal{U}}(\boldsymbol{\theta}_{\mathcal{U}})$ and an estimation $\hat{\sigma}_{\mathcal{U}}(\boldsymbol{\theta}_{\mathcal{U}})$ of its standard deviation.

2.7. CONCLUDING REMARKS

IN this chapter, we have provided background knowledge on forecasting in the energy domain and on other concepts used in the first part of the thesis. We have started the chapter by motivating the research on forecasting methods to incentivize RESs. Then, we have provided a short literature survey of electricity price forecasting, electricity market integration, solar irradiance forecasting, feature selection for electricity prices, and deep learning. Next, we have described the different methodologies to evaluate forecasting algorithms and explained the structure and working principles of wholesale electricity markets. Finally, we have introduced the area of deep learning and hyperparameter optimization.

3

FORECASTING ELECTRICITY PRICES: THE IMPORTANCE OF MARKET INTEGRATION

If you've got a good idea, and it's a contribution, I want you to go ahead and do it. It is much easier to apologize than it is to get permission.

Grace Hopper

The key to making a good forecast is not in limiting yourself to quantitative information.

Nate Silver

In this chapter, motivated by the increasing integration among electricity markets, we propose two different methods to incorporate market integration in electricity price forecasting. First, we propose a deep neural network that considers features from connected markets to improve the predictive accuracy in a local market. Second, we propose an additional model that, by simultaneously predicting prices from two markets, improves the forecasting accuracy even further. As an additional contribution, we propose a feature selection algorithm based on Bayesian optimization and functional analysis of variance. As a case study, we consider the electricity market in Belgium and the market integration effects from the French market. We show that the two proposed models lead to improvements that are statistically significant and that the predictive accuracy is improved from 15.7% to 12.5% symmetric mean absolute percentage error.

Parts of this chapter have been published in [137].

3.1. INTRODUCTION

DUE to the unique properties of electricity as a commodity, electricity markets have become a central point of research in the energy sector and accurate electricity price forecasting has emerged as one of the biggest challenges faced by the different market entities. In this context, despite its importance, a topic that has been not yet addressed is the influence of neighboring and connected markets, i.e. market integration, on the forecast accuracy. In particular, as different regions in the world, e.g. the European Union [116], are enforcing a larger level of integration across national electricity markets, it is sensible to assume that including information of neighboring markets might play a role in the forecasting efficiency. Yet, although there exist some studies on market integration in the context of price forecasting, the two available papers [175, 258] are limited to the case where the day-ahead prices of neighboring markets are known in advance.

Therefore, while the existing studies provide a first modeling approach for market integration, the methodologies are very specific and can only be applied in limited situations. In particular, most European electricity markets release their day-ahead prices at the same time, and thus, the prices of neighboring markets cannot be obtained in advance. Besides being limited to the case of prices being known in advance, neither [258] nor [175] analyzed the relevance of market integration nor quantified its impact on forecasting accuracy. Hence, even though the effects of market integration can dramatically modify the dynamics of electricity prices, there is a lack of a general modeling framework that could model this effect and analyze its impact on the electricity market.

CONTRIBUTIONS AND ORGANIZATION OF THE CHAPTER

To address this gap, in this chapter we provide general models to identify these relations and a technique to quantify the importance of market integration. As we will show, understanding these relations is key to improve the accuracy of forecasting models, and thus, to obtain energy systems that are economically more efficient.

In contrast to [175, 258], we propose a general modeling framework that is able to model and analyze market integration for any given market. In particular, we propose a modeling framework based on *deep neural networks (DNNs)* that considers market integration features that are available beforehand in all European markets. Using past prices and publicly available load/generation forecasts in neighboring markets, we propose a first forecaster that models market integration effects on price dynamics. Next, we propose a second forecaster that further generalizes market integration: besides modeling market integration using input features, the second forecaster also includes this effect in the output space. In particular, by simultaneously predicting prices in multiple markets, the proposed forecaster is able to improve the predictive accuracy.

Finally, we also contribute to the field of feature selection algorithms. More specifically, although the feature selection methods for electricity price forecasting proposed in the literature provide good and fast algorithms, they suffer from three main drawbacks. First, they all [2, 7, 9, 52, 56, 106, 123, 198, 224] perform a filter step where the model performance is not directly considered; therefore, the resulting selected features might be redundant or incomplete. Second, in the case of the algorithms for nonlinear models [2, 7, 9, 123], the inputs are transformed to lower-dimensional spaces; as a result, fea-

ture information might be lost. Third, while they provide a selection of features, none of these methods computes the relative importance of each feature.

To address these issues, we propose a wrapper selection algorithm based on functional ANOVA (analysis of variance) that directly selects features using nonlinear models and without any feature transformation. Even though the proposed approach is computationally more expensive than previously proposed methods, it can perform a more accurate feature selection as it avoids transformations, selects the features based on the original model, and computes the individual performance of each feature.

The remainder of the chapter is organized as follows: Section 3.2 starts by describing the data that are used in the research. Next, Section 3.3 defines the proposed forecasting framework to model market integration. Section 3.4 derives a novel approach for feature selection and uses it to select the optimal features in the case study. Then, Section 3.5 evaluates the proposed modeling framework by means of predictive accuracy. Finally, Section 3.6 concludes the chapter and outlines the main results.

3.2. DATA

IN this section, we introduce the data used in the case study of this chapter, namely the electricity market and the period under study, the exogenous inputs considered for the forecasting models, and the data preprocessing. These data are used to illustrate the proposed methods with concrete examples. Nonetheless, it is important to note that the proposed methods are general and can be applied to other datasets.

3.2.1. DATA SELECTION

To select the market under study, we consider the EPEX-Belgium and EPEX-France day-ahead markets, two markets where evidence suggests strong common dynamics [164]. Moreover, we select the timeframe from 01/01/2010 to 31/11/2016 as the period under study as to incorporate the recent effects of renewable source integration. To select the exogenous inputs affecting the prices, we try to make sure that the selected data are not only related to the price dynamics, but also fulfill some minimum requirements. More specifically, we only choose data that are freely available for most European markets so that the proposed models can easily be exported to other EU markets:

1. Day-ahead prices from the EPEX-Belgium and EPEX-France power exchanges. They are respectively denoted as p_B and p_F.

2. Day-ahead forecasts of the grid load and generation capacity in Belgium and France. Like in other European markets, these forecasts are available before the bid deadline on the website of the *transmission system operators (TSOs)*: ELIA for Belgium and RTE for France. They are respectively denoted as l_B and g_B for Belgium, and as l_F and g_F for France.

3. Set of dates h_F and h_B respectively representing the public holidays in France and Belgium in the defined time range.

While it could be argued that different weather data could also be easily accessible and important for the forecasting, for our research, we have decided to disregard them for two main reasons:

1. Weather factors are already indirectly taken into account in the grid load and generation forecasts provided by the TSO. In particular, the generation forecast has to consider weather information regarding wind speed and solar radiation. Likewise, load forecasts also need to consider temperature and other weather variables to obtain the electricity consumption.

2. Weather data are local phenomena, and as such, they can greatly vary from one part of a country to another. As a result, unlike the grid load or generation data, it is not possible to select a single value of the temperature or any other weather data for a given time interval.

3.2.2. DATA PROCESSING

It is important to note that the data used are mostly unprocessed. Particularly, as we intend to forecast and detect spikes, price outliers are not eliminated. The only data transformation is a price interpolation[1] and elimination every year corresponding respectively to the missing and extra values due to the daylight saving time. In addition, while all the metrics and tests are computed using the real prices, the training of the neural networks is done with data normalized to the interval $[-1,1]$. This last step is necessary because the input features have very different ranges; therefore, if the data are not normalized, the training time increases and the final result is a network that displays, in general, worse performance [147].

3.2.3. DATA DIVISION

To perform the different experiments, we divide the data into three sets:

1. Training set (01/01/2010 to 31/11/2014): These data are used for training and estimating the different models.

2. Validation set (01/11/2014 to 31/11/2015): A year of data is used to select the optimal hyperparameters and features and to perform early-stopping [253] (see also Section 4.4.2) to ensure that the model does not overfit.

3. Test set (01/11/2015 to 31/11/2016): A year of data, which is not used at any step during the model estimation process, is employed as the out-of-sample dataset to compare and evaluate the models.

3.2.4. DATA ACCESS

For the sake of reproducibility, we have only used publicly available data. The load and generation day-ahead forecasts are available on the webpages of RTE [90] and Elia [91], the respective TSOs in France and Belgium. In the case of the prices, they can be obtained from the ENTSO-E transparency platform [230].

3.3. PROPOSED FORECASTING FRAMEWORK

IN this section, two different models are proposed to include market integration in day-ahead forecasting. The two models are similar to each other as both of them try to

[1] Since only one hour is missing, we perform a linear interpolation between the closest hours.

forecast the full set of day-ahead prices. However, they differ from each other in the number and type of prices that they predict: while the first model predicts the day-ahead prices of a single market, the second model combines a dual market prediction into a single model.

3.3.1. SINGLE-MARKET DAY-AHEAD FORECASTER

The basic model for predicting day-ahead prices uses a DNN in order to forecast the set of 24 day-ahead prices.

MODEL DEFINITION

Based on the results of [135] (see also Chapter 4), we select a DNN with two hidden layers as forecasting model. Defining the input of the model as the relevant data $\mathbf{x} = [x_1, \dots, x_n]^\top \in \mathbb{R}^n$ available at day $d-1$ in the local and neighboring markets, and letting n_1 and n_2 be the number of neurons of the first and the second hidden layer respectively, and $\mathbf{p} = [p_1, p_2, \dots, p_{24}]^\top \in \mathbb{R}^{24}$ the set of 24 day-ahead prices to be forecasted, the proposed model can be represented as in Figure 3.1.

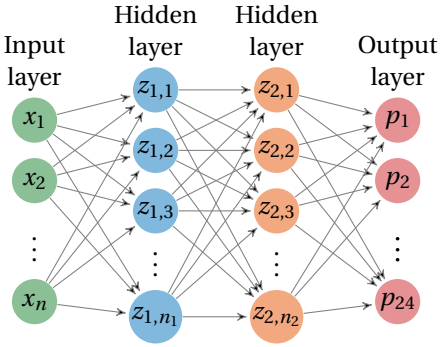


Figure 3.1: DNN to forecast day-ahead prices.

As described in Section 2.5.2, the model parameters are the set of weights that establish the mapping connections between the different neurons.

MODEL EQUATIONS

Using the definitions provided in Section 2.5.2, the equations of the DNN are defined as:

$$z_{1,i} = f_{1,i}\left(\mathbf{w}_{\mathrm{i},i}^\top \cdot \mathbf{x} + b_{1,i}\right), \quad \text{for } i = 1, \dots n_1, \tag{3.1a}$$

$$z_{2,i} = f_{2,i}\left(\mathbf{w}_{\mathrm{h},i}^\top \cdot \mathbf{z}_1 + b_{2,i}\right), \quad \text{for } i = 1, \dots n_2, \tag{3.1b}$$

$$p_i = f_{\mathrm{o},i}\left(\mathbf{w}_{\mathrm{o},i}^\top \cdot \mathbf{z}_2 + b_{\mathrm{o},i}\right), \quad \text{for } i = 1, \dots 24, \tag{3.1c}$$

NETWORK STRUCTURE

As the rectified linear unit [167] has become a standard for hidden layers of DNNs [86], it is selected as the activation function of the two hidden layers. However, as the prices are real numbers, no activation function is used for the output layer. To select the dimension n of the network input and the dimensions n_1 and n_2 of the hidden layers, a feature selection and hyperparameter optimization are performed (see Section 3.4 for more details).

TRAINING

The DNN is trained by minimizing the mean absolute error. In particular, given the training set $\mathcal{S}_{\mathrm{T}} = \{(\mathbf{x}_k, \mathbf{p}_k)\}_{k=1}^{N}$, the optimization problem that is solved to train the neural network is:

$$\underset{\mathbf{w}}{\text{minimize}} \quad \sum_{k=1}^{N} \|\mathbf{p}_k - F(\mathbf{x}_k, \mathbf{w})\|_1, \tag{3.2}$$

where $F : \mathbb{R}^n \to \mathbb{R}^{24}$ is the neural network map. The selection of the mean absolute error instead of the more traditional root mean square error is done for a simple reason: as the electricity prices have very large spikes, the Euclidean norm would put too much importance on the spiky prices.

The optimization problem is solved using the Adam optimizer [127], a version of the stochastic gradient descent method that computes adaptive learning rates for each model parameter. Adam is selected for a clear reason: as the learning rate is automatically computed, the time needed to tune the learning rate is smaller in comparison with other optimization methods. Together with Adam, the forecaster also considers early-stopping [253] to avoid overfitting.

3.3.2. DUAL MARKET DAY-AHEAD FORECASTER

A possible variant of the single-market model is a forecaster that predicts the prices of two markets in a single model. While this might seem counter-intuitive at first, i.e. adding extra outputs to the model could compromise its ability to forecast the set of 24 prices that we are really interested in, this approach can, in fact, lead to neural networks that are able to generalize better.

CONCEPTUAL IDEA

The general idea behind forecasting two markets together is that, as we expect prices in both markets to be interrelated and to have similar dynamics, by forecasting both time series in a single model we expect the neural network to learn more accurate relations. In particular, it has been empirically shown that DNNs can learn features that can, to some extent, generalize across tasks [254]. Similarly, it has also been shown that, by forcing DNNs to learn auxiliary related tasks, the performance and learning speed can be improved [115, 150]. There are some possible hypotheses that can explain why training with multiple outputs can help to improve the performance:

1. The amount of data: As more data are available, the neural network can learn more relevant features. Moreover, as the tasks are related, the neural network has more data to learn features that are common to all tasks.

2. Regularization: By solving different tasks, the network is forced to learn features useful for all tasks and to not overfit to the data of a single task.

MODEL IMPLEMENTATION

Consider an electricity market B and a second electricity market F that is connected to B. Then, defining the output of the network by $\mathbf{p} = [p_1^{\mathrm{B}}, \ldots, p_{24}^{\mathrm{B}}, p_1^{\mathrm{F}}, \ldots, p_{24}^{\mathrm{B}}]^\top \in \mathbb{R}^{48}$, i.e. the set of 48 day-ahead prices from markets B and F, and keeping the rest of the DNN parameter definitions the same, the new DNN structure can be represented as in Figure 3.2. In addition, as both models only differ in the output size, the implementation details are exactly the same as defined for the single-market model in Section 3.3.1.

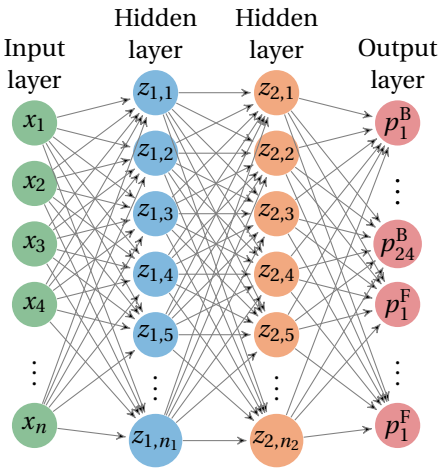


Figure 3.2: DNN to simultaneously forecast day-ahead prices in two markets.

3.4. FEATURE SELECTION ALGORITHM

BESIDES the forecasting models that exploit market integration, we also propose a novel feature selection method. Particularly, as explained in Chapter 2, the existing feature selection methods for electricity price forecasting have two drawbacks. First, they perform a filter step where the model performance is not considered. Second, for the nonlinear methods, the different inputs have to be transformed and some feature information might be lost. To tackle that, we propose a nonlinear wrapper method that directly evaluates the features on the prediction model; although the approach is more computationally demanding than existing methods, it can provide a better selection as it uses the real predictive performance without data transformations. As an additional advantage, the proposed feature selection method can also be used to analyze the effect of market integration on forecasting accuracy.

3.4.1. ALGORITHM DEFINITION

In Section 2.6, we have introduced the *tree-structured Parzen estimator (TPE)*, a method for hyperparameter optimization, together with functional ANOVA, an approach for assessing hyperparameter importance. In this section, we combine both methods to build a feature selection algorithm that consists of four steps:

1. Model the features as hyperparameters.
2. Optimize the hyperparameters/features.
3. Analyze the results.
4. Select the important features.

FEATURES AS HYPERPARAMETERS

The first step of the algorithm is to model the selection of features as model hyperparameters. Particularly, we consider two types of features:

1. Binary features θ_B, whose selection can be done through a binary variable, i.e. $\theta_B \in \{0,1\}$, where $\theta_B = 0$ would represent feature exclusion and $\theta_B = 1$ feature inclusion. Binary features represent the type of features considered by traditional algorithms. An example would be whether to include data of holidays or whether to select a specific lag order in an autoregressive model.
2. Integer features θ_I, which not only can model the inclusion-exclusion of an input, but also select some associated size or length, i.e. $\theta_I \in \mathbb{Z}$, where $\theta_I = 0$ represents exclusion. Examples would be the number of past days of price data or the maximum lag of an autoregressive model.

Given these definitions, the binary features are modeled as hyperparameters using the hyperparameter space $\Theta_B = \Theta_1 \times \ldots \times \Theta_{n_B}$ and the hyperparameter set $\mathcal{B} = \{1,\ldots,n_B\}$. Likewise, the integer features are modeled by the hyperparameter space $\Theta_I = \Theta_{n_B+1} \times \ldots \times \Theta_{n_B+n_I}$ and the hyperparameter set $\mathcal{I} = \{n_B+1,\ldots,n_B+n_I\}$. Finally, the full hyperparameter space is defined by $\Theta = \Theta_B \cup \Theta_I$ and the hyperparameter set by[2] $\mathcal{Z} = \mathcal{B} \cup \mathcal{I}$.

FEATURE OPTIMIZATION

The second step of the algorithm is to perform a TPE optimization over the hyperparameter-feature space. The result of the algorithm is the optimal feature selection $\boldsymbol{\theta}^*$ together with the set $\mathcal{H} = \{(\boldsymbol{\theta}_k, \eta_k)\}_{k=1}^{n_{\text{iter}}}$ of feature-performance pairs, where η_k represents the model predictive accuracy when using the feature selection $\boldsymbol{\theta}_k$.

The fact that a feature is part of $\boldsymbol{\theta}^*$, does not guarantee that the feature is relevant; more specifically, a feature might have little or no effect in the performance, and still, as long as it does not have a negative effect, it might appear in the optimal configuration. As a result, if no further processing is considered, the algorithm might select redundant features, and in turn, lead to more computationally expensive models and increase the risk of overfit.

[2] See Section 2.6.2 for a formal definition of hyperparameter sets and spaces.

FEATURE ANALYSIS

To solve the problem of detecting unnecessary features, the algorithm comprises a third step where feature importance is analyzed. In particular, using the functional ANOVA methodology [111] described in Section 2.6.2, the algorithm analyzes $\mathcal{H}$ and provides the importance F_i of each feature i and the importance $F_{i,j}$ of each pairwise interaction $\{i,j\}$ of features i and j:

$$F_i = \frac{V_i}{V}, \qquad F_{i,j} = \frac{V_{i,j}}{V}, \tag{3.3}$$

where V_i and $V_{i,j}$ are respectively the percentage contribution to the performance variance V of feature i and pairwise interaction $\{i,j\}$. Besides the importance, for each feature $i \in \mathcal{Z}$, the algorithm also provides the predicted marginal performance $\hat{m}_i(\theta_i)$ for each instantiation $\theta_i \in \Theta_i$ of feature i, i.e. the average model accuracy across all features j, with $j \neq i$, when fixing feature i to the instantiation θ_i.

FEATURE SELECTION

The fourth and final algorithm step is the selection itself. Particularly, making use of the obtained F_i, $F_{i,j}$ and $\hat{m}_i(\cdot)$, the selection procedure performs the following steps:

1. Define a threshold parameter $\epsilon \in (0,1]$.

2. Make a pre-selection by discarding features that do not improve nor decrease the performance, i.e. select features i whose importance F_i is larger than ϵ:

$$\mathcal{U}_1^* = \{i \in \mathcal{Z} \mid F_i > \epsilon\}, \tag{3.4a}$$

   or features i that have at least one pairwise contribution $F_{i,j}$ larger than ϵ:

$$\mathcal{U}_2^* = \{i \in \mathcal{Z} \mid \exists\ j \in \mathcal{Z} \setminus \{i\} :\ F_{i,j} > \epsilon\}. \tag{3.4b}$$

3. With the remaining features in $\mathcal{U}_1^* \cup \mathcal{U}_2^*$, perform a second selection $\mathcal{U}^*$ by discarding those features whose predicted marginal performance $\hat{m}_i(\cdot)$ is lower when being included than when being excluded, i.e.:

$$\mathcal{U}^* = \{i \in \mathcal{U}_1^* \cup \mathcal{U}_2^* \mid \forall\ \theta_i \in \Theta_i :\ \mu_{\theta_i=0} < \hat{m}_i(\theta_i)\}, \tag{3.4c}$$

   where $\mu_{\theta_i=0}$ represents the marginal performance $\hat{m}_i(0)$ of excluding feature i.

4. Finally, the set of selected binary features can be obtained by:

$$\mathcal{U}_\mathrm{B}^* = \mathcal{U}^* \cap \mathcal{B}. \tag{3.4d}$$

   Similarly, for the set of optimal integer features $\mathcal{U}_\mathrm{I}^*$, the selection is done in terms of the feature itself and the instantiation with the best performance:

$$\mathcal{U}_\mathrm{I}^* = \left\{\{i, \theta_i^*\} \mid i \in \mathcal{U}^* \cap \mathcal{I},\ \theta_i^* = \underset{\theta_i}{\operatorname{argmax}}\ \hat{m}_i(\theta_i)\right\}. \tag{3.4e}$$

3.4.2. FEATURE SELECTION RESULTS

To evaluate the proposed algorithm, we use it to select the features for predicting Belgian prices and to obtain a first assessment of the effect of market integration, i.e. the effect of using French features in forecasting Belgian prices. To perform the analysis, for the sake of simplicity, we consider the first and simpler DNN proposed in Section 3.3.

FEATURE DEFINITION

In order to perform the feature selection, we first need to model each possible input as either a binary or an integer feature. As described in Section 3.2, the available features are the day ahead prices p_{B} and p_{F}, the day-ahead forecasts l_{B} and l_{F} of the grid load, the day-ahead forecasts g_{B} and g_{F} of the available generation, and the calendar of public holidays H_{B} and H_{F}.

Considering that, given the market at time h, we aim at forecasting the time series vector $\mathbf{p}_h^{\mathrm{B}} = [p_{h+1}^{\mathrm{B}},\dots,p_{h+24}^{\mathrm{B}}]^\top$ of Belgian day-ahead prices, the use of the day-ahead loads $\mathbf{l}_h^{\mathrm{B}} = [l_{h+1}^{\mathrm{B}},\dots,l_{h+24}^{\mathrm{B}}]^\top$ and $\mathbf{l}_h^{\mathrm{F}} = [l_{h+1}^{\mathrm{F}},\dots,l_{h+24}^{\mathrm{F}}]^\top$, and the use of the day-ahead capacity generations $\mathbf{g}_h^{\mathrm{B}} = [g_{h+1}^{\mathrm{B}},\dots,g_{h+24}^{\mathrm{B}}]^\top$ and $\mathbf{g}_h^{\mathrm{F}} = [g_{h+1}^{\mathrm{F}},\dots,g_{h+24}^{\mathrm{F}}]^\top$, should be modeled as binary features $\theta_{\mathrm{l_B}}$, $\theta_{\mathrm{l_F}}$, $\theta_{\mathrm{g_B}}$, and $\theta_{\mathrm{g_F}}$.

Similarly, for the public holidays, the features can also be modeled as binary variables $\theta_{\mathrm{H_B}}$ and $\theta_{\mathrm{H_F}}$. In particular, as the set of 24 hours of a day is either a holiday or not, the holidays are defined as model inputs $x_{H_{\mathrm{B}},h}$, $x_{H_{\mathrm{F}},h} \in \{0,1\}$, with 0 and 1 representing respectively no holiday and holiday.

To model the Belgian prices, we need to use an integer feature to select the number of the considered past values. As the prices display daily and weekly seasonality, we have to use two integer features: $\theta^{\mathrm{d}}_{p_{\mathrm{B}}} \in \{1,2,\dots,6\}$ as the feature modeling the number of past days during the last week (daily seasonality) and $\theta^{\mathrm{w}}_{p_{\mathrm{B}}} \in \{1,2,3\}$ as the feature modeling the number of days at weekly lags (weekly seasonality). Based on the selection of $\theta^{\mathrm{d}}_{p_{\mathrm{B}}}$ and $\theta^{\mathrm{w}}_{p_{\mathrm{B}}}$, the considered EPEX-Belgium past prices can be decomposed as the input $\mathbf{x}^{\mathrm{d}}_{p_{\mathrm{B},h}}$ representing past prices at daily lags and the input $\mathbf{x}^{\mathrm{w}}_{p_{\mathrm{B},h}}$ representing past prices at weekly lags:

$$\mathbf{x}^{\mathrm{d}}_{p_{\mathrm{B},h}} = \left[p^{\mathrm{B}}_{h-i_1},\dots,p^{\mathrm{B}}_{h-i_{N_{\mathrm{d}}}}\right]^\top, \tag{3.5a}$$

$$\mathbf{x}^{\mathrm{w}}_{p_{\mathrm{B},h}} = \left[p^{\mathrm{B}}_{h-j_1},\dots,p^{\mathrm{B}}_{h-j_{N_{\mathrm{w}}}}\right]^\top, \tag{3.5b}$$

where:

$$\{i_1,\dots,i_{N_{\mathrm{d}}}\} = \{i \mid 1 \le i \le 24\cdot\theta^{\mathrm{d}}_{p_{\mathrm{B}}}\} \tag{3.5c}$$

$$\{j_1,\dots,j_{N_{\mathrm{w}}}\} = \{j \mid 1 \le k \le \theta^{\mathrm{w}}_{p_{\mathrm{B}}},\ k\cdot 168 - 24 < j \le k\cdot 168\} \tag{3.5d}$$

It is important to note that, as this is the time series to be predicted, we disregard the cases where no daily nor weekly seasonality is used, i.e. $\theta^{\mathrm{d}}_{p_{\mathrm{B}}} = 0$ or $\theta^{\mathrm{w}}_{p_{\mathrm{B}}} = 0$, by simply removing these values from the feature search space.

Finally, to select the historical prices in the EPEX-France market, we could use similar integer features as for the EPEX-Belgium market. However, for simplicity, we consider the same lags for both time series and model the inclusion of French prices as a binary feature $\theta_{\mathrm{p_F}}$. Particularly, if the binary feature $\theta_{\mathrm{p_F}}$ is selected, then the lagged prices

in France are selected based on the selected lags for Belgium. It is important to note that, although both inputs have the same length, the selection of input features is still independent; particularly, the lags are only defined for Belgium, and the same lags for French prices are simply excluded or included via $\theta_{\mathrm{p_F}}$. The modeled input features are summarized in Table 3.1.

Table 3.1: Definition of the modeled input features.

Feature	Domain	Definition
$\theta^{\mathrm{d}}_{p_{\mathrm{B}}}$	$\{1,\ldots,6\}$	Number of past days for input price sequence
$\theta^{\mathrm{w}}_{p_{\mathrm{B}}}$	$\{1,\ldots,3\}$	Days at weekly lags for input price sequence
$\theta_{\mathrm{p_F}}$	$\{0,1\}$	Day-ahead price in France
$\theta_{\mathrm{l_B}}$	$\{0,1\}$	Load in Belgium
$\theta_{\mathrm{l_F}}$	$\{0,1\}$	Load in France
$\theta_{\mathrm{g_B}}$	$\{0,1\}$	Generation in Belgium
$\theta_{\mathrm{g_F}}$	$\{0,1\}$	Generation in France
$\theta_{\mathrm{H_B}}$	$\{0,1\}$	Holiday in Belgium
$\theta_{\mathrm{H_F}}$	$\{0,1\}$	Holiday in France

HYPERPARAMETER OPTIMIZATION

In order to guarantee that the network is adapted according to the input size, we simultaneously optimize the hyperparameters of the DNN, i.e. the number of neurons n_1 and n_2. In particular, as the feature selection method is based on a hyperparameter optimization, we directly include the number of neurons as integer hyperparameters that are optimized together with the features. We set the domain of n_1 as the set of integers $\{100,101,\ldots,400\}$ and the one of n_2 as $\{0\}\cup\{48,49,\ldots,360\}$, where $n_2 = 0$ represents removing the second hidden layer and using a network of depth one.

EXPERIMENTAL SETUP

In order to use the proposed algorithm, we first need to define the threshold ϵ for the minimum variance contribution; in our case, we select[3] $\epsilon = 0.5\%$. In addition, we also need to select the maximum number of iterations n_{iter} of the TPE algorithm; we found $n_{\mathrm{iter}} = 1000$ to offer a good trade-off between performance and accuracy. Particularly, considering that training a single model takes 2 min, the full feature selection requires 30 h. While this might seem a long time, this step is only performed after some periodic time, e.g. a month, to reassess feature dependencies; therefore, the proposed approach and settings yield a feasible and accurate method for the time scale of day-ahead prices.

[3] In general, as a second step decides which features are really important and ϵ is only used in the first preselection procedure, its value is not very critical and it can be chosen based on different criteria, e.g. to leave in the model as many input features as the available computational power permits. For this study, the selection of ϵ was done because in the experimental results there was a clear separation between features with a variance contribution above 0.8-1% and features with a contribution below 0.1%. So we deemed 0.05% to be a good threshold.

For implementing the functional analysis of variance, we use `fANOVA`, a `python` library developed by the authors of [111]. Likewise, for implementing the TPE algorithm, we use the `python` library `hyperopt` [24].

RESULTS

After performing the feature selection, an unexpected result was obtained: inclusion/exclusion of the generation capacity in Belgium g_B accounts for roughly 75% of the performance variance V, with inclusion of g_B dramatically decreasing the predictive accuracy. Since the generation capacity has been successfully used by other authors as a market driver [243], this result is rather surprising. From Figure 3.3 displaying the time series of g_B this result can be understood: right before the transition from the training to the validation set, the average g_B suffers a major change and drops from approximately 14 GW to 9 GW. Because of the drastic drop, it is likely that some relations that are learned based on the training set, do not hold in the validation set, and that as a result, the predictive performance in the validation set worsens when g_B is considered. As this regime change in g_B violates the assumption that conditions in the training, validation, and test sets are equal, a correct feature selection should disregard θ_{g_B}.

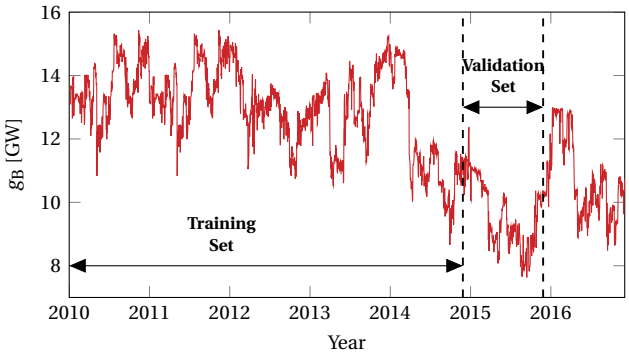


Figure 3.3: Generation in Belgium in the considered period.

Table 3.2 lists the results of the feature selection after disregarding θ_{g_B}. As g_B is a big source of error, the variance V of the *symmetric mean absolute percentage error* (sMAPE) performance is reduced by a factor of 5. In addition, as it could be expected, the results

Table 3.2: Performance variance with and without g_B.

	V
Feature selection with g_B	$0.58\%^2$
Feature selection without g_B	$0.1\%^2$

obtained in this new experiment display a more distributed contribution among the different features. In the first experiment, g_B was responsible for 75% of the performance

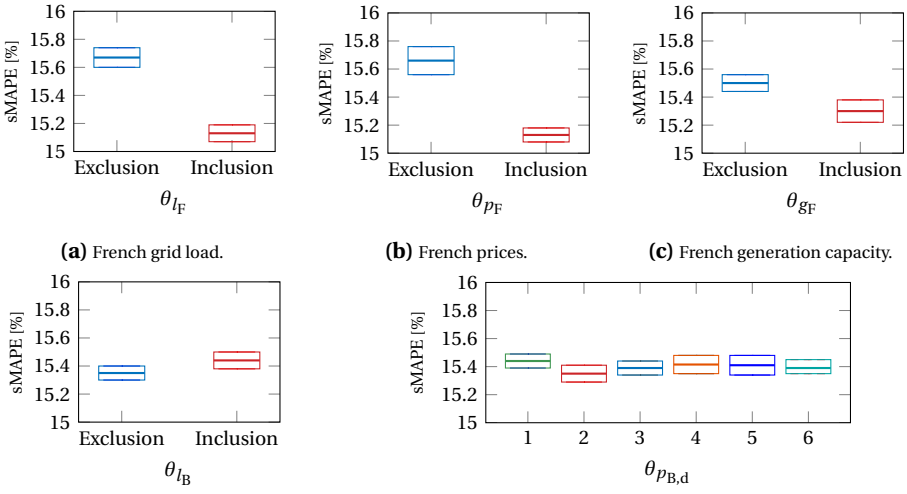
variance. Now, as depicted in Table 3.3, French prices and load account for roughly 50% of the total performance variance, and the available generation in France, the load in Belgium, and the number of past days play a minor role.

Table 3.3: Variance contribution of single features for the second feature selection experiment.

	Contribution to V
All main effects	64.9%
French load	28.4%
French prices	25.7%
French generation	4.78%
Belgium load	1.0%
Past days number	0.8%

Based on the above results, we can make a first selection and remove from the set of possible inputs the public holidays θ_{H_B} and θ_{H_F} as both seem not to be decisive. Similarly, we can select $\theta^{W}_{p_B} = 1$ as the number of days at weekly lags seems to be non-critical. Finally, to complete the feature selection, we should use the marginal performances of the five important features represented in Figure 3.4; based on them, it is clear that we should select the price, load and generation in France, discard the grid load in Belgium, and use two days of past price data.

Together with the features, we have also optimized the hyperparameters of the model. The results show that the suitable numbers of neurons are $n_2 = 200$ and $n_1 = 320$.



(a) French grid load. **(b)** French prices. **(c)** French generation capacity.

(d) Belgian grid load. **(e)** Number of days of past prices in the last week.

Figure 3.4: Marginal performance on the validation set of the five most important features.

SUMMARY
Based on the results of the feature selection algorithm, we should include the following features as model inputs: (i) the day-ahead load and generation in France, (ii) the last two days of Belgian and French prices, (iii) and the Belgian and French prices a week before.

3.5. FORECASTING FRAMEWORK EVALUATION

THE analysis provided by the feature selection algorithm is based on the validation set; while this dataset is not used for training the network, it is employed for early-stopping and hyperparameter optimization. Therefore, to have a fully fair and unbiased evaluation, we need an extra comparison using unseen data to the full training process. Moreover, as the feature selection results were obtained using the first proposed model, results for the second model are also required. Finally, to have a meaningful assessment, the statistical significance of the results should be computed. To fulfill the requirements, the goal of this section is twofold:

1. Provide statistical significance of the improvements of using French market data, i.e. market integration, by performing a *Diebold-Mariano (DM)* test on the out-of-sample data.

2. Based on the same statistical test, demonstrate how a dual-market forecaster can provide significant improvements in predictive accuracy.

3.5.1. DIEBOLD-MARIANO TEST

To assess the statistical significance in the difference of predictive accuracy, we use the DM test as defined by (2.4)-(2.8). Since the neural network is trained using the absolute mean error, we also use the absolute error to build the loss differential:

$$d_k^{M_1,M_2} = |\varepsilon_k^{M_1}| - |\varepsilon_k^{M_2}|. \tag{3.6}$$

In addition, we follow the same procedure as in [173, 258] and perform an independent DM test for each of the 24 time series representing the different hours of a day. The reason for this is that, as we use the same information to forecast the set of 24 prices, the forecast errors within the same day would exhibit a high correlation. However, to have an assessment of the whole error sequence, we also perform the DM test considering serial correlation of order 23. Particularly, recalling that optimal n-step-ahead forecast errors are at most $(n-1)$-dependent [66], we perform a DM test on the full loss differential considering serial correlation of order 23.

In the various experimental setups of this case study, we employ the one-sided DM test given by (2.7) at the 95% confidence level. This selection is done because we want to assess whether the performance of a forecaster A is statistically significantly better than that of a forecaster B, not whether the performances of forecasters A and B are significantly different (like it would be the case in the two-sided DM test as defined in (2.5)).

In detail, for each hour $h = 1,\dots,24$ of the day, we test the null hypothesis of a model M_1 that uses French data having the same or worse accuracy than a model M_2 that uses no French data. Particularly, for each hour $h = 1,\dots 24$, we perform the following test:

$$\begin{cases} H_0 : \mathbb{E}\left[d_{h,k}^{M_1,M_2}\right] \geq 0, \\ H_1 : \mathbb{E}\left[d_{h,k}^{M_1,M_2}\right] < 0, \end{cases} \tag{3.7}$$

where $d_{h,k}^{M_1,M_2}$ represents the k^{th} loss differential of the forecasts at hour h, i.e. $d_{h,k}^{M_1,M_2} = d_{24(k-1)+h}^{M_1,M_2}$. As explained in Section 2.3, the test approximates the expected value $\mathbb{E}\left[d_{h,k}^{M_1,M_2}\right]$ as the average of the sequence of loss differentials[4] $\{d_{h,k}^{M_1,M_2}\}_{k=1}^{N/24}$.

As indicated above, we also perform the same test but considering the full loss differential sequence and assuming serial correlation of order 23:

$$\begin{cases} H_0 : \mathbb{E}\left[d_{k}^{M_1,M_2}\right] \geq 0, \\ H_1 : \mathbb{E}\left[d_{k}^{M_1,M_2}\right] < 0. \end{cases} \tag{3.8}$$

For the specific details on performing the test under the assumption of serial correlation we refer to [66].

3.5.2. FRENCH MARKET DATA: STATISTICAL SIGNIFICANCE

In Section 3.4.2, we have showed that using market data from connected markets can help to improve the performance. In this section, we extend the analysis by directly comparing a model that includes this type of data against a model that excludes it, and then, performing a DM test to analyze the statistical significance.

EXPERIMENTAL SETUP

The model used to perform the evaluation is the single-market forecaster employed for the feature selection. In particular, based on the obtained hyperparameter results, we select $n_1 = 320$ and $n_2 = 200$; similarly, considering the optimized prices lags obtained in the feature selection, we consider, as input sequence for the model, the Belgium prices during the last two days and a week before. Then, we discard as input features the capacity generation in Belgium as well as the holidays in both countries. Then, in order to compare the effect of French data, we consider the remaining features as possible inputs for the model, i.e. we compare the first model excluding all the French data and only considering Belgian prices with respect to the second model including the French data. We respectively refer to these two models as M_{NoFR} and M_{FR}.

In addition, although the load in Belgium l_{B} appears to be non-relevant, we decided to repeat the previous experiment but including l_{B} in both models. The reason for this is twofold:

1. By adding the Belgian load, we ensure that the good results of using French data are not due to the fact that the model does not include specific Belgian regressors.
2. Furthermore, with this experiment, we can also validate the results of the feature selection algorithm. In particular, as the load does not seem to play a big role, we expect the performance difference between models with and without l_{B} to be insignificant.

Similar as before, we refer to these models by $M_{\text{NoFR},l_{\text{B}}}$ and $M_{\text{FR},l_{\text{B}}}$.

[4] Note that the sequence contains $N/24$ losses $d_{h,k}$ per hour h as there are N time points

CASE 1: MODELS WITHOUT l_B

In this experiment, we compare M_{NoFR} against M_{FR} by evaluating their performance on the year of unused data represented by the test set. As in a real-world application, to account for the last available information, the two models are re-estimated after a pre-defined number of days/weeks. In our application, considering that a model takes around 2 minutes to be trained on the GPU, we decide to re-estimate them using the smallest possible period of a day.

A comparison of the models by means of sMAPE is listed in Table 3.4. From this evaluation, we can see that including the French data seems to really enhance the performance of the forecaster.

Table 3.4: Performance comparison between M_{NoFR} and M_{FR} in the out-of-sample data in terms of sMAPE.

Model	M_{NoFR}	M_{FR}
sMAPE	16.0%	13.2%

To provide statistical significance to the above result, we perform a DM test as described in Section 3.5.1. The obtained results are depicted in Figure 3.5, where the test statistic is represented for each of the 24 hours of a day and where the points above the dashed line accept, with a 95% confidence level, the alternative hypothesis of M_{FR} having better performance accuracy. As we can see from the plot, the forecast improvements of the model M_{FR} including French data are statistically significant for each one of the 24 day-ahead prices.

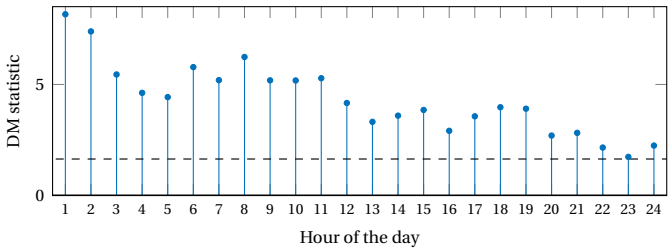


Figure 3.5: DM test results when comparing M_{NoFR} and M_{FR}. Values above the dashed line reject the null hypothesis with a 95% confidence level, and in turn, represent cases where the accuracy of M_{FR} is significantly better.

When the DM test is performed on the full loss differential and taking into account serial correlation, the obtained metrics completely agree with the results obtained for the individual 24 hours. In particular, the obtained p-value is $1.2 \cdot 10^{-11}$, which confirms the strong statistical significance of using the French data in the prediction model.

CASE 2: MODELS WITH l_B

Using the same procedure, we compare M_{NoFR,l_B} against M_{FR,l_B}. From Table 3.5 we can see how, as before, the model including French data outperforms the alternative.

Table 3.5: Performance comparison between M_{NoFR,l_B} and M_{FR,l_B} in the out-of-sample data in terms of sMAPE.

Model	M_{NoFR,l_B}	M_{FR,l_B}
sMAPE	15.7%	13.1%

To provide statistical significance to the obtained accuracy difference we again perform the DM tests. The obtained results are illustrated in Figure 3.6; as before, including French data leads to improvements in accuracy that are statistically significant for the 24 predicted values.

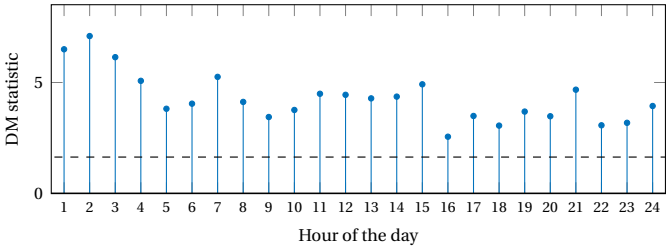


Figure 3.6: DM test results when comparing M_{NoFR,l_B} and M_{FR,l_B}. Values above the dashed line reject the null hypothesis at a 5% significance level, and in turn, represent cases where the accuracy of M_{FR,l_B} is significantly better.

As before, when we consider the DM test for the full loss differential with serial correlation, the p-value is $1.6 \cdot 10^{-12}$, a value that agrees with Figure 3.6 and confirms once more that the improvements of using French data are statistically significant.

ACCURACY OF THE FEATURE SELECTION

Using the results of the previous two sections, we can illustrate the accuracy of the proposed feature selection algorithm in Section 3.4. In particular, when performing the feature selection, we have observed that the contribution of the Belgian load l_B was rather insignificant and even slightly negative; this led to discarding l_B as an input feature. In this section, to verify that the selection algorithm performed the right choice, we perform DM tests to compare M_{NoFR,l_B} against M_{NoFR} and M_{FR,l_B} against M_{FR}. Particularly, we perform a two-sided DM test per model pair with the null hypothesis of the models having equal accuracy.

For the sake of simplicity, we avoid depicting the DM test results for each individual hour; instead we directly illustrate the p-values of the DM test when considering the whole sequence of loss differentials and serial correlation. As can be seen from Table 3.6, the obtained p-values for both tests are above 0.05, and as a result, the null hypothesis of equal accuracy cannot be rejected, i.e. there is no statistical evidence of the models using Belgian load having a different accuracy than the models without it.

Based on the obtained results, it is clear that using l_B is not relevant, and thus, that the choice performed by the feature selection algorithm is correct. In particular, while

Table 3.6: p-values for DM test with the null hypothesis of models with l_B having equal accuracy as models without it.

Model Pair	p-**value**
M_{FR,l_B} vs M_{FR}	0.435
M_{NoFR,l_B} vs M_{NoFR}	0.275

this experiment does not analyze the performance of the feature selection on all the inputs, it does consider the most problematic feature. More specifically, as many researchers have successfully used the load as an explanatory variable [9, 56, 171, 175] and as the load itself does not display any regime change in the considered time interval, it is rather striking to see its minimal effect on the performance. Therefore, by demonstrating that the algorithm is correct when discarding the load, we obtain an assessment of its general accuracy, and we can conclude that the algorithm performs a correct feature selection.

3.5.3. EVALUATION OF A DUAL-MARKET FORECASTER

In this section, we evaluate the possible improvements of using the dual-market forecaster and auxiliary tasks by comparing the single-market model against the dual-market forecaster when the latter simultaneously forecasts the day-ahead prices in Belgium and France. The models are denoted by M_{Single} and M_{Dual} and they both use the optimal features and hyperparameters obtained for the single-market model in Section 3.4. It is important to note that, while in an ideal experiment the hyperparameters of the dual-market forecaster should be re-estimated, for simplicity we decided to directly use the hyperparameters obtained for the single-market forecaster.

The initial comparison is listed in Table 3.7. From this evaluation it seems that using dual-market forecasts can improve the performance.

Table 3.7: Performance comparison between the single and dual-market forecasters in terms of sMAPE.

Model	M_{Single}	M_{Dual}
sMAPE	13.2%	12.5%

To provide statistical significance to these results, we again perform the DM test for each of the 24 hours of a day. The obtained statistics are depicted in Figure 3.7; as before, the points above the upper dashed line accept, with a 95% confidence level, the alternative hypothesis of M_{Dual} having a better performance accuracy. In addition, as not every hourly forecast is statistically significant, we represent in the same figure the alternative DM test with the null hypothesis of M_{Single} having equal or lower accuracy than M_{Dual}. This test is characterized by the lower dashed line and any point below this line accepts, with a 95% confidence level, that M_{Single} has a better predictive accuracy.

As we can see from the plot, the forecast improvements of the dual-market forecaster are statistically significant in 7 of the 24 day-ahead prices. In addition, the single-market forecaster is not significantly better in any of the remaining 17 day-ahead prices. There-

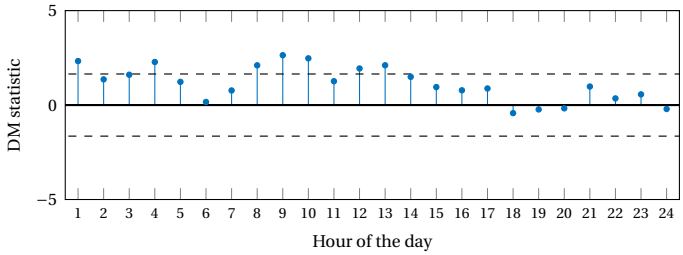


Figure 3.7: DM test results when comparing M_{Single} and M_{Dual}. Values above the top dashed line represent cases where, with a 95% confidence level, M_{Dual} is significantly better. Similarly, values below the lower dashed line accept, with a 95% confidence level, that M_{Dual} is significantly worse.

fore, as M_{Dual} is approximately better for a third of the day-ahead prices and not worse for the remaining two-thirds, we can conclude that the dual-market forecaster is a statistically significant better forecaster.

Finally, we also perform the DM test on the full loss differential considering serial correlation. Once again, the obtained metrics agree with the results obtained for the individual 24 hours: with a p-value of $9.5 \cdot 10^{-03}$, the test results confirm the statistical significance of the difference in predictive accuracy when using the dual-market forecaster.

3.5.4. ANALYSIS AND DISCUSSION

To understand and explain the obtained results, we have to note that market integration across European electricity markets has been increasing over the years due to EU regulations [116]. This highly nonlinear and complex effect dramatically modifies the dynamics of electricity prices and is behind the obtained improvements of our models. In particular, our forecasters use this effect to outperform alternative techniques that have traditionally ignored it. Indeed, the first forecaster proposed, which models market integration in the input space, obtains statistically significant improvements w.r.t. to model counterparts that disregard market integration. Moreover, the second proposed forecaster, which goes one step further by modeling market integration in the output space, is shown to be crucial to obtain further significant improvements. For our case study, this translates to the following conclusions:

1. Using features from the French market significantly enhances the predictive accuracy of a model forecasting Belgian prices. The results are statistically significant and independent of whether Belgian features are considered or not.

2. A dual-market forecaster simultaneously predicting prices in France and Belgium can improve the predictive accuracy. In particular, by solving two related tasks, it is able to learn more useful features, to better generalize the price dynamics, and to obtain improvements that are statistically significant.

3. The proposed feature selection algorithm is able to perform a correct assessment of the importance of features.

In addition, it is interesting to see how explanatory variables from the EPEX-Belgium, e.g. load and generation, have almost no influence on the day-ahead prices. In fact, from the obtained results, it is surprising to observe how French factors play a larger role in Belgian prices than the local Belgian features.

As a final discussion, it is necessary to indicate why, despite being neighboring countries of Belgium, The Netherlands and Germany and their respective markets have not been considered in the study. The reason for not considering The Netherlands was the fact that the amount of available data was smaller than in France and Belgium, and thus, training the DNNs became harder. In the case of Germany, the reason for not considering it is that, at the time when this research was performed, there was not a direct interconnection of the electrical grid between Belgium and Germany.

3.6. CONCLUSIONS

WE have analyzed how market integration can be used to enhance the predictive accuracy of day-ahead price forecasting in electricity markets. In particular, we have proposed a first model that, by considering features from connected markets, improves the predictive performance. In addition, we have proposed a dual-market forecaster that, by solving auxiliary tasks and due to market integration, can further improve the predictive accuracy. As a case study, we have considered the electricity markets in Belgium and France. Then, we have showed how, considering market integration, the proposed forecasters lead to improvements that are statistically significant. Additionally, we have proposed a novel feature selection algorithm and using the same case study, we have shown how the algorithm correctly assesses feature importance.

In view of these results, it is clear that market integration can play a large role in electricity prices. In particular, the influence of neighboring markets seems to be important enough to create statistically significant differences in terms of forecasting accuracy. As a consequence, as the EU has implemented regulations to form an integrated EU market but there is still little insight in the outcome of such regulations, these results are important in terms of policy making. In detail, the fact that market integration largely modifies the price dynamics between Belgium and France is an indicator that the regulations that were put in place are working. As a result, using the proposed methodology, policy makers can benefit from a general tool to evaluate the market integration regulations in other EU regions.

In addition, these results are also of high importance in terms of grid stability and economic profit of market agents. As the knowledge of the dynamics of electricity prices increases, the grid operator might be able to better prevent some of the grid imbalances characterized by large price peaks.

As a first step to help policy markets, in future work the performed experiments will be expanded to the other European markets. Moreover, we will also examine other techniques and concepts to further exploit market integration and improve the accuracy of forecasting models. In Chapter 4, inspired by the models developed in this chapter, we develop this line of research by exploring deep learning algorithms for forecasting electricity prices that also exploit market integration to improve the forecasting accuracy.

4

DEEP LEARNING ALGORITHMS FOR FORECASTING ELECTRICITY PRICES

The combination of some data and an aching desire for an answer does not ensure that a reasonable answer can be extracted from a given body of data

John Tukey

It is far better to foresee even without certainty than not to foresee at all

Henri Poincaré

In this chapter, a novel modeling framework for forecasting electricity prices is proposed. While many predictive models have been already proposed to perform this task, the area of deep learning algorithms remains yet unexplored. To fill this scientific gap, we propose four different deep learning models for predicting electricity prices and we show how they lead to improvements in predictive accuracy. In addition, we also consider that, despite the large number of proposed methods for predicting electricity prices, an extensive benchmark is still missing. To tackle that, we compare and analyze the accuracy of 27 common approaches for electricity price forecasting. Based on the benchmark results, we show how the proposed deep learning models outperform the state-of-the-art methods and obtain results that are statistically significant. Finally, using the same results, we also show that: (i) machine learning methods yield, in general, a better accuracy than statistical models; (ii) moving average terms do not improve the predictive accuracy; (iii) hybrid models do not outperform their simpler counterparts.

Parts of this chapter have been published in [135].

4.1. INTRODUCTION

As the penetration of *renewable energy sources (RES)* increases, so does the volatility in electricity prices [16, 32, 88, 165] and the imbalances between production and consumption. In turn, the behavior of market agents becomes more unpredictable and the electrical grid becomes more unstable. To tackle these issues, electricity price forecasting have become a central point of research: by improving the forecasting accuracy, the negative effects of price uncertainty can be mitigated, the grid can be stabilized, and economic profits can be made. In this context, a research area that has been not yet being investigated is the use of *deep learning (DL)* for electricity price forecasting. Particularly, despite DL algorithms having revolutionized several computer science applications [15, 103, 131] and, more recently, several energy-related applications [50, 75, 79, 130, 240, 241], there has not been yet any attempt to use DL for forecasting electricity prices.

In more detail, although neural networks have been proposed for electricity price forecasting, they have been traditionally limited to one-hidden-layer networks, e.g. *multilayer perceptrons (MLPs)* [39, 123, 175, 224], or to simple versions of *recurrent neural networks (RNNs)*, e.g. Elman networks [10, 213]. While these simpler models are sometimes suitable, there are at least three arguments suggesting that using deeper structures could potentially benefit predictive accuracy:

1. Advanced RNN structures, e.g. *long-short term memory (LSTM)* [104] or *gated recurrent unit (GRU)* [44] networks, have shown to be a much better alternative to accurately model complex nonlinear time sequences [47, 87, 222], e.g. electricity prices.

2. While a single layer network can in theory model any nonlinear continuous function, a network with more than one hidden layer might be able to model the same function with a reduced number of neurons. Therefore, deep networks might actually be less complex and still generalize better than a simple MLP.

3. Considering the excellent results obtained in forecasting time series in other energy-related applications [50, 75, 79, 130, 240, 241], it is possible that forecasting electricity prices might also benefit from using DL architectures.

CONTRIBUTIONS AND ORGANIZATION OF THE CHAPTER

Based on the arguments above, the focus and main contribution of this chapter is to fill the scientific gap by proposing a collection of different DL models that can be successfully used for forecasting day-ahead electricity prices. In detail, the chapter develops a DL modeling framework comprising four models: (i) a *deep neural network (DNN)* as an extension to the traditional MLP, (ii) a hybrid LSTM-DNN structure, (iii) a hybrid GRU-DNN structure, and (iv) a *convolutional neural network (CNN)* model. Then, considering a large benchmark comparison and a case study, it shows that the proposed DL modeling framework leads to improvements in predictive accuracy that are statistically significant.

As a second contribution, the chapter also tries to establish an extensive benchmark of commonly used forecasters for predicting electricity prices. Particularly, since even the largest benchmarks in the literature [7, 54, 56, 244] have been limited to 4-10 different forecasters, the chapter considers that a conclusion on the relatively accuracy of the

different forecasters cannot be drawn. With that motivation, we aim at providing a large empirical evaluation of 27 common forecasters for day-ahead electricity prices to bring new insights on the capabilities of the various models.

The remainder of the chapter is organized as follows: Section 4.2 starts by presenting the proposed DL framework. Then, Section 4.3 defines the baseline forecasters that are considered in the benchmark. Section 4.4 evaluates the baseline and DL models and compares their predictive accuracy by means of hypothesis testing. Next, Section 4.5 discusses and analyzes the obtained results. Finally, Section 4.6 concludes the chapter and provides suggestions for future work.

4.2. DL MODELING FRAMEWORK

As indicated in the introduction, the main goal of this chapter is to propose a DL modeling framework as a forecasting tool for day-ahead electricity prices. As a first step to achieve that, this section develops the four DL models comprising the framework.

4.2.1. MARKET INTEGRATION

Before describing each model separately, it is important to note that a common feature to all DL models is market integration: to improve the predictive accuracy, all the DL models simultaneously predict electricity prices of various day-ahead markets. The idea behind is that, as shown in Chapter 3, due to market integration and solving auxiliary tasks, i.e. predicting prices in different markets, the models can learn more general features and integrate relations across neighboring markets.

In detail, regarding a local market L whose prices need to be forecasted and a set of c neighboring markets $\mathrm{M}_1, \ldots, \mathrm{M}_c$, each DL model predicts the following output:

$$\mathbf{p} = [p_1^{\mathrm{L}}, \ldots, p_{24}^{\mathrm{L}}, p_1^{\mathrm{M}_1}, \ldots, p_{24}^{\mathrm{M}_c}]^\top, \tag{4.1}$$

where $\mathbf{p}^{\mathrm{L}} = [p_1^{\mathrm{L}}, \ldots, p_{24}^{\mathrm{L}}]^\top$ is the vector of day-ahead prices in the local market, and $\mathbf{p}^{\mathrm{M_i}} = [p_1^{\mathrm{M_i}}, \ldots, p_{24}^{\mathrm{M_i}}]^\top$ is the vector of day-ahead prices in the neighboring market M_i.

4.2.2. DNN MODEL

As a simple extension of the traditional MLP, the first DL model for predicting day-ahead prices is a deep neural network with two hidden layers. Defining by $\mathbf{x} = [x_1, \ldots, x_n]^\top$ the input of the model, by n_1 and n_2 the respective number of neurons of the first and the second hidden layer, and by $\mathbf{p} = [p_1^{\mathrm{L}}, \ldots, p_{24}^{\mathrm{L}}, p_1^{\mathrm{M}_1}, \ldots, p_{24}^{\mathrm{M}_c}]^\top$ the vector of day-ahead prices that we intend to forecast, the corresponding model is represented in Figure 4.1.

4.2.3. LSTM-DNN MODEL

The second DL model for predicting day-ahead prices is a hybrid forecaster combining an LSTM and a DNN network. The motivation behind this hybrid structure is to include a recurrent layer that can learn and model the sequential relations in the time series data as well as a regular layer that can learn relations that depend on non-sequential data.

In detail, for this new model, the inputs are divided between those that model sequential time data, e.g. past electricity prices, and those that model regular data, e.g. day of the week or day-ahead forecasting of the grid load. This division is necessary because

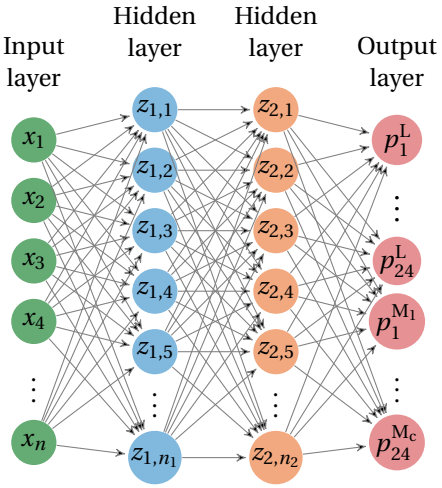


Figure 4.1: Deep neural network to simultaneously forecast day-ahead prices in several countries.

the LSTM network requires a sequence of time series values as an input. However, considering all the possible regressors for electricity price forecasting, it is clear that some of them do not have that property.

In general, for the case of electricity prices, the distinction between these two types of data can be done by considering the time information represented in the data. Specifically, if the data represent a collection of past values, they can normally be modeled as time sequential data and used as an LSTM regressor. By contrast, if the data represent some specific property associated with the day ahead, i.e. they represent direct information of a future event, they cannot be modeled as a time sequence. Examples of the first could be past day-ahead prices or the measured grid load; examples of the second could be the day-ahead forecast of the weather or whether tomorrow (day-ahead) is a holiday. Using this distinction, the inputs of the model are divided into two groups:

- Input vector $\mathbf{x}^{\mathrm{F}} = [x_1^{\mathrm{F}}, \ldots, x_{n_{\mathrm{F}}}^{\mathrm{F}}]^\top \in \mathbb{R}^{n_{\mathrm{F}}}$ representing future information.
- A collection $\{\mathbf{x}_i^{\mathrm{P}}\}_{i=1}^{q}$ of q input sequences, where $\mathbf{x}_i^{\mathrm{P}} = [x_{i,1}^{\mathrm{P}}, \ldots, x_{i,n_{\mathrm{P}}}^{\mathrm{P}}]^\top \in \mathbb{R}^{n_{\mathrm{P}}}$ is a vector representing a specific type of past information, e.g. historical prices in a neighboring market.

Using this separation, the model uses a DNN to process the inputs $\mathbf{x}^{\mathrm{F}}$ and an LSTM to process the time sequences $\{\mathbf{x}_i^{\mathrm{P}}\}_{i=1}^{q}$. Then, the outputs of these two networks are concatenated into one vector and this vector is fed into a regular output layer.

Defining the number of neurons of the DNN and LSTM layers respectively by n_{D} and n_{R}, and by z_i^{F} and $[z_i^{\mathrm{P}}, c_i^{\mathrm{P}}]^\top$ the internal state of their neuron i, an example of the proposed model is represented in Figure 4.2.

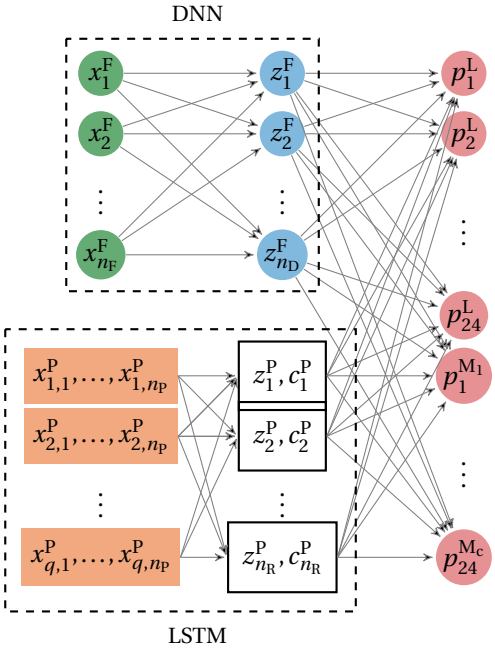


Figure 4.2: Hybrid DNN-LSTM network to simultaneously forecast day-ahead prices in several countries.

4.2.4. GRU-DNN MODEL

The third DL model for predicting day-ahead prices is a hybrid model combining a GRU and a DNN network. As with the LSTM-DNN hybrid structure, the motivation behind this model is to include a layer that is tailored to sequential data. However, to reduce the computational burden of the LSTM layer, a GRU layer is used instead to model the time data sequences $\{\mathbf{x}_i^{\mathrm{P}}\}_{i=1}^{q}$. Specifically, if in Figure 4.2 the LSTM cell states $[z_i^{\mathrm{P}}, c_i^{\mathrm{P}}]^\top$ are replaced by the corresponding GRU cell state z_i^{P}, the modified figure would represent an example of the new proposed model.

4.2.5. CNN MODEL

The fourth DL model for predicting day-ahead prices is a CNN network. As in the previous two cases, the inputs are divided between those that model sequential past data and those that model information regarding the day ahead. For the hybrid models, the division was necessary because the recurrent layers needed sequential data. In this new case, the separation is required in order to group data with the same dimensions as inputs for the same CNN. In particular, the data are separated into two parts:

- The same collection $\{\mathbf{x}_i^{\mathrm{P}}\}_{i=1}^{q}$ of q input sequences used for the hybrid models. As before, $\mathbf{x}_i^{\mathrm{P}} = [x_{i,1}^{\mathrm{P}}, \ldots, x_{i,n_{\mathrm{P}}}^{\mathrm{P}}]^\top \in \mathbb{R}^{n_{\mathrm{P}}}$ is a vector representing some sequential past information.

- A new collection $\{\mathbf{x}_i^{\mathrm{F}}\}_{i=1}^{r}$ of r input vectors, where each vector $\mathbf{x}_i^{\mathrm{F}} = [x_{i,1}^{\mathrm{F}}, \ldots, x_{i,24}^{\mathrm{F}}]^\top \in \mathbb{R}^{24}$ represents future information of the 24 hours of the day ahead. These data are equivalent to the input $\mathbf{x}^{\mathrm{F}} = [x_1^{\mathrm{F}}, \ldots, x_{n_{\mathrm{F}}}^{\mathrm{F}}]^\top$ in the hybrid models, but grouped in several vectors of size 24 instead of a single one of size n_{F}. In particular, the values in $\mathbf{x}^{\mathrm{F}} \in \mathbb{R}^{n_{\mathrm{F}}}$ representing hourly day-ahead values, e.g. forecast of the grid load, are directly mapped into the corresponding 24-values sequence. By contrast, the values in $\mathbf{x}^{\mathrm{F}}$ representing some scalar day-ahead property, e.g. holidays, are repeated 24 times to build the equivalent vector.

Given this separation, the model uses 2 parallel CNNs to model the electricity price dynamics. In detail, a first CNN considers the r input sequences $\{\mathbf{x}_i^{\mathrm{F}}\}_{i=1}^{r}$ as r input channels. Then, a parallel CNN regards the remaining q input sequences $\{\mathbf{x}_i^{\mathrm{P}}\}_{i=1}^{q}$ as q input channels. Next, both networks perform a series of convolution and pooling operations. Finally, the feature maps at the end of both CNNs are connected into a fully connected layer that models the day-ahead prices $\mathbf{p} = [p_1^{\mathrm{L}}, \ldots, p_{24}^{\mathrm{L}}, p_1^{\mathrm{M}_1}, \ldots, p_{24}^{\mathrm{M}_{\mathrm{c}}}]^\top$. As with the hybrid networks, the motivation behind using this structure is to have a network with layers tailored to sequential past data as well as with layers tailored to non-sequential data.

Defining the internal states of both networks by $z_{j,k}^{\mathrm{F},i}$ and $z_{j,k}^{\mathrm{P},i}$, with i representing the layer of the network, j the specific feature map in layer i, and k the state within the feature map j of layer i, Figure 4.3 depicts an example of this type of structure. For the sake of simplicity, the example illustrates both CNNs performing just a single convolution and pooling operation and using only two filters.

4.2.6. SELECTION OF THE NETWORK STRUCTURE

While the structure of the proposed models is general for any electricity market, the specific architecture and implementation details might be not. Specifically, hyperparameters such as the number of neurons might depend on the market under study, and thus, they should be optimized accordingly. In this section, we define which hyperparameters are optimized in the context of the modeling framework.

COMMON HYPERPARAMETERS

While some hyperparameters are model-specific, three of them are common to the four models:

1. **Activation function**: Except for the output layer that does not use any, all the layers within a network use, for the sake of simplicity, the same activation function. This function is chosen using a single hyperparameter that is defined in a discrete hyperparameter space containing four activation functions: the sigmoid function, the hyperbolic tangent function, the rectified liner unit, and the softplus. In the case of the hybrid models, i.e. GRU-DNN and LSTM-DNN, two hyperparameters are used so that each network type can employ a different activation function.

2. **Dropout**: Dropout [217] is included as a possible regularization technique to reduce overfitting and to improve the training performance. To do so, at each iteration, dropout selects a fraction of the neurons and prevents them from training. This fraction of neurons is defined as a real hyperparameter between 0 and 1.

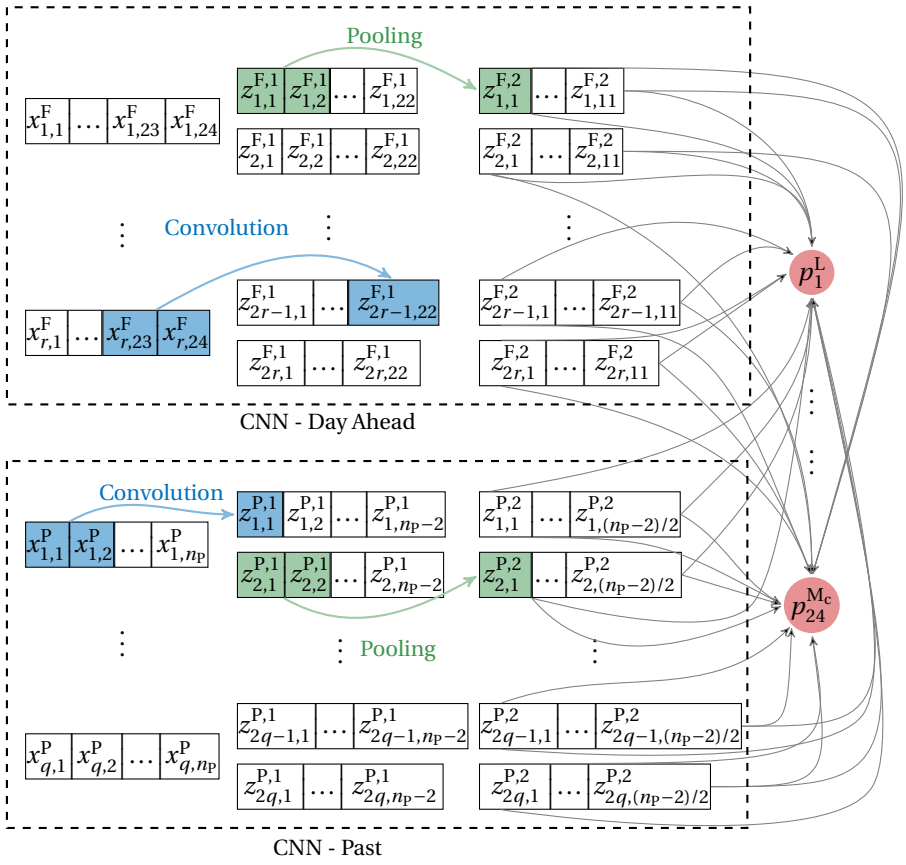


Figure 4.3: Hybrid DNN-LSTM network to simultaneously forecast day-ahead prices in several countries.

3. **L_1-norm penalization**: In addition to dropout, the models can add an L_1-norm penalization to the network parameters as a different way of regularizing. Defining the network weights by $\mathbf{w}$ and using another binary hyperparameter, the models can choose whether to add to the cost function the following term:

$$\lambda \|\mathbf{w}\|_1. \tag{4.2}$$

If regularization is selected, λ becomes a real-valued hyperparameter.

DNN HYPERPARAMETERS

The DNN model uses two additional model-specific hyperparameters:

- $\boldsymbol{n_1}$ / $\boldsymbol{n_2}$: The number of neurons in the first and second hidden layers are optimized with two independent integer hyperparameters.

LSTM-DNN / GRU-DNN HYPERPARAMETERS

For the two hybrid models, there are three additional model-specific hyperparameters:

1. $n_{\text{LSTM}}/n_{\text{GRU}}$: The number of neurons in the recursive layer is optimized with an integer hyperparameter.

2. n_{DNN}: The number of neurons in the DNN layer is optimized with another integer hyperparameter.

3. **Sequence length**: For the LSTM structure, each input is modeled as a sequence of past values. Considering that values too far in the past do not cause any effect in the day-ahead prices, selecting the right length for the input sequences might remove unnecessary complexities. Therefore, a third integer hyperparameter is used to select the length of the input sequences.

CNN HYPERPARAMETERS

Depending on which of the two CNN structures they affect, the specific hyperparameters of the CNN model can be divided into three groups:

1. The hyperparameters that are common and equal to the two CNN structures:

   (a) **Pooling frequency**: The pooling operation does not have to be always performed right after every convolution. Therefore, an integer hyperparameter is used to select how frequently, i.e. after how many convolutional layers, pooling is performed.

   (b) **Pooling type**: To enlarge the number of possible architectures, a binary hyperparameter selects whether the model uses the average pooling or the maximum pooling operation.

2. The hyperparameters that only apply to one of the two CNN structures:

   (c) **Channel length**: For the CNN with past sequences, the length of the input channels is selected as an integer hyperparameter. In the case of the other CNN, the input channels have a length of 24 that correspond with the 24 hours of the day ahead.

3. The integer hyperparameters that apply to both networks but that are independently optimized for each CNN structure:

   (d) **Filter size**: The size of the filter of each convolution operation is optimized with an integer hyperparameter.

   (e) **Number of convolutions**: An integer hyperparameter selects the number of convolutional layers in each CNN.

   (f) **Feature maps in first layer**: The number of feature maps in every layer is determined by selecting the number of feature maps in the first layer. Then, the number of feature maps in successive layers is simply doubled every two

convolutional layers. This choice is used to reduce the total number of hyperparameters. In particular, a more general approach could be to select the number of convolution layers, and then, to model the number of features maps in each of these layers with a different hyperparameter. However, this approach is avoided as it requires a much larger computational cost.

4.2.7. MODEL ESTIMATION

In the proposed framework, all the neural networks are trained by minimizing the mean absolute error. In particular, given the training set $\mathcal{S}_{\mathrm{T}} = \{(\mathbf{x}_k, \mathbf{p}_k)\}_{k=1}^{N}$ with N data points, the networks are trained via the following optimization problem:

$$\underset{\mathbf{w}}{\text{minimize}} \quad \sum_{k=1}^{N} \|\mathbf{p}_k - F(\mathbf{x}_k, \mathbf{w})\|_1, \tag{4.3}$$

where $\mathbf{w}$ represents the vector of all network weights and $F : \mathbb{R}^n \to \mathbb{R}^{24(c+1)}$ the neural network map. The selection of the mean absolute error instead of the more traditional root mean square error is done for a simple reason: as the electricity prices have large spikes, the Euclidean norm would put too much importance on the spiky prices. The optimization problem is solved using Adam [127], a stochastic gradient descent method [29] that uses adaptive learning rates. The advantage of using this optimization method is that the learning rate does not need to be tuned online. Together with Adam, the proposed models also considers early-stopping [253] to avoid overfitting (see Section 4.4.2).

4.3. BENCHMARK FOR ELECTRICITY PRICE FORECASTING

IN order to have a large benchmark study, we consider, in addition to the 4 proposed DL forecasters, a set of 23 different models that have been proposed in the literature of electricity prices forecasting. In addition, to further enlarge the benchmark, we consider different versions of each of the 27 individual models in order to have a benchmark of 98 models.

As the 23 models from the literature will be used to evaluate the proposed DL models, they are referred to as baseline forecasters. Moreover, as the aim of this study is not only the evaluation of the DL models but also to establish a large benchmark within the community of electricity price forecasting, we try to consider a fair selection of baseline models by including the most common and known forecasters from the literature. To make the selection as complete as possible, we use the excellent literature review of [243] and the newest advances in the field. It is important to note that, although the main principles of each baseline model are defined below, the model equations are not provided. Instead, we refer to the original papers for full documentation.

Based on the model separation of [243], the 23 baseline forecasters are divided into three different classes: statistical methods without exogenous inputs, statistical methods with exogenous inputs, and machine learning methods.

4.3.1. STATISTICAL METHODS WITHOUT EXOGENOUS INPUTS

The first class of models comprises statistical methods that only use past prices as input features. Among them, we make the distinction between *autoregressive (AR)* models, *generalized autoregressive conditional heteroscedasticity (GARCH)* models, and exponential smoothing methods.

AR-TYPE MODELS

The first subclass of forecasters assumes homoscedasticity, i.e. constant variance and covariance functions, and models time correlation in the time series using a linear model. Within this subclass, we have selected four models:

1. The well-known Wavelet-*autoregressive integrated moving average (ARIMA)* model [52], a method that has been regularly used in other empirical evaluations [9, 191, 210, 233]. This model will be denoted as *wavelet-ARIMA (WARIMA)*.

2. The *double seasonal ARIMA (DSARIMA)* model [56], an ARIMA model that considers the double seasonality, i.e. weekly and daily, of electricity prices.

3. The AR model of [244], an autoregressive model with lags of 24, 48, and 168 hours, that also models differences among days of the week.

4. The Wavelet-ARIMA-*radial basis function (RBF)* model [210], a forecaster that considers the traditional Wavelet-ARIMA structure but adds an RBF network to model the residuals. This model will be denoted as WARIMA-RBF.

GARCH-BASED MODELS

Unlike the AR-type models, GARCH-based models do not require homoscedasticity in the time series. However, unlike the former, GARCH models are not accurate in forecasting spot electricity prices in standalone application and they need to be coupled with AR-type models to boost their predictive accuracy [243, Section 3.8.6]. As a result, within this subclass, we regard the following hybrid model:

5. The ARIMA-GARCH model [82], a forecaster that considers a standard ARIMA model with GARCH residuals.

EXPONENTIAL SMOOTHING METHODS

The last subclass is exponential smoothing, a family of algorithms that make a prediction using an exponentially weighted average of past observations. Among these methods, we have selected two different forecasters:

6. The *double seasonal Holt-Winter (DSHW)* [226] model, an algorithm that was successfully used by [56] for forecasting spot electricity prices.

7. The *exponential smoothing state space model with Box-Cox transformation, ARMA errors, trend and seasonal components (TBATS)* [61], a forecaster that is able to model multiple seasonality. While this method has never been used before for electricity price forecasting, it is a generalization of the DSHW model [61]. Therefore, it is an interesting method to consider.

4.3.2. STATISTICAL METHODS WITH EXOGENOUS INPUTS

The second class of models are statistical methods that consider regressors to enhance the predictive accuracy. Typical regressors for forecasting electricity prices are the grid load, the available capacity, or the ambient temperature. Among these models, we can distinguish four subclasses: *autoregressive with exogenous inputs (ARX)*-type models, regime-switching models, semiparametric models, and models with automated input selection.

ARX-TYPE MODELS

The first subclass is the natural generalization of adding exogenous inputs to the AR-based models of Section 4.3.1. Like the AR models, they also assume homoscedasticity of the data. For the benchmark, we consider four ARX models:

8. The *dynamic regression (DR)* model [171], an ARX model that uses the grid load as a regressor and that has been used in other empirical evaluations [54].

9. The *transfer function (TF)* model [171], an ARX model with moving average terms that, like the DR model, uses the grid load as a regressor and that has also been used in other comparisons [54].

10. The ARX model proposed in [244], an extension of the AR method defined in Section 4.3.1 that uses the grid load as a regressor. We will refer to this model as ARX.

11. The *full-ARX (fARX)* model [232], a parameter-rich ARX model that considers many input features.

REGIME-SWITCHING MODELS

The second subclass, i.e. regime-switching models, considers that the time series can be modeled by different regimes, that each regime has an independent model, and that switches between regimes can be modeled by the value of some variable. We consider one regime switching model:

12. The *threshold ARX (TARX)* model defined in [244], a model with two regimes that separate normal prices from spiky dynamics. As decision variable, the model uses the difference between the mean price of one day and of eight days before. Then, each of the regimes is modeled with an ARX model that uses the grid load as an exogenous input.

SEMIPARAMETRIC MODELS

Semiparametric models are based on the premise that, given some empirical data, a nonparametric kernel density estimator might lead to a better fit than any parametric distribution. To benefit from this hypothesis, the assumption about the probability distribution that is typically needed when estimating their parametric counterparts is relaxed. An example of semiparametric models are the semiparametric ARX models, which have the same functional form as the equivalent ARX models, but with the normality assumption needed for the maximum likelihood estimation [37, 244] being relaxed. For the benchmark, we regard two different semiparametric models:

13. The *Hsieh-Manski ARX (IHMARX)* estimator, an algorithm originally analyzed in [37] and studied in the context of electricity price forecasting in [244].

14. The *smoothed nonparametric ARX (SNARX)* estimator, a semiparametric model that was also originally analyzed in [37] and applied to electricity price forecasting in [244].

MODELS WITH AUTOMATED INPUT SELECTION

In the last subclass, we consider a set of models that automatically select the important exogenous inputs. Although this type of models are instantiations of the previous three subclasses, we separate them in a fourth subclass due to their special structure. For the benchmark, we consider two of them:

15. The *fARX regularized with Lasso (fARX-Lasso)* [232] model, the fARX model defined in the subclass of ARX models that uses Lasso [228] as a regularization tool to automatically reduce the contribution of unimportant inputs.

16. The *fARX regularized with an elastic net (fARX-EN)* [232] model, the same model but using elastic nets [259] as a regularization tool.

4.3.3. ARTIFICIAL INTELLIGENCE MODELS

The last class of models comprises the machine learning models, a family of algorithms that, while also including exogenous inputs, are able to model more complex nonlinear relations than the previously defined models. Within this class, we can distinguish three subclasses: models based on neural networks, *support vector regressor (SVR)*-based models, and ensemble methods.

NEURAL NETWORK BASED MODELS

This subclass can be seen as a family of simpler DL algorithms. For the benchmark, we regard two different models:

17. The traditional MLP model, a standard neural network with a single hidden layer widely used by many authors [39, 56, 224].

18. The RBF network, a model introduced in Section 4.3.1 as part of a hybrid forecaster that has also had standalone applications [151].

SVR-BASED MODELS

Support vector regressors perform a nonlinear mapping of the data to a higher dimensional space where linear functions are used to perform regression. For the benchmark, we include the following three models:

19. The plain SVR model as used in [76].

20. The *SVR with self-organizing maps (SOM-SVR)* [76, 170] model, a forecaster that first clusters data via *self-organizing maps (SOM)* and then predicts prices using a different SVR model per cluster.

21. The SVR-ARIMA [41] model, a hybrid forecaster that uses an SVR model to capture the nonlinearity of prices and an ARIMA model for the linearities.

ENSEMBLE MODELS

Within this final subclass, we include algorithms based on ensemble methods. Particularly, we consider the two well-known algorithms based on regression trees [100]:

22. The *random forest (RF)* [33] model, a forecaster that predicts data by combining several regression trees. It is based on the principle of bagging [100], i.e. combining models with low bias and high variance error in order to reduce the variance while keeping a low bias.

23. The *extreme gradient boosting (XGB)* [42] model, which also forecasts data by combining regression trees, but which is based on the principle of boosting [100, Chapter 10], i.e. combining models with high bias and low variance in order to reduce the bias whilst keeping a low variance.

It is important to note that, while to the best of our knowledge these models have never been used for electricity price forecasting, we include them in the benchmark as they display reasonable results.

4.3.4. MODELING OPTIONS

To have a more fair comparison, the benchmark models are not only considered in their traditional form. Particularly, for each model, three modeling options with two alternatives per modeling option are considered, i.e. a model that could use the 3 modeling options would have $2^3 = 8$ model alternatives.

MODELING OPTION 1: PREPROCESSING OF SPIKES

Due to the fact that the dynamics of electricity prices are characterized by large, but infrequent, spikes [243], better models might be obtained if spikes are disregarded during the estimation process. As a result, when estimating the model parameters, we consider two model alternatives:

1. **MO1A1**: A first alternative that cuts off the spike amplitude to the mean plus/minus three times the standard deviation.

2. **MO1A2**: A second one that uses raw prices.

MODELING OPTION 2: FEATURE SELECTION

For all the models that include exogenous inputs, there are two additional model alternatives:

1. **MO2A1**: A first alternative that uses the features from the original paper. For all the baseline models, the original input is the day-ahead grid load forecast given by the transmission system operator.

2. **MO2A2**: A second alternative where the features are optimally selected considering all the available data in the market under study. This step is done following the feature selection method described in Chapter 3, where the features are optimally selected by minimizing the *symmetric mean absolute percentage error (sMAPE)* of the model in a validation set.

MODELING OPTION 3: MARKET INTEGRATION

As explained in Section 4.2, all the DL models simultaneously predict electricity prices in various spot markets. This was done because, as shown in Chapter 3, the accuracy of forecasting electricity prices can be enhanced by including market integration. Therefore, for all the forecasters that model the day-ahead prices in a single model, i.e. that do not need 24 independent models, two additional model alternatives are considered:

1. **MO3A1**: A first alternative where the models only predict the prices in the local market.
2. **MO3A2**: A second alternative where the models consider market integration and simultaneously predict the prices in various markets.

Note that, even though this modeling option is only possible for some models, considering market integration is available for many more. In particular, for any of the models with exogenous inputs, market integration could be modeled using features from connected markets as model inputs.

4.3.5. HYPERPARAMETER OPTIMIZATION

In order to have a fair comparison, not only different modeling options should be considered, but also the hyperparameters of the models should be optimized. Particularly, considering that the hyperparameters of the DL models are tuned, the configuration of the baseline models should also be tuned. As described in Section 2.6, this optimization step is performed using Bayesian optimization. Examples of hyperparameters in the baseline models are: the size of the lags in all the AR-based models, the penalty coefficient in the SVR model, or the number of trees in the random forest.

4.3.6. SUMMARY

Table 4.1 summarizes all the considered benchmark methods with their properties and modeling options. The first column denotes whether a model is nonlinear, the second one whether it considers exogenous inputs, and the last three whether the model can make use respectively of modeling options 1, 2, and 3.

Considering the three modeling options, a total of $27 \cdot 2\,(\text{MO1}) + 14 \cdot 2\,(\text{MO2}) + 8 \cdot 2\,(\text{MO3}) = 98$ forecasters are included in the benchmark. As a comparison of 98 models would be too vast, the results in the case study are directly given as the best alternative for each of the 27 individual models. A description of which alternative performs the best for each model is listed in Section 4.4.3.

4.4. CASE STUDY

IN this section, we perform the empirical study to evaluate the proposed DL models and to analyze the predictive accuracy of the various baseline models. To do so, we consider the day-ahead market in Belgium, i.e. *European power exchange (EPEX)*-Belgium, in the period from 01/01/2010 to 31/11/2016. As a first step to analyze the models, we perform the required hyperparameter optimization so that all the forecasters employ an optimized structure. Then, we compare the predictive accuracy of the various forecasters using a year of out-of-sample data. From this comparison, we are able

Table 4.1: Compilation of methods considered in the benchmark. For each model, the first two columns indicate possible properties of the model, namely whether the model is nonlinear and whether it considers exogenous inputs. The last three columns respectively denote whether a model can make use of the 2 alternatives of modeling option 1, the 2 alternatives of modeling option 2, and the 2 alternatives of modeling option 3.

	Properties		**Options**		
Model	Nonlinear	Exog. Inputs	MO1	MO2	MO3
AR			X		
DSARIMA			X		
WARIMA			X		
WARIMA-RBF	X		X		
ARIMA-GARCH			X		
DSHW			X		
TBATS			X		
DR		X	X	X	
TF		X	X	X	
ARX		X	X	X	
TARX		X	X	X	
IHMARX		X	X	X	
SNARX		X	X	X	
fARX		X	X	X	
fARX-Lasso		X	X	X	
fARX-EN		X	X	X	
MLP	X	X	X	X	X
RBF	X	X	X	X	X
SVR	X	X	X	X	
SOM-SVR	X	X	X	X	
SVR-ARIMA	X	X	X	X	
RF	X	X	X		X
XGB	X	X	X		X
DNN	X	X	X		X
LSTM	X	X	X		X
GRU	X	X	X		X
CNN	X	X	X		X

to establish a first evaluation of the DL models as well as to rank the benchmark models according to their performance. Finally, the differences in performance are analyzed via statistical testing.

4.4.1. DATA

In general, when looking at the day-ahead forecasting literature, several inputs have been proposed as meaningful explanatory variables, e.g. temperature, gas and coal prices, grid load, available generation, or weather [243].

DATA SELECTION

For this research, in addition to the past prices p_B in the EPEX-Belgium, we consider several exogenous inputs. In particular, depending on the selected modeling alternative MO2, the specific subset of inputs is given as either one of the following alternatives:

1. A first subset that considers as exogenous input the day-ahead grid load forecast given by the transmission system operator. This selection is done as this variable has been widely used in the literature [232, 244], and for all the baseline models, it is the exogenous input used in the original papers.

2. A second subset that is obtained by regarding all the available information for the market under study and performing feature selection. This step is done following the feature selection method described in Chapter 3. The available input features are:

   (a) The day-ahead forecast l_B of the grid load in the EPEX-Belgium.

   (b) The day-ahead forecast g_B of the available generation in the EPEX-Belgium.

   (c) Past prices p_F in the neighboring EPEX-France market.

   (d) The day-ahead forecast l_F of the grid load in the EPEX-France.

   (e) The day-ahead forecast g_F of the available generation in the EPEX-France.

We make the distinction between these two alternatives because, while it is necessary to optimize each model for our case study, it is also important to evaluate the models in their original format, i.e. as they were originally proposed in the literature.

It is important to note that, although we optimize the input features for every model, discussing the results of the feature selection would be too large to include within the chapter (we evaluate 27 models, each model predicts 24 hours, and there are available more than 750 individual input features that can be selected per hour and per model). As a consequence, the main results of the feature selection, i.e. which features are in general relevant to predict the different hours of the day, are provided in Appendix A.1.

DATA DIVISION

To perform the different experiments, we divide the data into three sets:

1. Training set (01/01/2010 to 30/11/2014): These data are used for training and estimating the different models.

2. Validation set (01/12/2014 to 30/11/2015): A year of data is used to select the optimal hyperparameters.

3. Test set (01/12/2015 to 30/11/2016): A year of data that is not used at any step during the model estimation process, is employed as the out-of-sample data to compare the models.

Considering that there are 24 electricity prices per day, the training dataset comprises 43536 data points. Likewise, both validation and test datasets comprise 8760 data points each.

DATA PROCESSING

In order to obtain time series that are easier to forecast, the data used for the statistical models are processed using a Box-Cox transformation [30]. This preprocessing step, which includes the log-transformation as a special case, is a standard choice in the literature of electricity price forecasting [54–56, 225]. Its goal is to stabilize the variance of the data, alleviate heteroscedasticity, and transform the data to follow a distribution close to the normal one as normality is an important assumption for many statistical techniques. For the machine learning and DL models, the data are respectively normalized to the intervals $[0,1]$ and $[-1,1]$. This transformation is done because, based on experimental results using the validation set, these two preprocessing steps help to obtain more accurate models.

It is important to note that these transformations are only applied when estimating the parameters, not for computing metrics or statistical significance.

DATA ACCESS

For the sake of reproducibility, we have only considered data that are publicly available. The electricity prices can be obtained from the ENTSO-E transparency platform [230]. Similarly, the load and generation day-ahead forecasts are available on the webpages of RTE [90] and Elia [91], the respective TSOs in France and Belgium.

4.4.2. MODELING IMPLEMENTATION: FRAMEWORKS AND LIBRARIES

In order to implement the proposed DL framework, we use the `Keras` [46] DL library in combination with the mathematical language `Theano` [227]. The full framework is developed in `python`.

For the baseline models, the libraries employed differ more. In general, most of the forecasters are also modeled in `python`. The only exception are the DSHW and the TBATS forecasters, both of which are modeled using the `R` language and its `forecast` library [112]. For the remaining 17 models, we can distinguish several groups according to the library/framework used:

1. For the RF, the AR, the DR, the ARX, the TARX, the RBF, the three fARX-based models, and the three SVR-based models, the `scikit-learn` library [177] is used.

2. The XGB model is built using the `xgboost` library [42].

3. The MLP is modeled using the same frameworks as the DL models, i.e. `Keras` [46] in combination with `Theano` [227].

4. The remaining models, i.e. the IHMARX, the SNARX, the TF, and the 4 ARIMA-based models, are estimated by solving the corresponding maximum likelihood estimation problem. To do so, we employ `CasADi` [11], a symbolic framework for automatic differentiation and numerical optimization. Within this group, we also model the ARIMA part of the SVR-ARIMA model.

In addition, to solve the optimization problems that estimate the models' parameters, we distinguish between two different stopping criteria:

1. Except for the neural network models, the stopping criterion is given by the moment that a (local) minimum is reached. We assume that a local minimum is reached when the L_2-norm of the gradient of the objective function is lower than some tolerance; in our study, that was 10^{-6}.

2. For the neural network models, we monitor the performance of a validation set and we stop the training when the improvements on this validation set cease (we assume that the improvement ceases if the accuracy in the validation set worsens for ten consecutive epochs). This criterion is called early-stopping [253], and it is done because neural networks would overfit to the training data and would not generalize well if a (local) minimum is reached.

It is important to note that, for all non-convex models, the described stopping criteria cannot ensure that the best model is found, i.e. the optimal solutions are in local minima or in their vicinity. To improve this situation, we have added multi-start optimization with 10 initial points[1] to the hyperparameter selection; by doing so, when optimizing the hyperparameters, larger regions of the parameter space are explored and the quality of the obtained local solution can be improved.

4.4.3. BEST ALTERNATIVE PER MODELING OPTION

In Section 4.3.4, we have described the three modeling options that are available for each benchmark model. In this section, we present and explain the best alternative for each of the options when considering the case study. It is important to note that all the results listed here are based on the validation dataset.

The obtained results are listed in Table 4.2 where, for each benchmark model and each modeling option, i.e. MO1, MO2, and MO3, the best model alternative is shown. In particular, the optimal alternative is given by one of the following labels:

- A1 (A2) to respectively denote that alternative 1 (2) performs the best.
- NI (non-important) to denote that the modeling option has no effect, i.e. both alternatives perform similarly.
- No label if the model cannot use the modeling option.

Based on the results of Table 4.2 we can draw the following conclusions:

1. Considering the results of modeling option MO1, preprocessing price spikes (Alternative A1) seems to be helpful for all statistical models. In contrast, preprocessing seems to be irrelevant or decrease the performance in the case of machine learning models. A possible explanation for this effect is the fact that price spikes are nonlinear effects, and as such, they can compromise the prediction quality of statistical models since they are largely linear [5]. In contrast, as machine learning models are able to model more complex nonlinear relations, it is possible that they can predict up to certain degree some of the nonlinear price spikes.

[1] We considered 10 initial points because, based on empirical results, it was observed that 10 initial points were sufficient to improve the quality of the local solutions considerably.

Table 4.2: Summary of which alternatives of the three modeling options perform the best for each individual model. The labels A1|A2 respectively denote the case where alternative 1|2 performs the best and NI denotes the case where the modeling option has no effect. An empty cell means that the model cannot use the modeling option.

	MO1	MO2	MO3
AR	A1		
DSARIMA	A1		
WARIMA	A1		
WARIMA-RBF	A1		
ARIMA-GARCH	A1		
DSHW	A1		
TBATS	A1		
DR	A1	NI	
TF	A1	NI	
ARX	A1	NI	
TARX	A1	NI	
IHMARX	A1	A1	
SNARX	A1	A1	
fARX	A1	A2	
fARX-Lasso	A1	A2	
fARX-EN	A1	A2	
MLP	NI	A2	NI
RBF	A1	A2	A1
SVR	NI	A2	
SOM-SVR	NI	A2	
SVR-ARIMA	NI	A2	
RF	A2		A1
XGB	A2		A1
DNN	A2		A2
LSTM	A2		A2
GRU	A2		A2
CNN	A2		A2

2. Observing the results of modeling option MO2, it is clear that, except for the non-parametric models, when the input features are optimally selected (Alternative A2) the accuracy of the models improves. In particular, the models obtain better performance when, instead of simply considering the load in the local market (Alternative A1), the model also includes input features like the load or generation in a neighboring market.

3. Analyzing the results of modeling option MO3, we can observe how the accuracy improvements by predicting multiple markets at the same time (Alternative A2) are restricted to the DL models. As originally argued in Chapter 3, this result is due to using auxiliary tasks, a technique that can be employed to improve the predictive accuracy of DL models but that might not be helpful for other models. In particular, when DL models solve auxiliary and related tasks, e.g. predicting neighboring markets, they generalize better and avoid overfitting.

4.4.4. HYPERPARAMETER OPTIMIZATION

In Section 4.2.6, we have described the hyperparameters that should be optimized for each DL model. In this section, we present the obtained optimal configurations for the case study. For the baseline models, even though their hyperparameters are also optimized, including here the optimization results and hyperparameter definitions would require a large amount of space. Thus, for the sake of conciseness, the results and definitions are listed in Appendix A.2.

When analyzing the results, it is important to keep in mind that all the hyperparameter solutions (and in turn the model sizes) depend on the current amount of data. In particular, as deep learning models employ a large number of parameters, they also require large amounts of data to accurately estimate their parameters. If the amount of data is not enough, the hyperparameter optimization might select a model that, despite performing better with the current amount of data, has a smaller size and performs worse than the optimal model that would be obtained if more data were available. As we have argued in Section 4.5, this effect might explain the lower empirical performance observed for the most complex model, i.e. the CNN.

DNN MODEL

For the DNN, the optimal structure consists of a first and second hidden layers with respectively 239 and 162 neurons, the *rectifier linear unit (ReLU)* as the activation function, and no regularization nor dropout. The obtained optimal hyperparameters are summarized in Table 4.3.

Table 4.3: Optimal Hyperparameters for the DNN model.

Hyperparameter	**Value**
Activation Function	ReLU
Dropout	No
Regularization	No
n_1	239
n_2	162

LSTM MODEL

For the second proposed model, the optimal structure is an LSTM layer with 83 neurons and a regular layer with 184 neurons. Moreover, for the LSTM layer, the activation func-

tion is a *hyperbolic tangent (tanh)* function and the sequence length of input values is 2 weeks of past data. For the regular layer, the optimal activation is a ReLU function. In addition, none of the two layers require regularization nor dropout. The obtained optimal hyperparameters are represented in Table 4.4.

Table 4.4: Optimal Hyperparameters for the LSTM model.

Hyperparameter	Value
Activation Function - DNN	ReLU
Activation Function - LSTM	Tanh
Dropout	No
Regularization	No
n_{DNN}	184
n_{LSTM}	83
Sequence Length	2 weeks

GRU MODEL

Similar to the LSTM-DNN model, the optimal hyperparameters for the GRU-DNN model are summarized in Table 4.5.

Table 4.5: Optimal Hyperparameters for the GRU model.

Hyperparameter	Value
Activation Function - DNN	ReLU
Activation Function - LSTM	Tanh
Dropout	0.32
Regularization	No
n_{DNN}	166
n_{GRU}	132
Sequence Length	3 weeks

CNN MODEL

Finally, for the CNN model, the network that processes past data consists of three convolutional layers with respectively 64, 128, and 256 feature maps, each of them with a filter of size 3. After each of these layers, a max pooling operation and a batch normalization are performed. For the network that processes day-ahead data, the optimal structure is exactly the same. Both networks use the ReLU as activation function, a dropout factor of 0.31, and no regularization. The obtained optimal hyperparameters are summarized in Table 4.6.

Table 4.6: Optimal hyperparameters for the CNN model. The label D.A. refers to the network that processes day-ahead data. The label Past refers to the network for past data.

Hyperparameter	Value
Activation Function	ReLU
Dropout	0.31
Regularization	No
Pooling frequency	1
Pooling type	Max pooling
Filter size - Past	3
Filter size - D.A.	3
Number of convolutions - Past	3
Number of convolutions - D.A.	3
Initial feature maps - Past	64
Initial feature maps - D.A.	64
Channel length	1 week

GENERAL OBSERVATIONS

When analyzing the optimal hyperparameter results for the DL models, we can observe two interesting results that are common to the four models:

1. Except for the recurrent layers that require a tanh activation function, the optimal activation function for all the other deep learning layers is the ReLU function. This result agrees with the general observations in the field of DL, see e.g. [86], where ReLU is the default recommended activation function for any modern neural network with the exception of the LSTM and GRU cells (which by default require a tanh activation function).

2. Traditional regularization, i.e. performing dropout or penalizing with an L_1-norm the parameters of the neural network to impose sparsity on the network parameters, is in general not helpful (the only exception is the CNN model which does require dropout). Although this result might seem surprising (considering the small size of the datasets and the large number of parameters of the DL networks), it can be explained due to the combination of two effects:

   (a) While the proposed models are deep structures, they are less deep than DL networks used for more traditional applications, e.g. image or speech recognition. As a result, the number of parameters is smaller, and thus, the regularization step is less critical.

   (b) The models are trained using early-stopping [253]. Although this is not a regularization technique by itself, it prevents overfitting. As a result, the regularization step becomes less critical.

4.4.5. COMPARING PREDICTIVE ACCURACY

After describing the experimental setup and obtaining the optimal model structures, we can compute and compare the predictive accuracy of the various models. However, to have a meaningful and complete assessment, not only the accuracy of the models should be computed, but also the statistical significance of the results should be established. In this section, we perform the first step of this analysis, i.e. we compute the accuracy of the models. Next, in the following section, the statistical tests are performed.

MAIN RESULTS

To compare and analyze the predictive accuracy of the various forecasters, we compute their sMAPE on the test set. In addition, to guarantee that the assessment is similar to real conditions, i.e. that the forecaster is re-estimated when new data are available, the models are re-estimated on a daily basis. The obtained results are listed in Table 4.7.

OBSERVATIONS

From the results displayed in Table 4.7, we can make various observations:

1. The DNN, GRU, and LSTM models, i.e. 3 of the 4 proposed DL forecasters, seem to outperform all the considered literature models.
2. A line can be drawn between statistical models and machine learning methods. In particular, except for the fARX-based models, the other statistical methods perform worse than any artificially intelligence model.
3. According to their performance, the models seem to be divided in eight clusters:
   (a) The DNN model with a 12.3% sMAPE.
   (b) The DL models with a recurrent layer, i.e. LSTM and GRU, with a 13% sMAPE.
   (c) The three SVR-based models and the MLP with a 13.3-13.4% sMAPE.
   (d) The CNN, the XGB, and the statistical models with automatic feature selection with a sMAPE between 13.7-13.9%.
   (e) The RF, the fARX, and the RBF models with a 14.7-15-3% sMAPE.
   (f) With a 16.7-17.9% sMAPE, the TBATS and the statistical methods with exogenous inputs but without moving average (except for the fARX).
   (g) With a 19.3-19.4% sMAPE, the ARIMA-GARCH and 2 of the 3 models without exogenous inputs nor moving average.
   (h) With a 22-23% sMAPE, the statistical methods with a moving average term (except for the ARIMA-GARCH) .
4. Surprisingly, the models with moving average seem to perform worse that their simpler AR counterparts.
5. The TBATS model appears to be the best alternative when no exogenous inputs are available. In particular, it even matches the performance of some statistical methods with exogenous inputs.

Table 4.7: Comparison of the predictive accuracy of the various forecasters by means of sMAPE. The labels ML and SM respectively refer to machine learning and statistical methods.

Model	sMAPE [%]	Class
DNN	12.34	ML
GRU	13.04	
LSTM	13.06	
MLP	13.27	
SVR	13.29	
SOM-SVR	13.36	
SVR-ARIMA	13.39	
XGB	13.74	
fARX-EN	13.76	SM
CNN	13.91	ML
fARX-Lasso	13.92	SM
RBF	14.77	ML
fARX	14.79	ST
RF	15.39	ML
IHMARX	16.72	ST
DR	16.99	
TARX	17.08	
ARX	17.34	
SNARX	17.58	
TBATS	17.90	
ARIMA-GARCH	19.30	
AR	19.31	
DSHW	19.40	
WARIMA-RBF	22.82	
WARIMA	22.84	
DSARIMA	23.40	
TF	23.57	

6. From the considered models from the literature, SVRs and MLPs perform the best.

7. The SVR hybrid methods, i.e. SVR-ARIMA and SOM-SVR, perform no different that the simple SVR model.

4.4.6. STATISTICAL TESTING

In this section, we study the statistical significance of the differences in predictive accuracy among the various forecasters.

DIEBOLD-MARIANO TEST

To assess this statistical significance, we use the *Diebold-Mariano (DM)* test as defined by (2.4)-(2.7). As in Chapter 3, the loss differential is built using the absolute error. Moreover, we follow the procedure of [173, 258] and we perform an independent DM test for each of the 24 time series representing the hours of a day. In detail, defining by $d_{h,k}^{M_1,M_2}$ the k^{th} loss differential of the forecasts at hour h, i.e. $d_{h,k}^{M_1,M_2} = d_{24(k-1)+h}^{M_1,M_2}$, we consider the sequence of loss differentials[2] $\{d_{h,k}^{M_1,M_2}\}_{k=1}^{N/24}$ for every hour $h = 1,\dots,24$. Then, for each model pair M_1 and M_2 and for each hour $h = 1,\dots,24$, we perform a one-sided DM test at a 95% confidence level with the null hypothesis of the predictive accuracy of M_1 being equal to or worse than that of M_2:

$$\text{DM}_h \begin{cases} H_0 : \mathbb{E}[d_{h,k}^{M_1,M_2}] \geq 0, \\ H_1 : \mathbb{E}[d_{h,k}^{M_1,M_2}] < 0. \end{cases} \tag{4.4}$$

Next, we perform the complementary one-side DM test with the null hypothesis of M_2 having the same or worse accuracy than M_1:

$$\hat{\text{DM}}_h \begin{cases} H_0 : \mathbb{E}[-d_{h,k}^{M_1,M_2}] \geq 0, \\ H_1 : \mathbb{E}[-d_{h,k}^{M_1,M_2}] < 0. \end{cases} \tag{4.5}$$

Finally, we establish that the predictive accuracy of M_1 is significantly better than M_2's if two conditions are met:

1. In at least one of the regular DM_h tests the null hypothesis is rejected, i.e. the predictive accuracy of M_1 is at least significantly better in 1 of the 24 prediction windows.

2. None of the complementary $\hat{\text{DM}}_h$ tests rejects the null hypothesis, i.e. the predictive accuracy of M_2 is not significantly better in any of the 24 prediction horizons.

If both M_1 and M_2 are at least significantly better in one of the 24 prediction windows, we perform a further DM test considering the full sequence of loss differential $\{d_k^{M_1,M_2}\}_{k=1}^{N}$. Specifically, recalling that optimal n-step-ahead forecast errors are at most $(n-1)$-dependent [66], we perform a DM test on the full loss differential considering serial correlation of order 23:

$$\text{DM}_{\text{sc}} \begin{cases} H_0 : \mathbb{E}[d_k^{M_1,M_2}] \geq 0, \\ H_1 : \mathbb{E}[d_k^{M_1,M_2}] < 0. \end{cases} \tag{4.6}$$

If the null hypothesis of DM_{sc} is rejected, we consider that, while at some hours M_2's accuracy is significantly better than M_1's, M_1's accuracy is significantly better when considering the full error sequence.

[2] Note that the sequence contains $N/24$ losses $d_{h,k}$ per hour h as there are N time points.

RESULTS

The obtained results are summarized in Table 4.8, which displays the results based on three possible scenarios:

1. Cells that display a ✓ represent the cases where the alternative hypothesis is accepted with a 95% confidence, i.e. the predictive accuracy of M_1 is statistically significantly better than the one of M_2.

2. Cells that display a $\checkmark_s$ represent the cases where, while the predictive accuracy of M_2 is at least significantly better in one of the 24 predictive horizons, the overall predictive accuracy of M_1 when considering the full loss differential is still statistically significantly better.

3. Empty cells represent the cases where M_1 is not significantly better than M_2.

Considering the results listed in Table 4.8, we confirm the various observations made in Section 4.4.5:

1. The DNN, LSTM, and GRU models, i.e. 3 of the 4 proposed forecasters, are indeed statistically significantly better than the rest. The DNN shows a predictive accuracy that is statistically significantly better than the accuracy of all others. The LSTM and GRU models have an accuracy that is statistically significantly better than all others except the MLP.

2. Except for the fARX-based models, the accuracy of the machine learning methods is statistically significantly better than the accuracy of statistical methods.

3. Based on accuracy differences that are statistically significant, we can observe a very similar group separation pattern as the one described in Section 4.4.5.

4. The models with moving average terms have an accuracy that is statistically significantly worse than their AR counterparts.

5. The TBATS model has an accuracy that is statistically significantly better than any other model without exogenous inputs.

6. The accuracy of the SVR and hybrid-SVR models is not statistically significantly different.

To illustrate the first observation, i.e. that the predictive accuracy of the proposed DNN, GRU and LSTM models is significantly better than the predictive accuracy of the other models, we depict in Figures 4.4 and 4.5 the test statistics obtained when applied to the DNN and GRU models. In these figures, at each hour h, the points above the upper horizontal line accept, at a 95% confidence, the alternative hypothesis in DM_h, i.e. that the specific DL model has an accuracy that is statistically significantly better. Similarly, any point below the lower horizontal line accepts, at a 95% confidence, the alternative hypothesis in $\hat{\mathrm{DM}}_h$, i.e. that the specific DL model has an accuracy that is statistically significantly worse.

Table 4.8: DM results for the baseline and DL models. ✓ represents the cases where M_1's accuracy is statistically significantly better than M_2's. ✓$_s$ represent the cases where, while M_2's accuracy is at least significantly better in one of the 24 hours, the accuracy of M_1 is still statistically significantly better if the whole loss differential sequence is considered. The labels ST and ML respectively refer to statistical and machine learning methods.

Model Class	Statistical Methods													ML	ST	ML	ST	ML	ST	Machine Learning							
M_1 \ M_2	TF	DSARIMA	WARIMA	WARIMA-RBF	DSHW	AR	ARIMA-GARCH	TBATS	SNARX	ARX	TARX	DR	IHMARX	RF	fARX	RBF	fARX-Lasso	CNN	fARX-EN	XGB	SVR-ARIMA	SOM-SVR	SVR	MLP	LSTM	GRU	DNN
TF																											
DSARIMA																											
WARIMA																											
WARIMA-RBF																											
DSHW	✓$_s$	✓$_s$	✓$_s$	✓$_s$																							
AR	✓	✓$_s$	✓$_s$	✓$_s$																							
ARIMA-GARCH	✓	✓	✓	✓																							
TBATS	✓$_s$	✓$_s$	✓$_s$	✓$_s$	✓$_s$	✓$_s$	✓$_s$																				
SNARX	✓	✓	✓	✓	✓$_s$	✓	✓$_s$																				
ARX	✓$_s$	✓$_s$	✓$_s$	✓$_s$	✓$_s$	✓$_s$	✓$_s$																				
TARX	✓$_s$	✓$_s$	✓$_s$	✓$_s$	✓$_s$	✓$_s$	✓$_s$			✓																	
DR	✓	✓	✓	✓	✓	✓$_s$	✓		✓$_s$																		
IHMARX	✓	✓	✓	✓	✓$_s$	✓$_s$	✓$_s$	✓$_s$	✓	✓																	
RF	✓	✓	✓	✓	✓	✓$_s$	✓	✓$_s$	✓	✓	✓$_s$	✓	✓$_s$														
fARX	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓$_s$	✓														
RBF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓$_s$													
fARX-Lasso	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓$_s$	✓$_s$												
CNN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓												
fARX-EN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓$_s$	✓		✓$_s$										
XGB	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓								
SVR-ARIMA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓$_s$	✓	✓	✓$_s$		✓$_s$								
SOM-SVR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓$_s$	✓	✓	✓	✓$_s$	✓$_s$								
SVR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓$_s$								
MLP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓$_s$	✓$_s$	✓$_s$	✓$_s$	✓$_s$						
LSTM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓$_s$	✓$_s$	✓$_s$	✓$_s$	✓$_s$				
GRU	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓$_s$	✓$_s$	✓	✓	✓				
DNN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓$_s$	✓$_s$	

From Figure 4.4 representing the DNN results we can observe how, except for the LSTM and GRU models, for any other forecaster the DNN is at least significantly better at one hour of the day and never significantly worse. In other words, the DNN is statistically significantly better than all other models except the LSTM and GRU forecasters. When compared with these two, while the DNN shows an overall accuracy that is statistically significantly better, the LSTM's accuracy is better at hours 01:00 and 22:00, and the GRU's accuracy at hours 01:00, 02:00, 03:00, and 06:00.

From Figure 4.5 representing the GRU results we can draw similar conclusions. In particular, the GRU model is statistically significantly better than all models except the

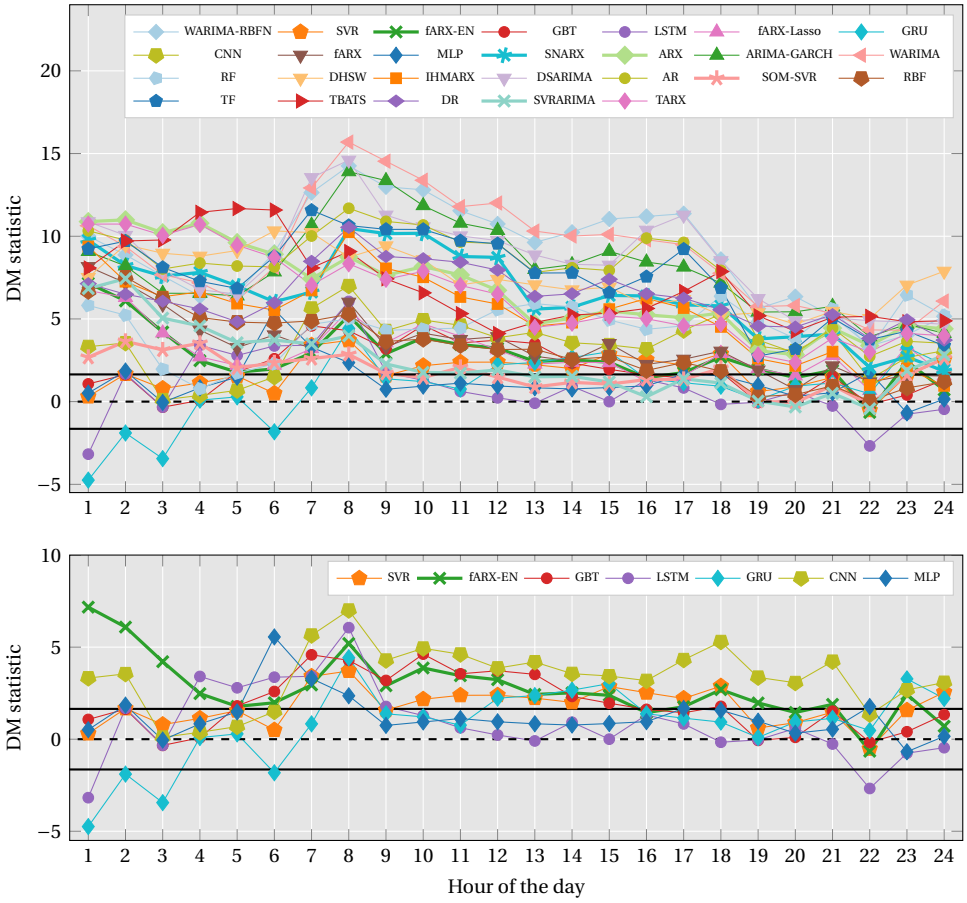


Figure 4.4: DM results for the DNN model. **Top**: test results for all 26 models. **Bottom**: test results for the top performing models. Values above the top dashed line represent cases where, with a 95% confidence level, the DNN is significantly better. Similarly, values below the lower dashed line accept at a 95% confidence level that the DNN is significantly worse.

DNN, LSTM, GRU, MLP, XGB and fARX-EN. However, for the XGB and fARX-EN models, while their accuracy is statistically significantly better at one hour, the GRU has an overall accuracy that is significantly better. From Figure 4.6 representing the LSTM results, we can draw similar conclusions as the ones obtained from Figure 4.5.

For the sake of simplicity, Table 4.8 only represents a summary of all the performed DM tests; particularly, as a total of 17550 DM tests were performed ($\frac{27!}{25!2!}$ model pairs × 50 DM test per model pair), it is impossible to list them all here or in an appendix. To address that, we have provided all the DM test results as an external online dataset [134].

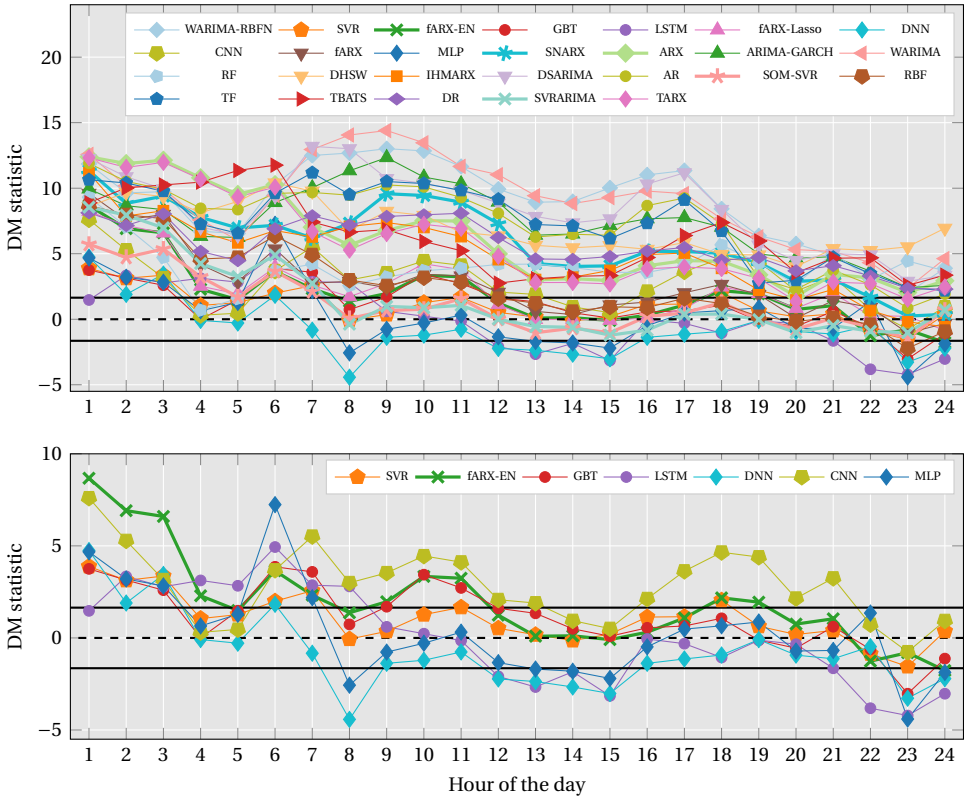


Figure 4.5: DM results for the GRU model. **Top**: results for all 26 models. **Bottom**: results for the top performing models. Values above the top dashed line represent cases where, with a 95% confidence level, the GRU is significantly better. Similarly, values below the lower dashed line accept at a 95% confidence level that the GRU is significantly worse.

4.5. DISCUSSION

To discuss the obtained results, we distinguish between three different topics: an analysis specific to the proposed DL models, an evaluation of the general results of the benchmark study, and a discussion on why neural networks have usually failed to predict electricity prices but in this chapter they represent the best model.

4.5.1. DL MODELS

From the results and observations that are drawn in the previous section, we can conclude that the proposed DNN, GRU, and LSTM models are the best alternative for forecasting day-ahead prices in the Belgian market. In particular, the benchmark is quite large and these 3 models outperform all the rest in a statistically significant manner.

Moreover, while the DNN is significantly better than the GRU and LSTM forecasters,

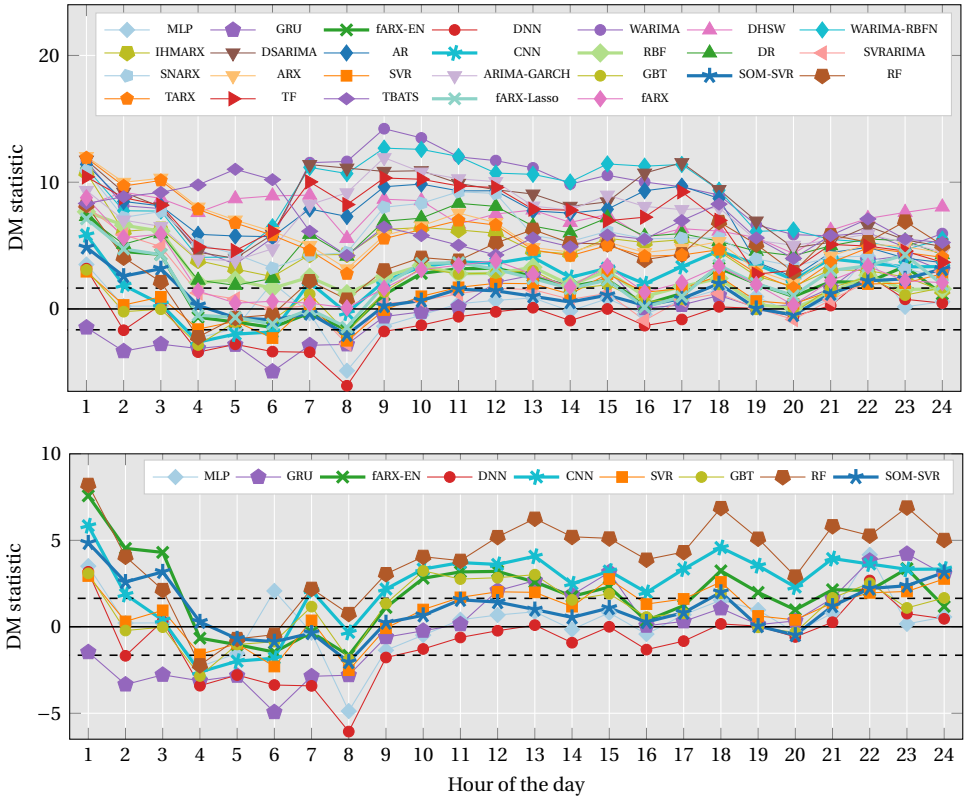


Figure 4.6: DM results for the LSTM model. **Top**: results for all 26 models. **Bottom**: results for the top performing models. Values above the top dashed line represent cases where, with a 95% confidence level, the LSTM is significantly better. Similarly, values below the lower dashed line accept at a 95% confidence level that the LSTM is significantly worse.

these two are better at some specific hours. Therefore, if a highly accurate system is targeted, e.g. by combining several forecasts, the three DL models are still necessary. However, if a single model is to be used, e.g. due to limitations in computation, the DNN is clearly the forecaster of choice.

Something that is interesting to discuss is the reason why the GRU, LSTM, and CNN models perform worse than the DNN. Particularly, all four of them are deep structures with the potential to model complex nonlinear patterns and, in the case of the GRU and LSTM models, they are especially appropriate for modeling time series data. So, how can it be that the DNN has an accuracy that is statistically significantly better? There are two possible hypotheses:

1. The amount of data: DL models require large amounts of data to be properly trained. When comparing the four DL models, the DNN has fewer parameters

than the other three; as a result, it might be easier to train. This hypothesis also agrees with the fact that the CNN performance is the worse of the four as it is the model with the largest number of parameters.

2. The structure of the networks: The GRU, LSTM, and CNN models separate the data corresponding to the day-ahead and the past data in two different networks. As a result, if some past data and day-ahead data are heavily related, none of the three structures is able to build these relations properly. By contrast, the DNN model makes no assumption about the input data and allows any possible relation to be built.

It is important to note that these are just hypotheses and that further research is necessary to properly explain this effect.

The last finding worth to discuss is the performance of the CNN as the proposed CNN model performs no better than simpler machine learning methods like XGB or SVR. An extra hypothesis to explain this effect (besides the amount of data or the separation of input data) is the fact that the CNN uses local operations. In particular, given some layer, the CNN does not interrelate all its values when making the connections to the next layer, but performs local convolution operations that interrelate local groups of data. As a result, while this structure is very convenient to process some specific type of data, e.g. pictures, it might not be appropriate if all the input data are highly correlated, e.g. seasonal time series data like electricity prices.

4.5.2. BENCHMARK

Regarding the benchmark results, besides being the proposed DNN, GRU, and LSTM models the best forecasters, several other effects need to be discussed.

MACHINE LEARNING VS. STATISTICAL METHODS

One of the most important effects to be examined is the fact that machine learning methods clearly outperform statistical methods. In particular, while several past studies led to empirical results that showed that the accuracy of machine learning methods was not better than the one of statistical methods, we can clearly observe that this is not the case in the EPEX-Belgium market. Possible explanations for this effect can be the following: as before, the market under study has large nonlinearities and spikes, and thus, it requires complex nonlinear models to accurately forecast the prices. In addition, the computational power has dramatically increased in the recent years, and thus, more data can be used for parameter estimation and the structure of the considered machine learning methods can be more complex. The latter argument also agrees with the fact that DL models have the best performance.

FARX-BASED MODELS

An exception to the previous statement are the fARX-based models: despite being statistical methods, they clearly perform better than any other statistical method and even better that some machine learning algorithms. These results confirm the findings of [232] and show that this model is one of the best statistical methods for predicting electricity prices. A possible explanation for this performance is the combination of two characteristics:

1. The structure of these models is very general and includes many possible exogenous inputs, which makes them very flexible.

2. At the same time, they use automatic feature selection to reduce the model complexity and to make the models tailored to the market under study.

MOVING AVERAGE MODELS

Another effect worth discussing is the fact that statistical models with moving average terms perform worse than their AR-counterparts. As the moving average terms provide an additional resource to model error correlation, they should have the potential to be more accurate. However, if we consider the structure of the model estimation, we observe that moving average terms might not necessarily improve the accuracy: as the moving average term leads to models that are estimated using non-convex optimization, the global minimum is not guaranteed and the resulting models might have a lower performance. In this context, as day-ahead prices are very volatile and may have price spikes, the resulting optimization problems might be highly non-convex and the global minimum might be hard to reach. Hence, although the obtained results could seem surprising at first, they might actually be expected.

HYBRID MODELS

A third important consideration is the fact that, in general, hybrid models do not outperform their regular versions. In particular, neither SVR-ARIMA nor SOM-SVR outperform the simpler SVR model. Likewise, WARIMA-RBF does not outperform the simpler WARIMA. An exception might be the ARIMA-GARCH, which outperforms the WARIMA and DSARIMA models.

TBATS

A final remark to be made is the fact that the TBATS model is clearly the best choice to predict prices when no regressors are available, and it even is a good choice when exogenous inputs exist. This observation is very important as, to the best of our knowledge, nobody has ever tested the accuracy of the TBATS model for predicting day-ahead electricity prices.

4.5.3. WHY DO THE PROPOSED MODELS IMPROVE THE PERFORMANCE?

When we consider the literature of electricity price forecasting, there are many examples where neural networks have been outperformed by other forecasters [8, 54, 125, 151, 182, 191, 210, 233, 252]. The results obtained in this chapter lead to the opposite conclusion: in this case study, neural networks outperform all other models. In this section, to clarify this discrepancy, we provide the rationale behind the superior performance of the proposed DL models. In particular, we examine four features that past studies have typically not considered and we argue that by not considering them the accuracy worsens.

DEPTH

As briefly motivated in the introduction, deep neural networks can generalize and obtain better results than their shallow counterparts. This effect is related to the universal approximation theorem [107], which states that a neural network with a linear output layer

can approximate any continuous function on a compact subset of $\mathbb{R}^n$ provided that it has enough neurons, but does not indicate whether this number is tractable [86]. In particular, to approximate some families of functions, the number of neurons required by a shallow network can grow exponentially with the input size and in turn become intractable [86]. In the same context, the family of functions could be approximated by a tractable number of neurons if the depth is larger than some threshold number d [86].

As a result, when approximating a target function, a deep network might need a much smaller number of neurons than its shallow counterpart, and thus, it might be able to approximate the function more easily and better. In our case study, this effect is observed in Table 4.7, where a shallow neural network, i.e. the MLP model, has a lower accuracy than the DNN, GRU, and LSTM models.

If we look now at the literature of electricity price forecasting, the neural networks that have usually been proposed have been shallow networks [8, 39, 54, 125, 182, 191, 210, 233, 252]. Therefore, considering the above argument, it might be normal for them to perform worse that the deeper models than we propose.

NUMBER OF NEURONS

Intrinsically related to the previous argument and with the universal approximation theorem [107] is the fact that, in order for a network to correctly approximate a function, the number of neurons needs to be large enough. However, when we consider the literature of electricity price forecasting, most of the studies have employed small MLPs with less than 50 neurons [39, 54, 123, 125, 175, 182, 191, 210, 233] and have not performed any hyperparameter optimization to select the required number of neurons.

If this case study, the empirical results show that the optimal number of neurons for the MLP model is 117 (see Appendix A.2). Although the optimal number will change from case to case, we can use it as a reference to argue that the small-sized neural networks previously proposed in the literature might not be large enough to model the complex dynamics of electricity prices.

To strengthen our argument, we analyze this effect in our case study: we also consider a MLP with 50 neurons and we compare its performance against the optimized MLP using 117 neurons. As it would be expected, the MLP with 50 neurons fails to perform as good as the optimized one: its accuracy on the test set drops from 13.27% to 14.30% sMAPE and this difference in accuracy is statistically significantly for all 24 hours. In addition, another finding that reinforces our argument is the fact that, if we were to use this smaller MLP in the benchmark it would not be better than half of the models, which would agree with the literature results.

SIZE OF TRAINING DATASET

Even if the network is large enough to approximate the targeted function, the optimizer might fail to estimate the right parameters [86]. A possible problem that the optimizer might face is not having enough training data to estimate the large number of parameters in a neural network, e.g. in our MLP model with 117 neurons there are approximately 28200 parameters.

When we examine the literature of electricity price forecasting, studies have usually considered networks that were trained using 1 year of data or less [8, 39, 54, 125, 151, 182, 191, 210, 233, 252]. If we consider our case study, that might not be enough: if trained

with 1 year of data, the accuracy of the DNN drops from 12.34% to 13.27% sMAPE, an accuracy that is worse than the performance of many benchmark models, and which might explain again some of the literature results.

STOCHASTIC GRADIENT DESCENT

A second problem that might also affect the parameter estimation regards the properties of the optimization algorithm itself. In particular, in the literature on electricity price forecasting, network parameters have traditionally been estimated using standard gradient descent methods, e.g. batch gradient descent (also known as back-propagation) or Levenberg–Marquardt optimization [8, 39, 54, 123, 125, 175, 210, 243]. Although these methods might work well for small sized-networks, they display computational and scalability issues and they often obtain worse results [147].

A better alternative is the family of stochastic gradient descent methods [86, 147], which, instead of computing the gradient w.r.t. to the whole training dataset, do it w.r.t. to subsets of it. In our case study, if batch gradient descent is used instead of Adam, the accuracy of the DNN drops from 12.34% to 14.15%. Based on this empirical result and the argument above, it is clear that this effect might also account for some of the discrepancies between our work and the literature.

4.6. CONCLUSIONS

IN this chapter, four different *deep learning (DL)* models to predict day-ahead electricity prices have been proposed. Moreover, a large benchmark study has been set up in order to compare the predictive accuracy of the proposed models w.r.t. to other models in the literature. This benchmark is selected to comprise as many models as possible and to serve as a reference within the field of day-ahead electricity price forecasting.

Three of the four proposed DL forecasters, i.e. the *deep neural network (DNN)* model, the *long-short term memory (LSTM)* model, and the *gated recurrent unit (GRU)* model, are shown to obtain a predictive accuracy that is statistically significantly better than all other models. In addition, among these three models, the DNN is able to outperform the other two with a difference in accuracy that is statistically significant. Despite this difference, the three models are necessary to obtain a high-performing forecaster, as the accuracy of the GRU and LSTM models still better at some specific hours.

Among the rest of the forecasters, a clear division is observed between machine learning and statistical methods, where the former display an accuracy that is statistically significantly better. In addition, models with moving average terms are shown to have the worst performance and hybrid methods are shown not to outperform their simpler counterparts.

In future work, this research will be expanded to more advanced DL techniques, e.g. autoencoders, and the use of expert advice to combine the individual benchmark models. In addition, we will explore the usage of these forecasting techniques for improving the profits in electricity trading. As starting point, in Chapter 8, we propose novel control algorithms for demand response via electricity trading that make use of the proposed forecasting techniques to accurately model price uncertainty.

5

A GENERALIZED MODEL FOR FORECASTING SOLAR IRRADIANCE

General notions are generally wrong.

Mary Wortley Montagu

Occurrences in this domain (weather) are beyond the reach of exact prediction because of the variety of factors in operation, not because of any lack of order in nature.

Albert Einstein

Due to the increasing integration of solar power into the electrical grid, forecasting short-term solar irradiance has become key for many applications. In this context, as solar generators are geographically dispersed and ground measurements are not always easy to obtain, it is very important to have general models that can predict solar irradiance without the need of local data. In this chapter, a model that can perform short-term forecasting of solar irradiance in any general location without the need of ground measurements is proposed. To do so, the model considers satellite-based measurements and weather-based forecasts, and employs a deep neural network that is able to generalize across locations. Particularly, the network is trained using only a small subset of sites where ground data are available, and the model is able to generalize to a much larger number of locations where ground data do not exist. As a case study, 25 locations in The Netherlands are considered and the proposed method is compared against four local models that are individually trained for each location using ground measurements. Despite the general nature of the model, it is shown show that the proposed model is equal to or better than the local models.

Parts of this chapter have been published in [132, 133].

5.1. INTRODUCTION

WITH the increasing integration of renewable sources into the energy mix, forecasting the generation of renewable sources has turned into a very important challenge within the energy transition. In this context, as solar energy is one of the most unpredictable sources, forecasting solar irradiance over short time horizons has become specially critical. Particularly, short-term forecasts of solar irradiance are paramount for activation of reserves, operational planning, switching sources, programming backup, short-term power trading, peak load matching, scheduling of power systems, congestion management, and cost reduction [97, 196, 236].

Despite its importance, the methods proposed so far for short-term forecasting of solar irradiance have a large limitation: they all need past ground measurements of the solar irradiance at every site where a forecast is needed [65]. Since solar generators are geographically dispersed, if ground measurements of all these sites are required, the cost of forecasting irradiance can become very expensive and obtaining the required data can become very challenging. Therefore, in order to obtain scalable solutions for solar irradiance forecasting, global models that can forecast without the need of local data are needed. In this context, although current cloud-moving vector models might accomplish that, they are not always easy to deploy as they are complex forecasting techniques that involve several steps [65].

CONTRIBUTIONS AND ORGANIZATION OF THE CHAPTER

To fill this scientific gap, in this chapter we propose a novel forecasting technique that, in addition to being accurate and easy to deploy, forecasts the *global horizontal irradiance (GHI)* in any general location without the need of ground measurements[1]. The prediction model is based on a *deep neural network (DNN)* that, using SEVIRI[2] satellite images and *numerical weather prediction (NWP)* forecasts, is as accurate as local time series models that consider ground measurements. While the model uses satellite images just as cloud-moving vector models do, it is easier to deploy as it requires less complex computations. In addition, although obtaining satellite data might not always be easier or cheaper than installing local ground sensors, there are several locations where satellite data are available and the proposed model avoids going to the ground to install local measurement equipment.

As a case study, 30 location in The Netherlands are considered and the model is estimated using 5 of these locations. Then, for the remaining 25 locations, the performance of the proposed estimated model is compared against individual time series models specifically trained for each site using ground data.

The chapter is organized as follows: Section 5.2 presents the proposed general model for forecasting solar irradiance. Next, Section 5.3 introduces the case study and discusses the performance of the proposed model when compared with local models. Finally, Section 5.4 summarizes the main results and concludes the chapter.

[1] Note that, while the method could also be used for forecasting the direct normal irradiance [146], the method is tailored for the GHI as the direct normal irradiance is just a component of the GHI.

[2] The *SEVIRI (Spinning Enhanced Visible and InfraRed Imager)* is a measurement instrument of the METEOSAT satellite.

5.2. PREDICTION MODEL

In this section, the proposed prediction model for solar irradiance forecasting is presented. The proposed model is based on a DNN that uses satellite images and weather forecasts to replace ground measurements.

5.2.1. MODEL STRUCTURE

A key element to build a prediction model that can be used without the need of ground data is to employ a model whose structure is flexible enough to generalize across multiple geographical locations. As DNNs are models that can generalize across tasks [86] (see Section 5.2.5 for more details), they are selected as the base model for the proposed forecaster. In this context, to maximize the forecasting accuracy, the structure of the forecasting model and the type of inputs are optimally selected. Likewise, the outputs of the model are carefully designed.

In particular, in terms of the outputs, the model consists of 6 output neurons representing the forecasted hourly irradiance $\hat{\mathbf{I}}_h = [\hat{I}_{h+1}, \dots, \hat{I}_{h+6}]^\top$ at every hour h over the next 6 hours. This horizon is the standard choice for short-term irradiance forecasting [65][3]. In terms of the structure, the model is not subject to any specific depth; instead, depending on the case study, i.e. the geographical area where the forecasts are made, the number of hidden layers are optimized using the hyperparameter optimization described in Section 2.6. To select the number of neurons per layer, the same methodology applies, i.e. they need to be optimized for each geographical location.

5.2.2. MODEL INPUTS

As for the inputs, the aim of the model is to forecast solar irradiance without the need of ground data. Thus, to perform the selection of model inputs, the model considers the subset of inputs that, while correlating with solar irradiance, are general enough so that they can be easily obtained for any given location. Given that restriction, the proposed model considers three types of inputs: NWP forecasts of the solar irradiance, the clear-sky irradiance, and satellite images representing maps of past solar irradiance.

NUMERICAL WEATHER PREDICTION FORECAST

The first type of input are NWP forecasts of the solar irradiance obtained from the *European center for medium-range weather forecasts (ECMWF)*. As indicated in the introduction, NWP forecasts of the solar irradiance are less accurate than time series models for short-term horizons. However, as they strongly correlate with the real irradiance, they are very useful regressors to build time series models.

For the proposed model, at every hour h, the input data consist of the forecasted irradiance values $\hat{\mathbf{I}}^{\mathrm{E}}_h = [\hat{I}^{\mathrm{E}}_{h+1}, \dots, \hat{I}^{\mathrm{E}}_{h+6}]^\top$ over the prediction horizon given by the latest available ECMWF forecast (typically available every day around 08:00-09:00 CET).

CLEAR-SKY IRRADIANCE

As second input, at every hour h, the model considers the clear-sky irradiance $\mathbf{I}^{\mathrm{c}}_h = [I^{\mathrm{c}}_{h+1}, \dots, I^{\mathrm{c}}_{h+6}]^\top$ over the prediction horizon. The clear-sky irradiance is a deterministic input representing the GHI under clear-sky conditions. For the proposed model,

[3] 6 hours is the limit prediction horizon before NWP forecasts outperform time series models [65].

the clear-sky irradiance is obtained using the clear-sky model defined in [114], which computes $\mathbf{I}_h^c$ using the location and time of interest.

SATELLITE IMAGES

The third input are satellite data $\mathbf{I}_h^S = [I_h^S, I_{h-1}^S, \ldots, I_{h-n}^S]^\top$, representing the past irradiance values in the geographical area. In particular, the input data consist of images from the SEVIRI instrument of the METEOSAT satellite that are transformed to irradiance values using two different methods:

1. For data corresponding to solar elevation angles above 12°, the SEVIRI-based images are mapped to irradiance values using the *Surface insolation under clear and cloudy skies (SICSS)* algorithm [89].

2. For data corresponding to solar elevation angles below 12°, i.e. very early in the morning and late in the evening, the irradiance values are extracted by considering the interpolation method described in [64] applied to the clear-sky index.

This distinction depending on the solar elevation angle is required because the SICSS method considers cloud properties and at low solar elevation angles the uncertainty in the cloud properties increases strongly [64].

Once the satellite images are mapped to irradiance values, the input data simply consist of the past irradiance values in the individual pixel where the forecasting site is located. Then, to select which past irradiance values, i.e. which past images, are relevant for building the general model, the feature selection method defined in Chapter 3 is employed.

As a final remark, it is important to note that these irradiance values have a resolution that is limited by the resolution of the satellite images, which in the case of the SEVIRI instrument are pixels of 3×3 km. Therefore, even if the satellite-based irradiance values were perfectly accurate, they would only represent the average irradiance in the $9\,\text{km}^2$ area corresponding to each pixel. As a result, to represent the solar irradiance in a specific location, the accuracy of satellite-based measurements cannot be better than that of ground measurements.

INPUT SELECTION

The three input features that the proposed model considers were selected from a larger set of input features. In detail, in order to ensure that the proposed model includes the most relevant input features, a feature selection process was performed. During this feature selection process, the three considered inputs, i.e. the NWP forecasts, the clear-sky irradiance, and the satellite images were selected as the most important features. However, in addition to these three, four other features were also considered: (i) historical values of the temperature, (ii) historical values of the humidity, (iii) forecast of the temperature, and (iv) forecast of the humidity.

To perform the feature selection between these 7 input features, the feature selection method described in Chapter 3 was employed; i.e. the 7 input features were modeled as binary hyperparameters and the selection was performed together with the hyperparameter optimization described in Section 5.2.3. This optimization resulted in the 3 selected inputs: the NWP forecasts, the clear-sky irradiance, and the satellite images

5.2.3. HYPERPARAMETER OPTIMIZATION AND FEATURE SELECTION

As introduced in Section 5.2.1, the proposed model needs to be tuned for the specific geographical area where it is applied. To do so, the following four DNN hyperparameter are optimized:

1. **Number of hidden layers**: The neural network depth is a parameter that needs to be tuned in order to obtain a model that can correctly generalize across multiple geographical locations.

2. **Number of neurons per layer**: Besides the number of hidden layers, the size of each layer also plays an important role in the generalization capabilities of the DNN.

3. **Initial learning rate**: Although the stochastic gradient descent method automatically adapts the learning rate at every iteration of the optimization process, the learning rate at the first iteration has to be selected.

4. **Dropout**: Dropout [217] is included as a possible regularization technique to reduce overfitting and to improve the training performance. To do so, at each iteration, dropout selects a fraction of the neurons and prevents them from training. This fraction of neurons is defined as a real hyperparameter between 0 and 1.

As explained in Section 5.2.2, in combination with the hyperparameter optimization, the proposed model also performs a feature selection. Particularly, the feature selection method selects the most relevant inputs among a subset of 7 features and it also selects which past historical irradiance values are required.

5.2.4. MODEL EQUATIONS

Using the notation for DNNs defined in Section 2.5.2, the equations of the DNN model assuming two hidden layers can be defined as:

$$z_{1,i} = f_{1,i}\left(\mathbf{w}_{\mathrm{i},i}^{\top} \cdot \mathbf{x} + b_{1,i}\right), \qquad \text{for } i = 1,\dots n_1, \tag{5.1a}$$

$$z_{2,i} = f_{2,i}\left(\mathbf{w}_{2,i}^{\top} \cdot \mathbf{z}_1 + b_{2,i}\right), \qquad \text{for } i = 1,\dots n_2, \tag{5.1b}$$

$$\hat{I}_{h+i} = \mathbf{w}_{\mathrm{o},i}^{\top} \cdot \mathbf{z}_2 + b_{\mathrm{o},i}, \qquad \text{for } i = 1,\dots 6. \tag{5.1c}$$

For the proposed model, the *rectified linear unit (ReLU)* [167] is selected as the activation function f_{ki} of the hidden layers. This choice is made because this activation function has become a standard for hidden layers of DNNs [86]. It is important to note that, as the irradiance is a real number, no activation function is used for the output layer.

5.2.5. TRAINING

The DNN is trained by minimizing the mean square error[4]. In particular, given the training set $\mathcal{S}_{\mathrm{T}} = \{(\mathbf{x}_k, \mathbf{I}_k)\}_{k=1}^{N}$, the optimization problem that is solved to train the neural net-

[4] Note that minimizing the mean square error is equivalent to minimizing the rRMSE metric used throughout the chapter to evaluate and compare the forecasting models.

work is given by:

$$\underset{\mathbf{w}}{\text{minimize}} \quad \sum_{k=1}^{N} \|\mathbf{I}_k - F(\mathbf{x}_k, \mathbf{w})\|_2^2, \tag{5.2}$$

where $F : \mathbb{R}^n \to \mathbb{R}^6$ is the neural network map and $\mathbf{w}$ is the set comprising all the weights of the network.

GENERALIZING ACROSS GEOGRAPHICAL SITES

A key element for the model to forecast without the need of ground data is to be able to generalize across locations. To do so, the proposed model is trained across a small subset of sites so that the model learns to generalize across geographical sites. It is important to note that, while ground data are required for this small subset of locations, the model generalizes across all other geographical locations where ground data are not needed. Moreover, as it is shown in the case study for The Netherlands (see Section 5.3), the number of locations where ground data are required is relatively small, e.g. 3-5 sites.

5

GENERALIZING ACROSS PREDICTION HORIZONS

Enforcing generalization is not only good for obtaining a model that does not require ground data, but in general, it is also beneficial to obtain a DNN that does not overfit and that obtains more accurate predictions [86]. In particular, as it has been empirically shown in Chapter 3, by forcing the network to solve multiple related tasks, e.g. forecasting multiple sites, the network might learn to solve individual tasks better.

Therefore, to further strengthen the generalization capabilities of the network, instead of training a different DNN for each hour h of the day, we employ a single DNN for all hours. As with the geographical site generalization, the goal is to build a DNN that, by performing several related tasks, is able to learn more accurate predictions.

IMPLEMENTATION DETAILS

The optimization problem is solved using Adam [127], a version of stochastic gradient descent that computes adaptive learning rates for each model parameter. The use of adaptive learning rates is selected for a clear reason: as the learning rate is automatically computed, the time needed to tune the learning rate is smaller in comparison with other optimization methods. Together with Adam, the forecaster also considers early-stopping [253] to avoid overfitting (see Section 4.4.2).

5.2.6. REPRESENTATION

Using the definitions above, the forecasting model can be represented as in Figure 5.1. As can be seen, for every hour h, the model forecasts the irradiance $[I_{h+1}, \ldots, I_{h+6}]^\top$ over the prediction horizon. As defined in Section 5.2.2, the model inputs include the clear-sky irradiance $\mathbf{I}_h^{\text{c}} = [I_{h+1}^{\text{c}}, \ldots, I_{h+6}^{\text{c}}]^\top$, the ECMWF forecast $\hat{\mathbf{I}}_h^{\text{E}} = [\hat{I}_{h+1}^{\text{E}}, \ldots, \hat{I}_{h+6}^{\text{E}}]^\top$, and the optimal past irradiance inputs[5] $\mathbf{I}_h^{\text{S}} = [I_h^{\text{S}}, \ldots, I_{h-n}^{\text{S}}]^\top$.

[5] Note that, as the past irradiance inputs have to be optimized for each case study, in this example they are simply represented by a general vector of past irradiance values. Similarly, for the sake of simplicity, it is assumed that the optimal depth of the DNN model is 2 hidden layers.

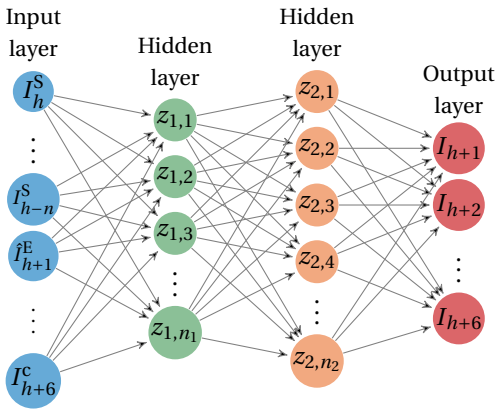


Figure 5.1: DNN to forecast solar irradiance.

5

5.3. CASE STUDY

IN order to evaluate the proposed model, 30 sites in the Netherlands are considered and the accuracy of the proposed model is compared with that of specific models individually trained using local data.

5.3.1. DATA DESCRIPTION

The dataset spans four years, i.e. from 01/01/2014 until 31/12/2017, and comprises, for each of the 30 sites, the following four types of input data:

- The historical ground data $\mathbf{I}^G$ measured on site.
- The satellite-based irradiance values $\mathbf{I}^S$.
- The daily ECMWF forecasts $\hat{\mathbf{I}}^E$.
- The deterministic clear-sky irradiance $\mathbf{I}^c$.

In all four cases, these data represent hourly average values between two consecutive hours, i.e. a variable given at a time step h represents the average variable in the time interval $[h, h+1)$.

DATA SOURCES

For the irradiance values obtained from SEVIRI satellite images, the processed irradiance values are directly obtained from the *Royal Netherlands Meteorological Institute (KNMI)* via their Cloud Physical Properties model [201]. The ECMWF forecasts are directly obtained through the ECMWF website [74]. The clear-sky irradiance is obtained via the `python` library `PVLIB` [12], which implements the clear-sky model [114] (see Section 5.2.2).

For the ground measurements, 30 of the meteorological stations in The Netherlands that are maintained by the KNMI [201] and that measure irradiance values using pyranometers are considered. In particular, the following 30 stations are employed: Arcen,

Berkhout, Cabauw, De Kooy, De Bilt, Deelen, Eelde, Eindhoven, Ell, Gilze-Rijen, Heino, Herwijnen, Hoek van Holland, Hoogeveen, Hoorn (Terschelling), Hupsel, Lauwersoog, Leeuwarden, Lelystad, Maastricht, Marknesse, Nieuw Beerta, Rotterdam, Schiphol, Stavoren, Twente, Vlissingen, Volkel, Westdorpe, and Wijk aan Zee. The geographical location of these 30 stations is illustrated in Figure 5.2.

DATA DIVISION

In order to perform the study, the data are divided into three subsets:

1. Training set (01/01/2014 to 31/12/2015): These 2 years of data are used for training and estimating the various models.

2. Validation set (01/01/2016 to 31/12/2016): A year of data is used to select the optimal hyperparameters and features and to perform early-stopping when training the network.

3. Test set (01/01/2017 to 31/12/2017): A year of data that is not used at any step during the model estimation process, is employed as the out-of-sample data to compare the proposed model against local models.

In addition to the time separation, the data are further divided according to the location:

1. Of the 30 sites, 5 are used to train the proposed models. In particular, the following 5 were randomly selected: Herwijnen, Wijk aan Zee, Schiphol, Twente, and Lelystad.

2. The remaining 25 act as out-of-sample data to show that the model can predict irradiance at any site without the need of local data.

This separation is depicted in Figure 5.2, which represents the geographical distribution of the 30 sites distinguishing between training and test sites.

In short, the proposed model is trained using data from 5 sites spanning three years and it is evaluated in 25 additional locations and using an additional year of data. It is important to note that this separation in 5+25 locations only applies for the proposed model. Particularly, for the local models used as benchmark, the data division is only performed as a function of time as, by definition, each local model considers only local data.

DATA PREPROCESSING

To evaluate the proposed models, the hours of the day for which the irradiance is very small, i.e. the hours representing the sunset and sunrise, are disregarded. This is done by discarding those hours that correspond with solar elevation angles below 3°. This limitation on the solar elevation angles implies that the number of daily forecasts changes throughout the year; e.g. while in June the model makes 11-12 forecasts per day, in January that number is reduced to 3-4.

In addition to the above preprocessing step, the hourly time slots that have missing values are also discarded.

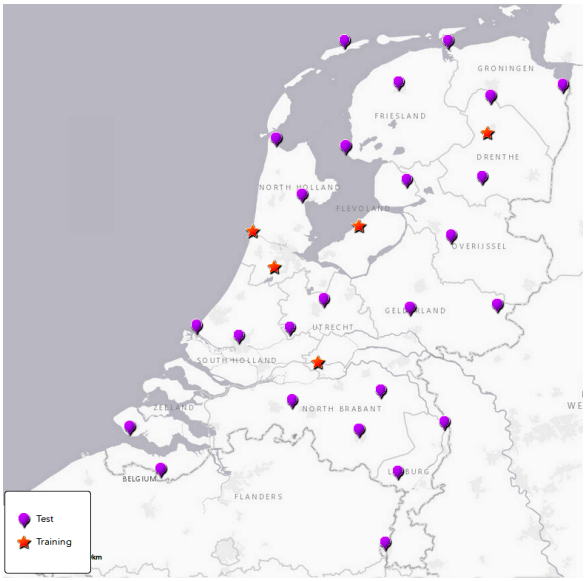


Figure 5.2: Geographical distribution of the 30 sites in the case study. The orange dots are the 5 sites used for estimating the model. The purple dots represent the 25 out-of-sample sites to evaluate the model.

5.3.2. LOCAL MODELS

To compare the proposed forecaster, four types of local models are considered: a persistence model [65], an *autoregressive model with exogenous inputs (ARX)* [145], a *gradient boosting tree (GBT)* algorithm [42], and a local neural network [145].

Moreover, in addition to the local models, the ECMWF forecast is also included in the benchmark. By doing so, the accuracy between the time series models and the NWP forecast can be compared as a function of the prediction horizon.

PERSISTENCE MODEL

When evaluating a new model, a standard approach in the literature of irradiance forecasting [65] is to check whether the new model provides better predictions than a trivial model. The trivial model that is normally used is a persistence model, which assumes that the clear-sky index does not change from one time interval to the other [65]. Defining the clear-sky index at hour h by:

$$k_h^c = \frac{I_h}{I_h^c}, \tag{5.3}$$

with I_h and I_h^c respectively representing the measured irradiance and the clear-sky irradiance at the same hour h, the persistence model predicts the irradiance I_{h+p} as:

$$\hat{I}_{h+p} = k_h^c I_{h+p}^c = \frac{I_h}{I_h^c} I_{h+p}^c. \tag{5.4}$$

Note that the above is equivalent to saying that I_{h+p} is the same as I_h but scaled by the irradiance diurnal cycle defined by the deterministic clear-sky irradiance.

LINEAR MODEL

Another standard benchmark choice in the literature of irradiance forecasting are autoregressive linear models [65, 145]; hence, the second model considered in the comparison is a linear autoregressive model that can optimally select its exogenous inputs. As the model is local, a different model per location, per hour of the day h, and for prediction time $h+p$ is considered. Therefore, as the proposed model is evaluated in 25 locations, 6 forecasts per day are made, and each forecast is made for 6 prediction times, a total of $25 \times 6 \times 6 = 900$ models are estimated.

The exogenous inputs of these models are similar to the DNN, but instead of using the satellite irradiance maps $\mathbf{I}_h^{\mathrm{S}}$, the models consider the local irradiance ground measurements $\mathbf{I}_h^{\mathrm{G}}$. In particular, each of the 900 models considers the clear-sky irradiance I_{h+p}^{c} and the ECMWF forecast $\hat{I}_{h+p}^{\mathrm{E}}$ at the specific prediction time $h+p$. Additionally, as with the global model and $\mathbf{I}_h^{\mathrm{S}}$, each model optimizes the lagged irradiance values $\mathbf{I}_h^{\mathrm{G}}$ using the feature selection method described in Chapter 3.

GRADIENT BOOSTING TREE

As a third model, the XGBoost algorithm [42] is considered, a GBT model that predicts data by combining several regression trees. The model is based on the principle of boosting [100], i.e. combining models with high bias and low variance in order to reduce the bias whilst keeping a low variance.

It is important to note that, while several models based on regression trees have been proposed in the literature for forecasting solar irradiance [236], the XGBoost algorithm has, to the best of our knowledge, not yet been used. Nevertheless, including this model in the benchmark was decided for two reasons. First, it has been shown to outperform other regression tree methods and has recently become the winner of several challenges in Kaggle, a site that hosts machine learning competitions [42]. Second, it has been successfully used in other energy-based forecasting applications, e.g. forecasting electricity prices [135] (see also Chapter 4).

As with the linear model, a different GBT per location, hour, and prediction time is estimated; i.e. 900 different models are estimated. Similarly, the model inputs are the same as the linear models, i.e. the clear-sky irradiance I_{h+p}^{c} and the ECMWF forecast $\hat{I}_{h+p}^{\mathrm{E}}$ at the prediction time, and the historical irradiance values $\mathbf{I}_h^{\mathrm{G}}$ that are optimally selected.

It is important to note that, as done with the proposed DNN, all the GBT hyperparameters (see [42]) are optimally selected using the hyperparameter optimization algorithm defined in Section 2.6.

NEURAL NETWORK

As a fourth model, a local DNN that considers very similar inputs, outputs, structure, and training algorithm as the proposed global DNN is considered. The only difference w.r.t. to the proposed DNN is that it considers the local measurements of the irradiance

$\mathbf{I}_h^G$ instead of the satellite irradiance maps $\mathbf{I}_h^S$. Nonetheless, the inputs $\mathbf{I}_h^G$ are optimally selected as done for the global DNN.

The reason for including this model in the case study is that, similar to the linear and the persistence models, neural networks are a standard choice in the literature on solar irradiance forecasting [64, 236].

As the proposed DNN is evaluated in 25 sites and the model is local, 25 different local DNNs are estimated. Unlike the linear and GBT models, the same DNN is used for the different hours of the day; this was done because it was empirically observed that the distinction of a different DNN per hour of the day led to worse predictive accuracy.

5.3.3. HYPERPARAMETER OPTIMIZATION AND FEATURE SELECTION

In this section, we present the results of the hyperparameter and feature optimization for the global and local models.

GLOBAL MODEL

As defined in Section 5.2, the hyperparameters and input features of the global DNN are optimally selected according to the geographical location. The range of the hyperparameters considered in the optimization search and their obtained optimal values are listed in Table 5.1.

Table 5.1: Optimal hyperparameters for the global DNN.

Hyperparameter	Optimal Value	Search Range
Number of hidden layers	2	{1, 2, 3, 4}
Neurons in 1st layer	208	[100, 400]
Neurons in 2nd layer	63	[50, 150]
Initial Learning Rate	1.16×10^{-3}	$[10^{-4}, 10^{-2}]$
Dropout	0.14	[0, 1]

In terms of the lags of satellite irradiance values, the optimal input features at hour h are defined by the vector $\mathbf{I}_h^S = [I_h^S, \ldots, I_{h-3}^S, I_{h-18}^S, \ldots, I_{h-23}^S]^\top$, i.e. the irradiance values at lags 0, 1, 2, and 3 w.r.t. the current hour h and at lag 24 w.r.t the 6 prediction hours $h+1, \ldots, h+6$.

LOCAL MODELS

For the local models, the hyperparameters and input features are also optimized. However, considering that 900 linear models, 900 GBT models, and 25 local DNNs are used, displaying all their optimal hyperparameters and input features is out of the scope of this chapter. However, the main results can be summarized as follows:

1. In terms of the input features, all the local models consider a similar selection as the global DNN. In particular, replacing the satellite-based values $\mathbf{I}_h^S$ by ground irradiance values $\mathbf{I}_h^G$, all the local models consider the irradiance values at lags 0 and 1 w.r.t. the current hour h and at lag 24 w.r.t. the prediction hour $h+p$. In

addition, most of them also consider the irradiance values at lags 2 and 3 w.r.t. the current hour h; the exception are models that predict the solar irradiance at early hours when lags of 2-3 hours represent irradiance values of 0.

2. In the case of the local DNNs, the number of hidden layers is 2 for all 25 sites. Moreover, the number of neurons in the first (second) hidden layer varies from 95 to 242 (51 to 199) neurons depending on the site. Similarly, the dropout and the learning rate respectively oscillate between 0 and 0.45, and between 5.825×10^{-4} and 5.800×10^{-2}.

3. In the case of the GBT models, the range of the hyperparameters values varies in a larger range, e.g. the number of trees per model fluctuates between 10 and 1000 and the depth of each tree varies between 1 and 20.

5.3.4. OVERALL RESULTS

After defining the setup of the case study and describing the selection of hyperparameters and features, in this section the average performance of the global DNN is compared against that of the local models. Particularly, following the standard in the literature, the solar irradiance forecasting methods are evaluated using the three described metrics in Section 2.3.2; i.e. *relative root mean square error (rRMSE)*, forecasting skill s, and *mean bias error (MBE)*, across the 25 sites and the 6 prediction times. These average metrics are listed in Table 5.2, where the forecasting skill was computed using the same window length employed in [160], i.e. 200 samples[6].

Table 5.2: Comparison of the average predictive accuracy across sites and prediction times by means of rRMSE, forecasting skill s, and MBE.

Model	rRMSE [%]	s [%]	MBE [W/m^2]
Global DNN	31.31	22.42	-1.04
Linear	32.01	21.22	-1.07
Local DNN	32.10	19.29	-1.43
ECMWF	34.94	9.75	-2.52
GBT	35.85	9.92	1.50
Persistence	41.98	0	11.60

From Table 5.2, several observations can be drawn:

1. In terms of square errors, i.e. rRMSE, the predictive accuracy of the proposed global model is slightly better than all the local models and significantly better than some of them, especially the GBT model or the persistence model. Among the local models, both the linear and local DNN perform the best and the persistence model the worst.

[6] As in [160], the window length for which s was stable was analyzed. Similar to [160], 200 samples were found to be a reasonable value.

2. This same observation can be inferred from looking at the forecasting skill: the proposed global model performs similar to the linear model, slightly better than the local DNN, and much better than the other models. In addition, when compared across all sites and prediction horizons, all models perform better than the persistence model.

3. In terms of model bias, i.e. MBE, all models show a very small bias which indicates that the bias of the models is negligible. Particularly, considering that the average irradiance of the dataset is approximate 350 W/m^2, the bias of all the models is around 0.3-0.8% of the average irradiance, which represents a negligible bias. The exception to this is the persistence model, whose bias of 3% of the average irradiance is a bit larger, but still quite small.

5.3.5. COMPARISON WITH PREVIOUSLY VALIDATED FORECAST MODELS

While the proposed global model seems to be a good replacement of the local models, it is also very important to establish its quality w.r.t. previously validated forecast models from the literature. As explained in Section 2.3.2, while this comparison cannot fairly be done using a metric like rRMSE, it can be roughly assessed using the forecasting skill *s*. In particular, using the results of [160], we can establish a comparison between the proposed global model, the local NARX model proposed in [160], and the cloud motion forecast of [179]. As both models from the literature were originally only evaluated for 1-hour step ahead forecasts, we also limit the comparison of the global model to that interval. The comparison is listed in Table 5.3.

Table 5.3: Comparison of the average predictive accuracy between the global model, a NARX model from the literature, and a cloud moving forecast from the literature. The comparison is done for 1-hour ahead forecasts and by means of forecasting skill .

Model	s [%]
Global DNN	10
NARX [160]	12
Cloud moving [179]	8

What can be observed from these results is that the overall quality of the proposed global model for 1-hour ahead forecasts is very similar to that of the approaches from the literature. Therefore, as initially observed when comparing the average performance of the global model w.r.t. to the local model considered in this chapter, the proposed global model seems to be an excellent candidate to save the operational costs of installing local sensors and collecting ground measurements.

5.3.6. COMPARISON ACROSS PREDICTION HORIZONS

A third important step to analyze the performance of the proposed global model is to verify that its average performance is satisfied across all prediction horizons. In particular, it is important to check whether the global models can provide accurate predictions at all short-term horizons. To perform this comparison, the two metrics used for com-

paring predictive accuracy, i.e. rRMSE and the forecasting skill *s*, are evaluated for each benchmark model and prediction horizon. This comparison is listed in Table 5.4 and illustrated in Figure 5.3.

Table 5.4: Comparison of the predictive accuracy of the various forecasters across the 6 prediction horizons by means of rRMSE and forecasting skill *s*. The best model is marked with bold font.

Horizon [h]	1	2	3	4	5	6
Model	rRMSE [%]					
Global DNN	**25.07**	**30.18**	**32.36**	**34.19**	**36.10**	38.71
Linear	26.67	31.36	33.11	34.63	36.44	**38.35**
Local DNN	26.82	30.90	32.91	34.67	36.68	39.88
GBT	30.05	34.78	36.95	39.04	40.67	43.59
Persistence	28.74	36.89	42.29	47.28	52.05	56.69
ECMWF	35.91	35.01	35.12	35.91	37.45	39.28
	s [%]					
Global DNN	**9.98**	**18.38**	**23.40**	**27.04**	**28.30**	27.38
Linear	7.67	15.71	21.73	26.03	27.76	**28.42**
Local DNN	6.34	16.98	22.13	22.64	25.13	22.51
GBT	-5.18	6.06	12.23	15.29	16.00	15.11
Persistence	0	0	0	0	0	0
ECMWF	-29.07	4.68	16.77	22.74	23.23	20.19

As can be seen from Table 5.4 and Figure 5.3, the global model seems to be the best model for the first 5 prediction horizons (both in terms of rRMSE and forecasting skill *s*), and the second best (very close to the best one) for the last prediction horizon. Based on these results it can be observed that not only does perform the global model overall as well as or better than the local models, but it also performs equally well or better than them across all prediction horizons. In addition to this analysis of the global model performance, three interesting observations can be made:

1. The persistence model performs the worst across all prediction horizons except in the first one. This result agrees with previous results from the literature [65] that stated that the persistence model only provides reasonable results for prediction horizons shorter than 1 hour.

2. Among the local models, the linear and DNN models show the best performance across all 6 prediction horizons.

3. The ECMWF forecast improves its accuracy relatively to the other models as the prediction horizon increases. In the case of the last prediction horizon, the ECMWF forecast has almost the same performance as the global DNN and the linear models. Considering previous results from the literature [65], this is highly

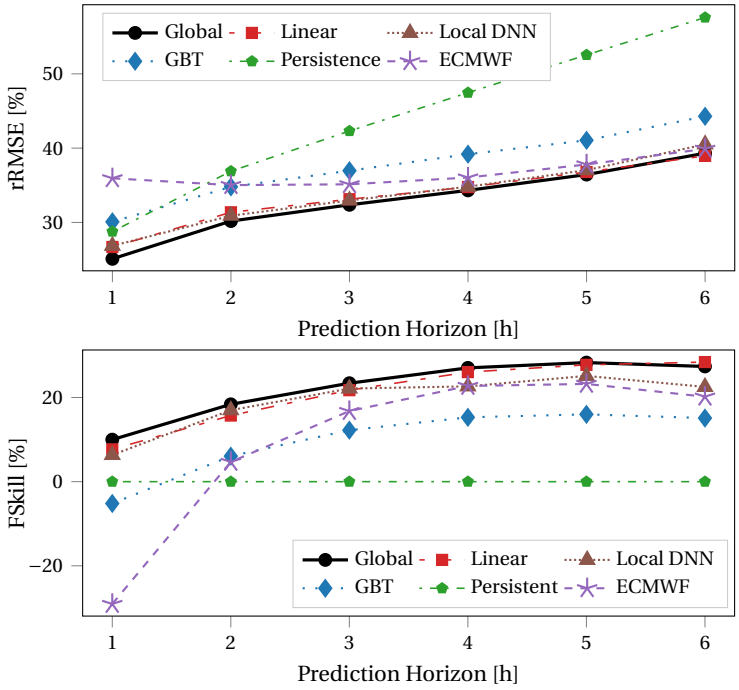


Figure 5.3: Comparison of the predictive accuracy of the various forecasters across the 6 prediction horizons. **Top:** Comparison by means of rRMSE. **Bottom:** Comparison by means of the forecasting skill *s*.

expected as NWP models start to perform better than time series models for prediction horizons larger than 4-6 hours.

4. For 1-hour ahead predictions, the ECMWF model performs the worst; specially, considering its *s* value for the first prediction horizon, the weather-based model performs much worse than a simple persistence model.

5.3.7. COMPARISON ACROSS GEOGRAPHICAL SITES

The final step to analyze the performance of the global model is to validate whether the quality of the performance is kept across the 25 different sites. In particular, it is important to check whether the global model can generalize and build accurate predictions across all geographical locations. For the sake of simplicity, this comparison is only done in terms of the rRMSE as, similarly to the previous results, the values of the forecasting skill *s* agree with the rRMSE across all locations and are a bit redundant. The comparison across the geographical locations is illustrated in Figure 5.4 and listed in Table 5.5.

As can be seen from Figure 5.4 and Table 5.5, the better performance of the global model seems to be maintained across all geographical locations. In particular, analyzing this results, it is clear that the global model performs equally well as or better than the local models across all 25 sites. As listed in Table 5.5, the global DNN is the best model

for 20 of the 25 locations, and shows an rRMSE performance that is very similar to the best model in the remaining 5 locations. Therefore, it can be concluded that the global model is a good replacement for the local models as the performance of the former is, at least, equal to the performance of the latter.

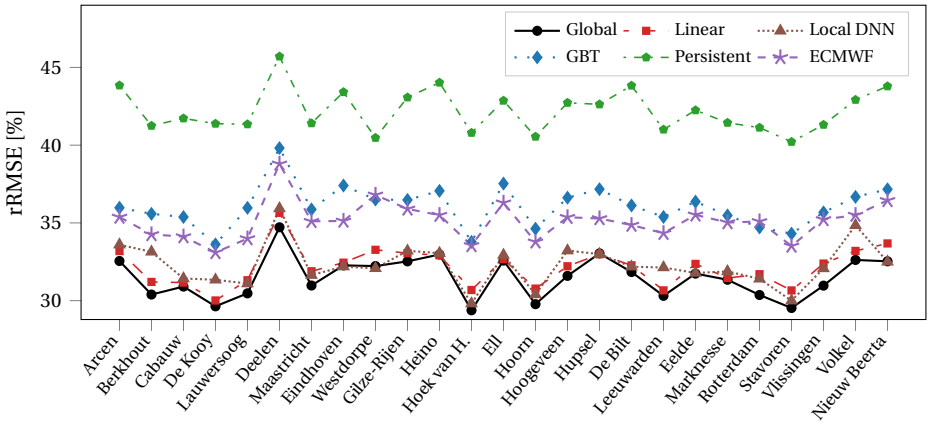


Figure 5.4: Comparison of the predictive accuracy of the various forecasters across the 25 locations.

5

GEOGRAPHICAL DEPENDENCIES

A final interesting study to analyze is whether the rRMSE has any geographical dependency, i.e. it might be possible that local geography or climate might have an effect on the rRMSE. To study this effect, a color map with the geographical distribution of the rRMSE can be used. Such a plot is represented in Figure 5.5, which depicts the geographical distribution of the rRMSE for the 6 different models. As can be observed, there is a clear difference between coastal and inland sites with the latter displaying rRMSEs that are consistently higher. While this difference is not large, it does seem to indicate that forecasting solar irradiance at inland locations is slightly harder than at coastal sites. While analyzing the cause behind this difference is out of the scope of this chapter, it is worth noting possible reasons that might cause it; particularly, differences in climate, altitude, or simple differences in irradiance ranges might explain this effect.

5.3.8. DISCUSSION

In the previous sections, the performance of the global model has been compared to that of the local models and that of validated models from the literature. Based on the obtained results one can conclude that: (i) the global model performs slightly better than the best of the local models; (ii) the global model performs similar to other models from the literature; (iii) the global model provides (almost) unbiased forecasts.

While based on these results it cannot be stated that the proposed model is significantly better than all other models, it is important to keep in mind that its main purpose is not to be the best, but to perform equally well as local models so that the operational

Table 5.5: Comparison of the predictive accuracy of the various forecasters by means of rRMSE. The best model is marked with bold font.

	Site								
Model	Arcen	Berkhout	Cabauw	De Kooy	Lauwersoog	Deelen	Maastricht	Eindhoven	Westdorpe
Global	**32.39**	**30.24**	**30.75**	**29.49**	**30.32**	**34.55**	**30.82**	32.11	32.07
Linear	33.03	31.05	31.01	29.87	31.16	35.47	31.73	32.28	33.11
DNN	33.43	32.77	31.27	31.14	30.95	35.75	31.48	**32.03**	**31.93**
GBT	35.80	35.41	35.20	33.45	35.79	39.62	35.68	37.22	36.33
Persistence	43.63	41.04	41.51	41.18	41.14	45.47	41.20	43.20	40.28
ECMWF	35.21	34.09	33.95	32.94	33.83	38.61	34.93	34.95	36.63

Model	Gilze-Rij.	Heino	Hoek van H.	Ell	Hoorn	Hoogeveen	Hupsel	De Bilt
Global	**32.37**	32.80	**29.24**	**32.42**	**29.63**	**31.44**	32.88	**31.68**
Linear	32.89	**32.75**	30.53	32.50	30.63	32.05	**32.82**	32.11
DNN	33.04	32.89	29.66	32.77	30.24	33.05	32.83	32.02
GBT	36.30	36.88	33.61	37.35	34.46	36.44	36.99	35.94
Persistence	42.86	43.80	40.59	42.65	40.35	42.51	42.42	43.61
ECMWF	35.73	35.32	33.39	36.12	33.62	35.19	35.11	34.69

Model	Leeuward.	Eelde	Marknesse	Rotterdam	Stavoren	Vlissingen	Volkel	N. Beerta
Global	**30.16**	**31.58**	**31.19**	**30.21**	**29.38**	**30.81**	**32.46**	32.37
Linear	30.51	32.2	31.3	31.54	30.51	32.23	33.04	33.52
DNN	31.97	31.62	31.72	31.25	29.85	31.92	34.68	**32.34**
GBT	35.2	36.19	35.3	34.53	34.14	35.5	36.5	36.98
Persistence	40.8	42.04	41.24	40.92	40.01	41.11	42.71	43.58
ECMWF	34.17	35.34	34.85	34.92	33.35	35.05	35.33	36.27

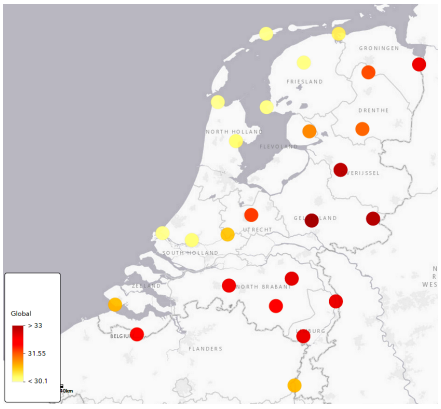
costs of installing and maintaining a wide sensor network are avoided. In that respect, it can be concluded that the proposed global model is an excellent replacement for the local models: the model performs overall slightly better than the local models and better or equally well across all individual geographical locations and prediction times.

5.4. CONCLUSIONS

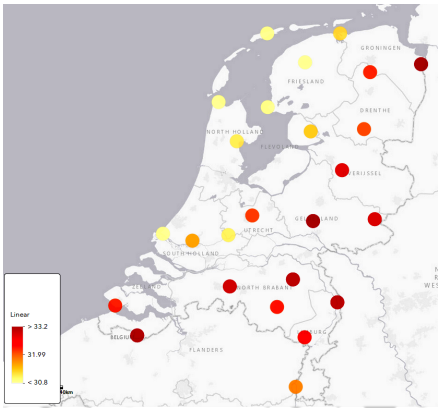
In this chapter, a general model for short-term forecasting of the global horizontal irradiance has been proposed. The main features of the model are that it replaces ground measurements by satellite-based irradiance values and that, unlike local models previously proposed in the literature, it does not need local measurements in each location where a forecast is needed.

The proposed model was shown to perform equal to or better than the local models used in the literature, and in turn, to be an excellent replacement of these local models to save the operational costs of installing local sensors and gathering ground data.

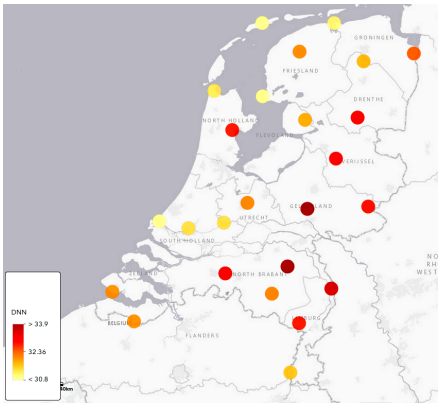
In future research, the current work will be expanded with two further investigations. First, the model will be extended to larger regions to analyze whether it generalizes to larger geographical areas than The Netherlands. Second, the model accuracy will be improved by adding other relevant sources of input data.



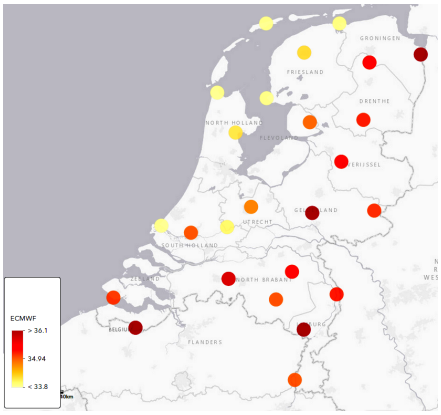
(a) Global



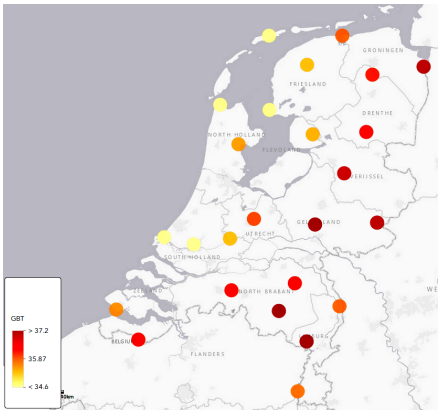
(b) Linear



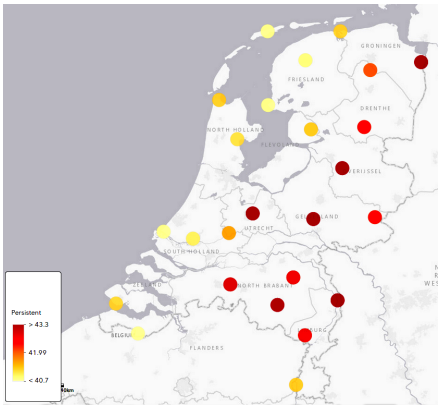
(c) Local DNN



(d) ECMWF



(e) GBT



(f) Persistent

Figure 5.5: Geographical distribution of the rRMSE based on the 25 out-of-sample sites. Light yellow and dark red respectively represent the lowest and highest rRMSE values. Across the 6 models, a clear difference between inland locations and coastal locations can be observed, with the latter having lower rRMSEs.

II

THERMAL SEASONAL STORAGE TO REDUCE GRID IMBALANCES

6

Background: Seasonal thermal energy storage systems

There are sadistic scientists who hurry to hunt down errors instead of establishing the truth.

Marie Skłodowska Curie

Some people use statistics as a drunken man uses lamp-posts — for support rather than for illumination.

Andrew Lang

This chapter provides background knowledge on the field of seasonal energy storage systems and on different algorithms and concepts used in the second part of the thesis. The chapter starts with a brief motivation on the research in seasonal energy storage together with a literature survey of the relevant research topics. Then, a mathematical model of the most widely used seasonal storage systems is presented. Next, a brief introduction into state-of-the-art control algorithms for seasonal storage is provided. Finally, the electricity markets that are relevant for this research are explained.

6.1. INTRODUCTION

As explained in the introductory chapter, while the energy transition [216] has the potential to highly improve our society, it also poses some problems [17]. Specifically, due to the weather dependence of renewable sources, a wide spread of renewables implies more uncertain energy generation. In the case of electricity, as generation and consumption have to be balanced at all times, the more renewable sources are integrated, the more imbalances between generation and consumption occur. This leads in turn to an electrical grid that is more complex to balance and control. To mitigate this issue, as proposed in Part I of the thesis, a potential solution is to employ accurate forecasting techniques to reduce the uncertainty (see Chapters 3–5).

A second option, which is explored in this second part of the thesis, is to reduce grid imbalances via seasonal storage systems interacting with electricity markets. Particularly, energy storage systems offer a promising solution by shifting generation and consumption of electricity, providing flexibility and ancillary services, and helping to obtain a smooth and reliable grid operation [194]. In this context, seasonal energy storage is arguably the most critical technology as the generation of renewables is season dependent and the share of renewables by 2030 is expected to reach very high levels [231]. With that motivation, in this part of the thesis we investigate modeling and control approaches for seasonal storage systems that can: (i) improve the state-of-the-art modeling and control solutions; (ii) reduce the imbalances on the electrical grid; and (iii) incentivize the use of seasonal storage systems.

Due to the importance of *seasonal thermal energy storage systems (STESSs)*, the focus of Part II is twofold: (i) to improve the modeling approaches for STESSs so that they can be efficiently integrated in optimization and control problems and (ii) to provide control approaches for STESSs that, through market trading, maximize the profits of STESSs while decreasing the grid imbalances. In detail, even though the perfect seasonal energy storage system does not exist, STESSs are one of the most efficient technologies due their matureness, cost-efficiency, and the large energy market for heat demands [194]. Particularly, considering that water and space heating accounts for nearly 80 % of the total energy consumption in European households [70], STESSs have a large energy market that they can use to shift the energy demand. In this context, to ensure the widespread adoption of STESSs and their use for grid balancing, it is paramount to develop control algorithms that not only reduce grid imbalances, but also maximize the profits of STESSs. The first step to achieve this goal is to develop a model for STESSs that can be efficiently integrated in optimization and control problems. Naturally, the second step is to develop trading and control strategies that maximize the profits of STESSs whilst reducing grid imbalances.

As introductory material, this chapter provides background knowledge on the field of seasonal storage and the other concepts explored in this second part of the thesis. In detail, the remainder of the chapter is organized as follows: Section 6.2 performs a brief literature review of the relevant research topics. Then, Section 6.3 describes the existing mathematical model for a commonly used STESS. Next, Section 6.4 introduces the existing state-of-the-art control approaches that are considered in the thesis. Finally, Section 6.5 provides a summary of the electricity markets that are relevant for trading with seasonal storage.

6.2. LITERATURE REVIEW

In this section, we present a brief literature review of the three topics that are relevant for Part II of the thesis: storage system technologies with a focus on seasonal (thermal) energy storage, modeling of seasonal stratified fluid tanks, and control of seasonal and non-seasonal energy storage systems.

6.2.1. ENERGY STORAGE SYSTEMS

Depending on the type of technology, there are different energy storage solutions [194, 231], e.g. lithium-ion batteries, pumped hydro storage, ultracapacitors, flywheels, molten-salt batteries, thermal storage systems, compressed air storage, or hydrogen storage. While most of these technologies can ensure efficient short-term and medium-term energy storage, efficient long-term energy storage has traditionally been more difficult to achieve. In particular, even though some of these technologies can store energy for long periods, they are economically not very efficient [194]. However, long-term energy storage is arguably one of the most important elements to ensure the success of the energy transition. Particularly, as the share of wind and solar energy by 2030 is expected to reach very high levels (70-80% in some countries), and as the generation of renewables is season dependent [231], seasonal energy storage solutions [231] that can store energy across several weeks or months are crucial in order to reduce seasonal fluctuations [194].

SEASONAL STORAGE

With regard to seasonal storage, there are primarily three solutions available that can provide electricity back to the grid: hydrogen storage, synthetic natural gas storage, and vanadium redox flow batteries [207, 231]. The first two approaches belong to the wider class of power-to-gas technologies that make use of renewable sources to generate synthetic fuels, i.e. primarily hydrogen and methane [180]. The third approach involves the next generation of batteries that can potentially store electricity for long horizon [157, 207]. In this context, besides vanadium redox flow batteries, there is also undergoing research into the next generation of post-lithium-ion technologies with capabilities of long-term storage [45, 238]. Despite their potential, in their current state these technologies still have several problems that make them economically non-viable. First, they are expensive technologies and in an early stage of development and testing [1, 45, 157, 180, 207, 238]. Second, synthetic fuels have a very low energy efficiency due to conversion losses [180]. Third, vanadium redox flow batteries and other post-lithium-ion batteries are yet not profitable and face multiple challenges that difficult their commercial deployment [19, 45, 207, 238].

Another option for storing energy over long horizons are *thermal energy storage (TES)* systems [247]. While in general these systems cannot provide electricity back to the grid, they are a more mature technology, have the advantage of being significantly less expensive than electrical energy storage [194], and can be used to satisfy heating and cooling demands. In particular, as the heating and cooling necessities correspond to 45% of the total domestic and commercial energy usage [166], TES can potentially be very helpful in the energy transition by performing demand side management.

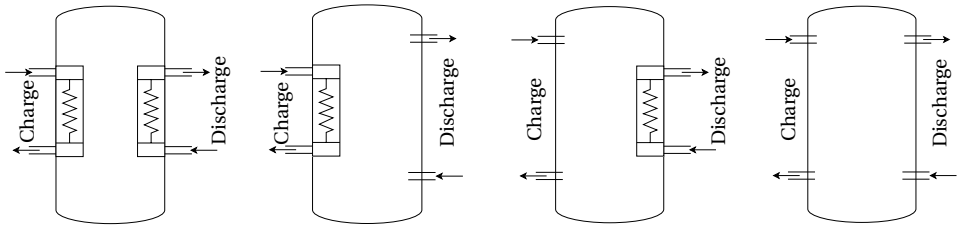
THERMAL SEASONAL STORAGE

In the context of TES technologies, there are three main categories: sensible heat storage, latent heat storage, and chemical energy storage [206, 247]. While the last two have higher energy densities, they are both more expensive and less mature, i.e. mostly at the laboratory testing stage and with no large-scale seasonal project completed [247]. By contrast, sensible energy storage is the simplest, cheapest, most widespread, and most mature technology [206]. In the framework of sensible heat storage, the use of water tanks for seasonal energy storage is arguably the cheapest and the most popular option [206]. In particular, multiple applications for these types of systems can be found in the literature [14, 136, 197]. Another advantage of these tanks is that they can use water stratification to improve the average net energy and exergy efficiency [206]. In addition, their energy balance can be easily described by ordinary or partial differential equations [136]. Henceforth, aligned with the literature [57, 200, 202], we will use the name of *seasonal thermal energy storage systems (STESSs)* to refer to TES systems based on sensible heat storage and with seasonal storage capabilities.

STRATIFIED FLUID TANK

One of the most important STESSs are stratified fluid tanks [99], which store energy by keeping fluid layers stratified at different temperatures. In detail, exploiting the fact that fluid density decreases as temperature increases, they are able to stratify fluid layers where the warmest layers are displaced to the top of the tank and the coldest layers to the bottom. This type of heat storage systems are widely used, and multiple applications can be found in the literature [14, 40, 96, 197, 218]. One of their main advantages is that, in comparison with a regular mixed fluid tank, a stratified tank improves the average net energy and exergy efficiencies by up to 60% [99]. To maximize the energy efficiency, the system is built such that the mixing between the stratified layers is minimized.

Within the family of stratified fluid tanks, there are several possible configurations depending on how the tank is charged, i.e. how heat is introduced in the tank, and on how the tank is discharged. In terms of charging, the fluid in the tank can be heated directly using a fluid flow or indirectly using a heat exchanger [99]. Likewise, when discharging the tank, heat can be directly extracted as a fluid flow or indirectly extracted using a heat exchanger. The four possible combinations are depicted in Figure 6.1.



(a) Indirect charge - Indirect discharge.

(b) Indirect charge - Direct discharge.

(c) Direct charge - Indirect discharge.

(d) Direct charge - Direct discharge.

Figure 6.1: Simplified configurations of stratified tanks when considering the two types of exchanging heat (direct and indirect) and the two directions of the heat exchange (charge and discharge).

The main advantage of indirect charge/discharge is that, as no flow is introduced in the tank, stratification is more easily maintained. However, since heat is indirectly transferred, the energy efficiency is lowered. In contrast, although direct heating introduces turbulences in the tank that might destroy the thermal stratification, it has a larger efficiency as heat is directly transferred [99]. In general, the best tank configuration will depend on several factors: the size of the tank, the stratification capabilities, or the flow of the input fluid.

6.2.2. MODELING OF STORAGE TANKS

The scientific literature regarding modeling of stratified thermal storage vessels is very large and diverse. Typically, the proposed models can be divided into three categories: 1-dimensional models [62, 128, 176, 193, 203], 2-dimensional models [58, 105], and 3-dimensional computational fluid dynamics models [186, 249]. Although 1D models are less accurate, they are the preferred choice in several applications: while 2D and 3D models are more accurate, their computational complexity makes them unsuitable for process optimization or long-term simulation of the storage tank [58, 99]. By contrast, to analyze the behavior of the fluid within the tank or the effect of new configurations, 2D and 3D models give detailed information that a 1D model cannot provide.

One of the most important reasons behind the reduced accuracy of 1D models is the approximation they make to model the mixing of layers due to buoyancy effects [58, 203]. In more detail, given two consecutive layers in a real tank, the lower layer might achieve a temperature higher than the top layer; in this scenario, the bottom layer would rise, the top layer would sink, and during this process both layers would mix. In a real tank, there are several scenarios when this effect might occur:

1. Due to its larger contact area with the environment, the top layer in the tank would normally suffer larger heat losses. Therefore, as the top layer loses more heat, there is a point when the layer below reaches a higher temperature and both layers mix.

2. When the tank is directly charged/discharged, new fluid enters the tank. If the temperature of the incoming fluid is higher than the temperature of any layer above the entrance point, the incoming fluid will rise and mix.

3. When the tank is indirectly charged/discharged, a lower layer might be heated more than a top layer. In this scenario, the fluid in the lower layer will rise and mix.

When considering 1D models for thermal stratified storage vessels from the literature, none of them can physically model this effect. More specifically, while 1D models consider heat transfers between fluid layers and input/output flows, they do not model the effect of gravity in the tank. To address this, the 1D models proposed in the literature usually include a post-processing step after each simulation step that approximates the mixing of layers due to buoyancy effects [58, 62]. This post-processing algorithm has the following structure:

1. Check the temperature of each layer and evaluate whether buoyancy effects are present.

2. If buoyancy effects are present, mix the corresponding layers.

3. Repeat steps 1-2 until buoyancy effects are removed.

6.2.3. CONTROL OF STORAGE SYSTEMS

In the framework of control for storage systems, we can distinguish between approaches for non-seasonal storage, which consider market interaction, and approaches for seasonal storage, which do not consider market interaction.

CONTROL OF NON-SEASONAL STORAGE SYSTEMS

The problem of controlling storage systems is a developed area of research that contains many approaches that consider market interaction. Nonetheless, within this context, research has mostly focused on short-term storage systems, i.e. non-seasonal storage. The aim of this section is to provide a brief overview of the different families of approaches within the field, describe which markets the control algorithms are designed for, and which control horizons are usually considered.

Optimization-based approaches have been employed in numerous applications [4, 13, 77, 85, 122, 126, 168, 172, 255] and are arguably the most widely used family. In order to interact with different markets, these approaches are formulated as sequential multi-stage optimization problems. Another family of approaches are based on dynamic programming [43, 94, 120]. While these approaches often provide global optimal solutions, do not scale for large systems as they suffer from the curse of dimensionality [22]. A third family are rule-based approaches [85, 174], which derive a set of logical rules to control the storage systems. Finally, there are game-theoretical models [260], which are based on competition economic models.

In terms of markets, control approaches have been proposed for many different cases. The most common of them is trading in the day-ahead market together with the balancing market [13, 122, 168, 255, 260] or with the real-time market [4, 77, 172]. Other proposed strategies include: frequency regulation coupled with energy arbitrage markets [43]; day-ahead market [94]; primary frequency response market [174]; real-time markets [126]; or day-ahead, intraday, and balancing markets [85, 120]. To the best of our knowledge, approaches that exploit the imbalance markets have not been proposed.

In terms of the horizon, the majority of the approaches perform price arbitrage between day-ahead and markets closer to real-time considering optimization horizons of one day [4, 13, 43, 77, 85, 120, 122, 126, 168, 172, 255]. In this context, no approaches provide solutions for trading energy over long horizons, e.g. months.

CONTROL OF SEASONAL STORAGE SYSTEMS

In the context of seasonal storage systems, several optimal control strategies have been proposed. However, none of the proposed methods are designed for market interaction. In [200] and [199], *model predictive control (MPC)* based strategies are proposed to control aquifer thermal energy storage systems; nonetheless, while the controller is designed to satisfy physical constraints and a stochastic heat demand, the STESS does not interact with electricity markets. Similarly, in [63], a dynamic programming approach is proposed to control borehole thermal storage systems; however, the controller assumes

a constant market price and does not distinguish between different markets. In [248], a model predictive control algorithm is proposed to control solar communities with a borehole thermal storage system; similar to other studies, price and markets are not considered and the controller is limited to satisfy the system constraints and the heat demand. In [57], a data-driven stochastic predictive control scheme to operate an energy hub with seasonal storage capabilities is proposed; the goal of the approach is to minimize the total energy consumption and to be cost efficient; however, here also, the algorithm does not consider real market prices nor market trading. Similarly, [242] proposes an optimal charging strategy for borehole thermal storage systems; nonetheless, the focus of the controller is to maximize the renewable energy use and to reduce CO_2 emissions, and also here, neither prices nor market interaction are considered.

6.3. MATHEMATICAL MODEL FOR STRATIFIED TANKS

THE standard 1D model [99, 128] for a stratified heat storage vessel divides the tank in n_L segments/layers. Then, it models each layer with a *partial differential equation (PDE)* based on the heat transfer equation. In its most general case, each layer i is characterized by a state T_i representing the temperature of the layer; this state can be controlled by the input flow $\dot{m}_i$ and its temperature T_i^{in} or by the external input heat $\dot{Q}_i$ (heat sink or hear source) in the layer. Figure 6.2 provides an example of this layout using a tank that is controlled via the external input heat $\dot{Q}_i$ and whose 1D model discretizes the tank in 8 layers.

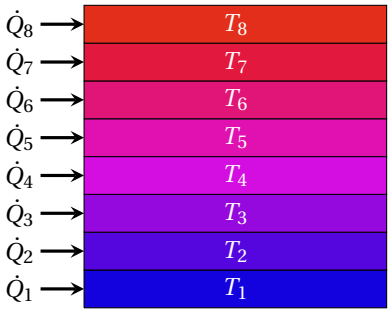


Figure 6.2: Example of the layout of the spatial discretization made by a 1D model. The tank is controlled by the external input heat $\dot{Q}_i$ and the 1D model uses 8 layers. The colors represent the difference in temperature due to stratification.

6.3.1. PARTIAL DIFFERENTIAL EQUATION

The PDE that models the state evolution of layer i is given by:

$$\frac{\partial T_i}{\partial t} = \alpha \frac{\partial^2 T_i}{\partial z^2} + \frac{P_i \kappa_i}{\rho c_p A_i}(T_\infty - T_i) + \frac{\dot{Q}_i}{\rho c_p A_i \Delta z_i} + \frac{\dot{m}_i (T_i^{\text{in}} - T_i)}{\rho A_i \Delta z_i}, \tag{6.1}$$

where α, ρ, and c_p respectively represent the fluid diffusivity, density, and specific heat; A_i, P_i, and Δz_i the cross-sectional area, perimeter, and thickness of layer i; κ_i the

thermal conductance of the isolation wall of layer i; and T_∞ the ambient temperature (ground temperature if the vessel is underground).

It is important to note that not all vessel architectures make use of all input controls: in the case of direct charging and discharging, only the input controls $\dot{m}_i$ and T_i^{in} are used; similarly, in the case of indirect charging and discharging, only $\dot{Q}_i$ is required. In addition, it is also important to remark that T_i^{in} might be T_{i-1} or T_{i+1} depending on whether the flow $\dot{m}_i$ comes from the bottom or top layer.

6.3.2. MIXING AND INVERSION OF LAYERS

This 1D model has an important drawback: as it is solely based on heat transfer, it cannot model the mixing of layers due to buoyancy effects. To address this issue, the traditional models from the literature perform, after each simulation time step, a *non-smooth* post-processing algorithm. In this post-processing step, the temperature of all layers is checked to detect buoyancy effects; if buoyancy is present, the layers involved are mixed; this process is repeated until all buoyancy effects are removed.

A simple example of this traditional simulation scheme is given in Algorithm 2. In this example, the vessel is divided into n_{L} layers, where layer n_{L} is at the top and layer 1 at the bottom of the tank, and the tank is steered using a generic control vector $\mathbf{u}$, where $\mathbf{u}$ can comprise the input flow $\dot{\mathbf{m}} = [\dot{m}_1, \dots, \dot{m}_{n_{\text{L}}}]^\top$, the input temperature $\mathbf{T}^{\text{in}} = [T_1^{\text{in}}, \dots, T_{n_{\text{L}}}^{\text{in}}]^\top$ of the flow, and/or the heat sink/source $\dot{\mathbf{Q}} = [\dot{Q}_1, \dots, \dot{Q}_{n_{\text{L}}}]^\top$. The algorithm simulates the system using a simulation time step of length Δt and for a total of N time steps. As can be seen in lines 3 and 4, the simulation routine involves two steps: a first part where the PDE is solved and a second part where the buoyancy effects are included as an iterative algorithm. For this second step, each layers i is mixed according to its volume $V_i = A_i \, \Delta z_i$.

6.3.3. MODEL DRAWBACK

While this traditional scheme is a very good approximation when simulating the dynamics of the heat storage vessel, it is not so suitable for using it in derivative-based optimization problems. In particular, the traditional scheme has the disadvantage that the dynamics of the tank are not defined by a single continuous equation, and as such, they cannot be used with derivative-based optimization algorithms, e.g. Newton-based methods, with analytical-based derivative computations.

This limitation is of crucial importance when controlling the vessel and estimating its parameters. In detail, if a specific temperature profile is required, the corresponding controls to steer the tank need to be computed. If the original 1D model is used, either heuristic methods or derivative-based optimization approaches with finite differences are needed. Since heuristic methods have larger computational requirements than derivative-based optimization methods, they might easily become unusable as their computation time can grow larger than the controller time step. Similarly, as the computation burden of finite differences scales badly with the number of optimization variables, derivative-based optimization approaches suffer from the same problem. In addition to the computation time, heuristic methods cannot guarantee that the obtained solution is a local minimum and thus they have no guarantees on the quality of the solution. Likewise, finite differences lose 75% of the numerical precision when computing

Algorithm 2 Traditional Simulation Scheme

function SIMULATOR($\mathbf{T}_1$, $[\mathbf{u}_1, \ldots, \mathbf{u}_N]$, Δt, $\mathbf{V}$)
  for $k \in \{1, 2, \ldots, N\}$ **do**
    $\mathbf{T}_{k+1} \leftarrow$ `simulatePDEStep`($\mathbf{T}_k, \mathbf{u}_k, \Delta t$)
    $\mathbf{T}_{k+1} \leftarrow$ `correctBuoyancy`($\mathbf{T}_{k+1}, \mathbf{V}$)
  end for
  return $\mathbf{T}_2, \ldots, \mathbf{T}_{N+1}$
end function
function CORRECTBUOYANCY($\mathbf{T}, \mathbf{V}$)
  while $\min\{T_i - T_{i-1} \mid i = 2, \ldots, n_\mathrm{L}\} < 0$ **do**
    for $i \in \{2, \ldots, n_\mathrm{L}\}$ **do**
      if $T_i < T_{i-1}$ **then**
        $T_i, T_{i-1} \leftarrow$ `mixLayers`($T_i, T_{i-1}, V_i, V_{i-1}$)
      end if
    end for
  end while
  return T
end function
function MIXLAYERS(T_top, T_bottom, V_top, V_bottom)
  $\Delta T = T_\mathrm{bottom} - T_\mathrm{top}$
  $T_\mathrm{bottom} = T_\mathrm{bottom} - \frac{V_\mathrm{top}}{V_\mathrm{top}+V_\mathrm{bottom}} \Delta T$
  $T_\mathrm{top} = T_\mathrm{top} + \frac{V_\mathrm{bottom}}{V_\mathrm{top}+V_\mathrm{bottom}} \Delta T$
  return T_top, T_bottom
end function

second-order derivatives and might also compromise the quality of the solution. Following the same argument, using the standard 1D model might not be a feasible choice when estimating the parameters. As with control applications, the quality of the obtained parameters might be lower and/or the required computation time might be too high.

6.4. STATE-OF-THE-ART CONTROL APPROACHES

WHILE several control approaches could potentially be used for seasonal storage systems, in the thesis we focus on the two most important state-of-the-art families: predictive control via MPC and artificial intelligence via *reinforcement learning (RL)*.

6.4.1. INTRODUCTION TO MPC

The general idea of MPC is to, at each discrete time step k, obtain the optimal control $\mathbf{u}_k^\star$ by using the following iterative structure:

1. Read current state $\mathbf{x}_k$.
2. Based on $\mathbf{x}_k$, solve the relevant *optimal control problem (OCP)* over a horizon of N time intervals.

3. Based on the solution of this optimization problem, obtain the optimal control $\mathbf{u}_k^\star$.
4. Apply this control to the system.
5. Repeat the process again for the next time step $k+1$.

For more details on MPC we refer to [195].

6.4.2. INTRODUCTION TO RL

The general idea of RL is to, at each time step k, obtain the optimal control $\mathbf{u}_k^\star$ by using an optimal policy $\pi^\star(\mathbf{s}_k)$, i.e. a function that outputs the optimal action $\mathbf{u}_k^\star$ for each state $\mathbf{s}_k$. To learn the policy $\pi^\star(\mathbf{s}_k)$, the RL algorithm assumes that the dynamical system and its environment can be modeled via a Markov decision process [73, 223]. In particular, in its most basic form, the RL algorithm assumes that:

- The system lives in a discrete-time world.
- The system is controlled by an agent that takes actions $\mathbf{u}$ among a discrete set of actions $\mathcal{U} = \{\mathbf{u}^1, \dots \mathbf{u}^{n_\mathrm{u}}\}$.
- The system and the environment are modeled by the agent state $\mathbf{s}$ where, in general, the state $\mathbf{x}$ of the system is part of the state $\mathbf{s}$ of the RL agent.
- At every discrete time step k, the agent takes an action $\mathbf{u}_k$ and transitions from state $\mathbf{s}_k$ to $\mathbf{s}_{k+1}$ based on some probabilistic dynamics $p(\mathbf{s}_{k+1}|\mathbf{s}_k, \mathbf{u}_k)$.
- In the transition, the agent receives a reward r_k based on a distribution $q(r_k|\mathbf{s}_k, \mathbf{u}_k)$.

POLICY ESTIMATION

During training, the RL agent iteratively performs an exploration step and an exploitation step:

- **Exploration**: The agent controls the system to interact with the environment for a number of n_steps steps. Then, it gathers the data resulting from that interaction in a memory dataset $\mathcal{M} = \{(\mathbf{s}_k, \mathbf{u}_k, \mathbf{s}_{k+1}, r_k)\}_{k=1}^{n_\mathrm{steps}}$. To select $\mathbf{u}_k$ during the exploration, the agent uses both optimal actions from $\pi^\star(\mathbf{s}_k)$ and random actions. In future repetitions of this exploration step, new data are added to the memory $\mathcal{M}$.
- **Exploitation**: The agent uses $\mathcal{M}$ to improve the optimal policy $\pi^\star(\mathbf{s}_k)$. In particular, $\pi^\star(\mathbf{s}_k)$ is estimated so that the expected value of the cumulative sum of discounted rewards R is maximized:

$$R = \sum_{k=1}^{T_\mathrm{e}} \gamma^{T_\mathrm{e}-k} \mathop{\mathbb{E}}_{q(r_k|\mathbf{s}_k,\mathbf{u}_k)} \{r_k\}, \tag{6.2}$$

  where T_e is the length of a RL episode, i.e. for how long the RL agent takes decisions, and γ is a discount factor that prioritizes earlier rewards and allows R to be finite even for episodes with an infinite horizon.

In general, what defines and separates the large family of RL algorithms is the manner in which these two steps are performed, i.e. the number of steps n_{steps}, the size of the memory $\mathcal{M}$, or the algorithm to estimate $\pi^\star(\mathbf{s}_k)$.

Q-LEARNING

One of the most widely used algorithms to estimate $\pi^\star(\mathbf{s}_k)$ is fitted Q-iteration [73]. In this RL scheme, the algorithm is trained by iteratively performing two steps.

In the first step, the RL agent interacts with the environment during a full episode and stores the dataset $\{(\mathbf{s}_k, \mathbf{u}_k, \mathbf{s}_{k+1}, r_k)\}_{k=1}^{T_e}$ of transitions, actions, and rewards that it experiences. To take the action $\mathbf{u}_k$, it uses an ϵ-greedy approach: it takes a random action with a probability ϵ and the optimal action (given by last estimated optimal policy $\hat{\pi}^\star(\mathbf{s}_k)$) with a probability $1-\epsilon$, where ϵ decays after each episode. To make the agent more robust, the initial state $\mathbf{s}_0$ of the episode is often randomly selected.

In the second step, the algorithm uses the dataset of stored data to estimate an optimal policy $\hat{\pi}^\star(\mathbf{s}_k)$. In particular, it estimates a model $\hat{Q}(\mathbf{s}_k, \mathbf{u}_k)$ of the so-called Q-functions $Q(\mathbf{s}_k, \mathbf{u}_k)$, which represent the expected cumulative reward at state $\mathbf{s}_k$ when taking action $\mathbf{u}_k$ and then acting optimally. Then, an estimation of the optimal policy is built as $\mathbf{u}_k^\star = \hat{\pi}^\star(\mathbf{s}_k) = \underset{\mathbf{u}_k}{\operatorname{argmin}}\, \hat{Q}(\mathbf{s}_k, \mathbf{u}_k)$

These two steps run one after the other until the algorithm converges. We refer to [73] and [223] for further details.

6

6.5. ELECTRICITY MARKETS FOR STESS

ALTHOUGH STESSs can theoretically trade in any market, there are two markets that are particularly important: the day-ahead market and the imbalance market. Trading only in the day-ahead market is the safest strategy as the day-ahead market has the largest volume of renewables and players incur no risks as they submit bidding curves. However, trading only in the day-ahead market might not be the most optimal economic strategy as, while on average prices in the imbalance market are larger than in the day-ahead market, there are periods of time where imbalance prices are much lower. These imbalance prices, albeit lower, have an associated risk: in the imbalance market agents take an action without knowing the imbalance price and the existing price forecasters are not very accurate. Hence, unlike in the day-ahead market, the final cost of the purchased energy is uncertain.

6.5.1. TRADING IN THE DAY-AHEAD MARKET

The day-ahead electricity market [219] is a type of power exchange widely used around the world. In this market, consumers/producers submit bids for day d before some deadline on day $d-1$, where a bid indicates how much they are willing to pay/ask for different power volumes. With some exceptions, these bids are usually hourly based, i.e. each market player submits 24 bids. After the deadline, the market operator takes into account all the bids, computes the market clearing price for each of the 24 hours, and consumer/producer bids higher/lower than or equal to the market clearing prices are approved. For a more detail description of the day-ahead market we refer to Section 2.4.2.

6.5.2. TRADING IN THE IMBALANCE MARKET

Electricity, unlike most commodities, cannot be stored in very large amounts and requires almost equal consumption and generation at all times [72, 235]. However, as electricity consumption and generation are uncertain and weather dependent, in practice there are always imbalances between generation and consumption created by market agents that do not consume or generate what they had promised in the markets [178]. These imbalances have an adverse impact on the electrical grid frequency [178], and can lead to grid problems and instabilities and in some cases even blackouts [72]. To correct these imbalances, the *transmission system operator (TSO)* manages a so-called reserve market [188, 235]. On this market, specific market agents sell their available energy reserves to the TSO, i.e. their capacity to reduce and increase their generation and consumption, and the TSO purchases some of these reserves days or weeks in advance. Then, the TSO activates the required reserves in real time to correct grid imbalances.

Based on the price the TSO had to pay to correct the imbalances, it invoices the market agents that have caused the imbalances [234]. This mechanism where all the market agents pay for their imbalances is known as imbalance market or imbalance settlement [234]. Usually, this market is cleared every 15 minutes, i.e. each market agent pays for their cumulative positive or negative imbalance in intervals of 15 minutes.

In some countries, it is discouraged or even forbidden to use such a market for electricity trade, i.e. market agents are expected to trade honestly in the markets available before delivery time and only produce unexpected imbalances. However, as during periods of positive imbalances, i.e. when generation is larger than consumption, prices are low, and during period of negative imbalances prices are high, the economic incentive of market agents in this imbalance market is aligned with the regulatory duties of the TSO. Based on that, some other countries, e.g. The Netherlands [113], allow and encourage participation in this market.

6.6. CONCLUDING REMARKS

In this chapter, we have provided background knowledge on seasonal energy storage systems and on other concepts used in the second part of the thesis. We have started the chapter by motivating the research on seasonal energy storage. Then, we have provided a short literature survey of energy storage systems, modeling of thermal storage tanks, and control of storage systems. Next, we have briefly described state-of-the-art control algorithms for seasonal storage systems. Finally, we have explained the structure and working principles of electricity markets.

7

A 1-D CONTINUOUS AND SMOOTH MODEL FOR THERMALLY STRATIFIED SEASONAL STORAGE

The threat to the planet is us. It's actually not a threat to the planet — it's a threat to us.

Margaret Atwood

All models are wrong, but some are useful.

George Box

Efficient seasonal thermal energy storage is critical to ensure the energy transition. In this chapter, we propose a new model for one of the most efficient types of seasonal heat storage systems: stratified thermal seasonal storage systems. The novelty of the model is twofold. First, unlike the non-smooth models from the literature, it identifies the mixing and buoyancy dynamics using a smooth and continuous function. This smoothness property is critical to efficiently integrate thermal storage vessels in optimization and control problems. Second, it considers two types of buoyancy, a distinction that is paramount to identify accurate models. To show the relevance of the model, we consider a real system and validate the proposed model by showing that the estimated parameters correctly identify the physical properties of the vessel. Then, we employ the model in a control problem and we compare the model performance against that of a non-smooth model from literature. We show that: (i) the smooth model obtains the best optimal solutions; (ii) its computation costs are 100 times cheaper; (iii) it is the best alternative for use in real-time model-based control strategies, e.g. model predictive control.

Parts of this chapter have been published in [136].

7.1. INTRODUCTION

SINCE the generation of renewables is seasonal-dependent, seasonal energy storage has become one of the most important storage technologies in the context of the energy transition [231]. While several options exist for long-term or seasonal storage of energy, *seasonal thermal energy storage systems (STESSs)* are one of the most efficient technologies. Within the family of STESSs, one of the most important systems are stratified fluid tanks [99], which store energy by keeping fluid layers stratified at different temperatures.

Despite their importance, existing models for stratified tanks are very limited and cannot be efficiently used in optimization and control applications. Particularly, as explained in Section 6.2.2, to model stratified thermal storage vessels, there exist 1D, 2D, and 3D models. The last two, despite being highly accurate, cannot be used in optimization and control applications because of their large computational costs. 1D models, albeit more suitable for optimization problems, are inaccurate because of their limitations when modeling buoyancy [58, 203]. Particularly, none of the existing 1D models can model buoyancy using a continuous equation but require a post-processing routine after each simulation step [58, 62].

While this post-processing step can be efficiently used in simulations, it has a large limitation when used in optimization problems as the dynamics of the tank are not defined by a single continuous equation but by a non-smooth algorithm. Hence, the existing 1D models cannot be used with derivative-based optimization algorithms that make use of analytical expressions. This in turn poses computational issues for control applications since the optimization problems can only be solved via heuristic optimization or via derivative-based optimization with finite differences. In both cases, solving the optimization problems requires large computation times and the controller time step is often not large enough for these methods to find a solution. Besides the computational issues, using the existing 1D models together with heuristic methods can potentially lead to bad solutions as the latter cannot guarantee that the obtained solution is a local minimum.

CONTRIBUTIONS AND ORGANIZATION OF THE CHAPTER

To solve the described problem, in this chapter we propose a 1D continuous and smooth dynamical model that can accurately model the buoyancy effects. Due to its properties, the proposed model can be used in derivative-based optimization algorithms that use analytical methods to compute derivative information, e.g. automatic differentiation. Although the model is proposed and tested in the context of seasonal stratified fluid tanks, the model is also suitable for non-seasonal stratified tanks. In this context, it is important to note that a model with similar characteristics has been proposed in [14]; however, that model included non-smooth expressions and only modeled the buoyancy effects due to direct charging and discharging. By contrast, the proposed model is smooth and can represent the buoyancy effects due to heat losses, indirect charging and discharging, and direct charging and discharging.

As a second contribution, we take an approach that differs from the existing literature by explicitly modeling and identifying two different types of buoyancy: slow buoyancy effects that are linked to naturally occurring processes, and fast buoyancy effects that

are associated with charging and discharging the vessel. As we show, this distinction is very important to obtain a smooth model that can accurately represent the buoyancy dynamics.

Finally, as a third contribution, to show the benefits of using the smooth model in optimization problems, we compare the smooth model against the non-smooth model in a real-life optimization setup: an optimal control application where a 1500 m^3 commercial seasonal storage vessel needs to satisfy a given heat demand over some time horizon and, knowing the electricity prices over the given period, it has to minimize the cost of charging the vessel while satisfying the demand. Using this case study we show that the smooth model does not only obtain the best optimal solutions, but its computation costs are 100 times cheaper.

The remainder of the chapter is organized as follows: Section 7.2 introduces the real thermal vessel used as a case study to validate and study the proposed model. Section 7.3 presents and discusses the proposed model. Next, Section 7.4 estimates the parameters of the model when applied to the real system and validates the model. Then, Section 7.5 illustrates the benefits of the model in an optimal control set-up by comparing the model performance against non-smooth models from the literature. Finally, Section 7.6 concludes the chapter and discusses future research.

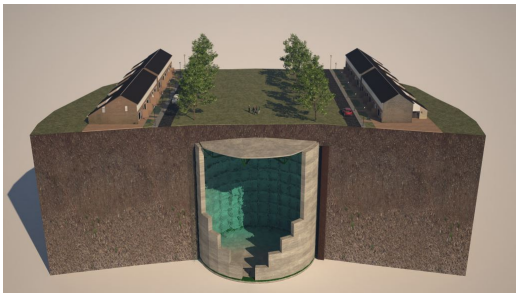
7.2. REAL SEASONAL VESSEL

IN this chapter, to illustrate the proposed model, we consider a real stratified STESS: the Ecovat vessel [69]. This system will be used as a case study to validate the model and it is briefly presented here as a short introduction to this type of storage systems.

The considered thermal vessel is a large subterranean thermal storage vessel with capabilities for seasonal thermal storage and with the ability to supply heat demand to a cluster of buildings. The thermal vessel is divided into several segments that can be charged and discharged separately. Due to this property and thermal stratification, each of these segments acts as a different heat buffer. The vessel employs indirect charging/discharging via heat exchangers located in the vessel walls that can receive heat from two difference sources: a heat pump or resistance heaters. The insulation structure of the vessel is such that it can very efficiently store energy between seasons. In particular, the heat losses of the vessel are about 25% in a period of 6 months. Another advantage of the vessel is that, due to its indirect charging/discharging structure, the stratification of the layers, i.e. the exergy of the system, can be better maintained. Figure 7.1 provides a schematic overview of the vessel and Figure 7.2 shows the real system when it was under construction.

7.2.1. TECHNICAL SPECIFICATIONS

Depending on the specific use case, the commercial versions of the Ecovat vessel can be built with different volumes and different numbers of heat buffers, with volumes and numbers of heat buffers respectively ranging from 20000 m^3 to 100000 m^3 and from 8 to 15. For our case study, as the first commercial vessel is still under construction, we consider the Ecovat prototype, a thermal storage system of 1500 m^3 and 5 heat buffers that has been operational for approximately one year.



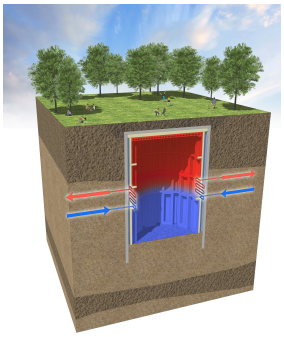


Figure 7.1: Schematic representation of the Ecovat system.

Figure 7.2: Construction of the real Ecovat system.

A technical schematic diagram of the prototype is depicted in Figure 7.3, where the gray parts represent concrete elements, the blue ones water, and the yellow ones the insulation material. In the schematic diagram, all dimensions are expressed in millimeters, the reference point of the system is located approximately 1 m above the ground, and the water level starts 4 m below the reference point (i.e. approximately 3 m below the ground level). The diameter of the vessel is approximately 11 m, and the water depth is 15.3 m (i.e. 19.3 m deep from the reference point). Moreover, the considered vessel has 5 heat buffers, each one of them with different isolation thickness.

The vessel consists of an external concrete wall, an intermediate isolation layer, and an internal concrete wall. Although it cannot be seen from the figure, the heat exchangers are not inside the water but embedded in the internal concrete walls. It is important to note that the thermal properties of the building materials, i.e. concrete and foam glass, are only known within a range of values. These parameters are listed in Table 7.1, which summarizes the thermal parameters of the concrete, the foam glass, and the fluid (water) used in the Ecovat prototype.

7.2.2. WORKING PRINCIPLE

As can be seen from Figure 7.3, the seasonal vessel consists of 5 different heat buffers. These heat buffers are separated along the wall by a horizontal layer of isolation material so that the heat conductance across the walls is restricted to each buffer. This struc-

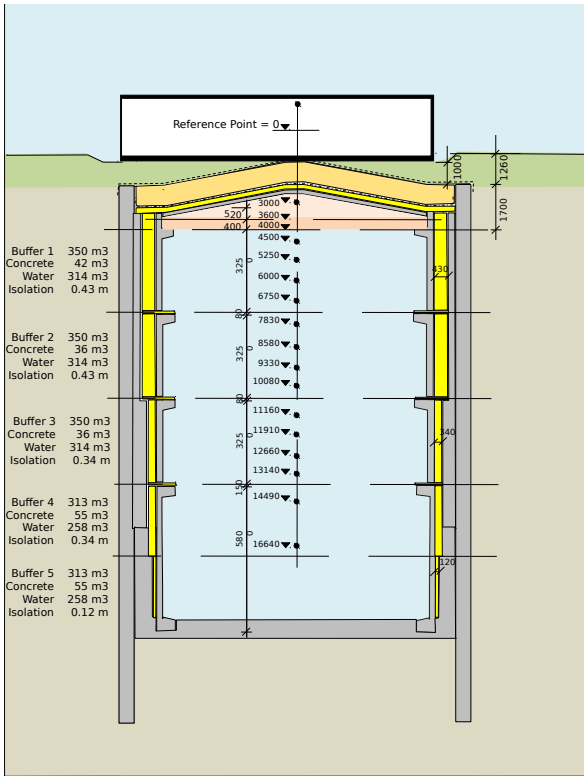


Figure 7.3: Technical schematic diagram of the Ecovat. All dimensions are given in mm (unless specified otherwise).

ture prevents the destruction of stratification due to wall conductance between the heat buffers and helps the vessel to store energy more efficiently.

In addition to being isolated, each of the heat buffers can be charged and discharged independently. Particularly, as mentioned in the previous section, each heat buffer has heat exchangers embedded in the internal walls, which allows charging and discharging the heat buffer independently. This specific location of the heat exchangers will play a very important role when estimating the system parameters: as explained in Section 7.4.2, this characteristic implies that the medium where the heat is stored is a mixture of fluid and concrete.

7.3. MATHEMATICAL MODEL

In this section, we derive and explain the proposed model for stratified tanks. To do so, we consider the existing models and their limitations, and we derive the new model based on them.

Table 7.1: Thermal properties of the materials. Some of them are only known within a range of values.

Parameter	Value/Range	Units
Density of concrete	2360	kg/m^3
Thermal conductivity of concrete	1.8	W/m K
Specific heat of concrete	[750, 1170]	J/kg K
Thermal diffusivity of concrete	$[10^{-6}, 6.5{\cdot}10^{-7}]$	m^2/s
Thermal conductivity of foam glass	0.041	W/m K
Density of water	1000	kg/m^3
Specific heat of water	4181.3	J/kg K
Thermal diffusivity of water	$0.143 \cdot 10^{-6}$	m^2/s

7.3.1. TRADITIONAL MODEL

As explained in Section 6.3, a standard 1D model divides the tank in n_L layers, characterizes each layer i by its temperature T_i, controls T_i via the input heat $\dot{Q}_i$ or the input flow $\dot{m}_i$ with temperature T_i^{in}, and models each layer with a *partial differential equation (PDE)* based on heat transfer:

$$\frac{\partial T_i}{\partial t} = \alpha \frac{\partial^2 T_i}{\partial z^2} + \frac{P_i \kappa_i}{\rho c_p A_i}(T_\infty - T_i) + \frac{\dot{Q}_i}{\rho c_p A_i \Delta z_i} + \frac{\dot{m}_i (T_i^{in} - T_i)}{\rho A_i \Delta z_i}. \tag{7.1}$$

In this equation, α, ρ, and c_p respectively represent the fluid diffusivity, density, and specific heat; A_i, P_i, and Δz_i the cross-sectional area, perimeter, and thickness of layer i; κ_i the thermal conductivity of the isolation wall; and T_∞ the ambient/ground temperature.

MODEL DRAWBACK

This 1D model has an important drawback: it cannot model the mixing of layers due to buoyancy. As described in Section 6.3, the existing models address this issue by performing a non-smooth post-processing algorithm after each simulation time step. In this post-processing step, the temperature of all layers is checked to detect buoyancy, if buoyancy is detected the layers involved are mixed, and the process is repeated until all buoyancy effects are removed. A simple example of this traditional simulation scheme has been given in Algorithm 2.

Although this traditional scheme is a very good approximation when simulating the dynamics, it is not so suitable for use in derivative-based optimization problems. In particular, this limitation becomes very important when controlling the vessel and/or estimating its parameters: to solve the related optimization problems either heuristic methods or derivative-based optimization methods with finite differences are needed. As explained before, these methods have two problems:

1. Their computational requirements can easily become infeasible, e.g. their computation time can grow larger than the controller time step.

2. They can compromise the quality of the solution: heuristic methods cannot guarantee that the obtained solution is a local minimum and finite differences lose up

to 75% of the numerical precision when computing derivatives.

7.3.2. MODELING SLOW BUOYANCY EFFECTS

In order to tackle these issues, derivative-based optimization methods with analytical derivative computations, i.e. automatic differentiation [93], can be used. However, in order to use them, the dynamics of the system need to be modeled by smooth expressions. In this chapter, we propose a possible solution to include the buoyancy effects within the dynamics of the system using a continuous and smooth function. In particular, in this first section, we propose a general methodology for buoyancy effects whose time span is larger than the simulation time step, i.e. slow buoyancy effects. Next, in Section 7.3.3, we expand the model to also include fast buoyancy effects, i.e. buoyancy effects with a time span shorter than the simulation time step.

DISCRETIZED DYNAMICS

In a simulation framework, the PDE defined by (7.1) is normally discretized and integrated over time using an expression of the following form:

$$T_{k+1,i} = F_i(\mathbf{T}_k, \dot{Q}_{k,i}, \dot{m}_{k,i}, T^{\text{in}}_{k,i}, \Delta t), \tag{7.2}$$

where $T_{k,i}$ is the temperature of layer i at time step k, $\mathbf{T}_k$ the vector of temperatures in the n_L layers at time step k, $\dot{Q}_{k,i}$, $\dot{m}_{k,i}$, and $T^{\text{in}}_{k,i}$ the control inputs of layer i at time step k, and Δt the length of the time step. As an example, we can derive F_i in two steps using a simple integration method:

1. Transforming the PDE into an *ordinary differential equation (ODE)* by approximating the second-order spatial derivatives by finite differences.
2. Using a numerical integration method to perform integration of the ODE.

In this case, if a forward Euler method is used for the numerical integration, (7.2) is equivalent to:

$$\begin{aligned} T_{k+1,i} = T_{k,i} + \Bigg(& \alpha \frac{T_{k,i+1} + T_{k,i-1} - 2\,T_{k,i}}{{\Delta z_i}^2} + \beta_i \,(T_\infty - T_{k,i}) \\ & + \frac{\lambda_i}{\Delta z_i}\,\dot{Q}_{k,i} + \frac{\phi}{\Delta z_i}\,\dot{m}_{k,i}\,(T^{\text{in}}_{k,i} - T_{k,i}) \Bigg) \Delta t \end{aligned} \tag{7.3}$$

where Δz_i is the thickness of layer i and:

$$\beta_i = \frac{P_i \kappa}{\rho c_\text{p} A_i}, \quad \lambda_i = \frac{1}{\rho c_\text{p} A_i}, \quad \phi = \frac{1}{\rho A_i}. \tag{7.4}$$

BUOYANCY EFFECTS VIA THE MAX FUNCTION

A single non-smooth mixing iteration in the standard approach (lines 11–13 in Algorithm 2), i.e.:

$$\begin{aligned} & \textbf{If } T_{k,i} < T_{k,i-1}: \\ & \quad T_{k,i}, T_{k,i-1} \leftarrow \texttt{mixLayers}(T_{k,i}, T_{k,i-1}, V_i, V_{i-1}) \\ & \textbf{End if} \end{aligned} \tag{7.5}$$

can be easily modeled using the max function [62] via the following two expressions:

$$T_{k,i} = T_{k,i} + \theta_{i,i-1} \max\big(0, T_{k,i-1} - T_{k,i}\big), \tag{7.6}$$

$$T_{k,i-1} = T_{k,i-1} - (1-\theta_{i,i-1}) \max\big(0, T_{k,i-1} - T_{k,i}\big), \tag{7.7}$$

where

$$\theta_{i,i-1} = \frac{V_{i-1}}{V_i + V_{i-1}} = \frac{A_{i-1}\,\Delta z_{i-1}}{A_i\,\Delta z_i + A_{i-1}\,\Delta z_{i-1}} \in [0,1] \tag{7.8}$$

is the ratio between the volume of layer $i-1$ and the sum of the volumes of both layers.

It can be argued that if the dynamics of the buoyancy effects are slow, a single mixing iteration might suffice. In particular, if the simulation time step Δt is small in comparison with the time span where the effects of buoyancy start to be noticed, a single mixing iteration can keep up with the changes that occur due to buoyancy. In this scenario, using (7.6) and (7.7), the mixing algorithm can be directly integrated in the discrete dynamics. More specifically, (7.2) can be expanded as follows:

$$\begin{aligned} T_{k+1,i} =& F_i(\mathbf{T}_k, \dot{Q}_{k,i}, \dot{m}_{k,i}, T_i^{\text{in}}, \Delta t) \\ &+ \theta_{i,i-1} \max(0, T_{k,i-1} - T_{k,i}) \\ &- \theta_{i,i+1} \max(0, T_{k,i} - T_{k,i+1}) \end{aligned} \tag{7.9}$$

BUOYANCY EFFECTS VIA A SMOOTH FUNCTION

While the max approximation allows to integrate the buoyancy effects directly within the dynamics, the resulting equations cannot easily be handled by a derivative-based optimization method as the max function is non-smooth. However, as indicated in [31], the max function can be approximated by the convex *log-sum-exp* function; in that case, (7.9) can be approximated by the following smooth expression:

$$\begin{aligned} T_{k+1,i} =& F_i(\mathbf{T}_k, \dot{Q}_{k,i}, \dot{m}_{k,i}, \Delta t) \\ &+ \theta_{i,i-1} \frac{1}{\mu} \log\big(e^0 + e^{\mu(T_{k,i-1} - T_{k,i})}\big) \\ &- \theta_{i,i+1} \frac{1}{\mu} \log\big(e^0 + e^{\mu(T_{k,i} - T_{k,i+1})}\big), \end{aligned} \tag{7.10}$$

where the parameter μ is a scaling factor to make the max approximation sharper. The specific value of μ should be selected according to the specific application so that, while the approximation of the max function is sharp, there are no numerical issues. In the case of heat storage vessels, considering the range of temperature differences between consecutive layers, we have found $\mu = 10$ to be a reasonable value.

With this new approximation, the model for the vessel dynamics includes buoyancy, is represented by a smooth function, and can be integrated in any derivative-based optimization framework.

EMPIRICAL ANALYSIS

As indicated in the previous section, an important requirement for the proposed approximation to work is to have buoyancy effects whose time span is large in comparison with the simulation time step.

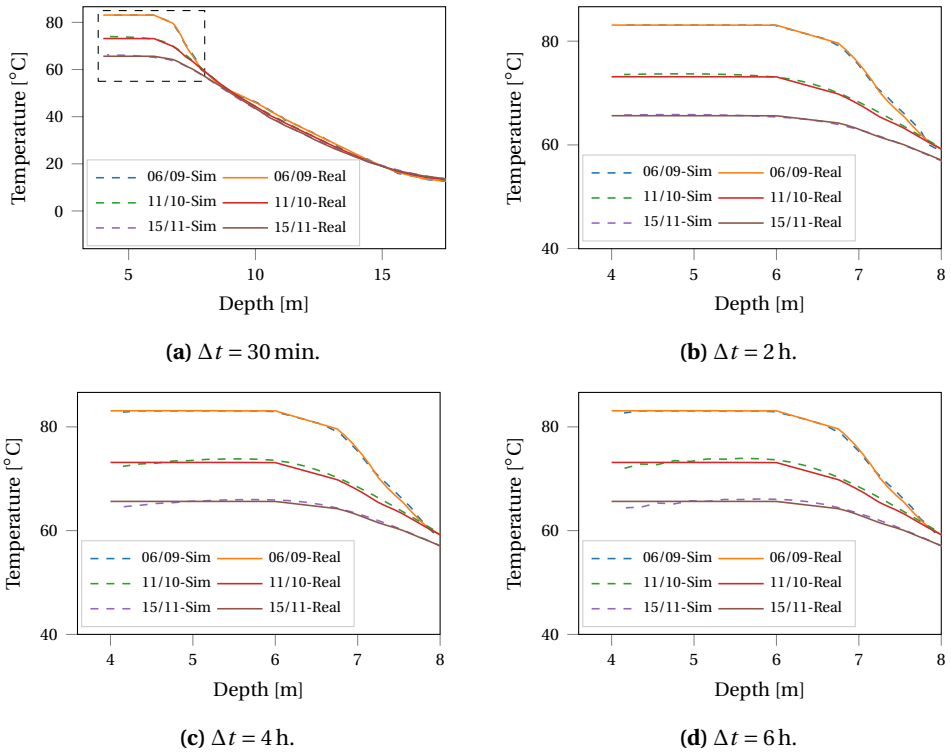
An example of a relative slow buoyancy effect is the natural recirculation of the fluid due to the higher heat losses in the top layer. In particular, as the top layer has a larger contact area with the environment, it suffers larger heat losses than the layers below it; as a result, the top layer reaches lower temperatures and the fluid from the lower layers rises to the top. As this effect depends on heat losses, it is a very slow process; in the case of the Ecovat vessel, it can be shown that, even for time steps Δt as large as 2 hours, the proposed model can accurately model this natural buoyancy effect.

This can be observed in Figure 7.4, where a simulation of the system using the slow buoyancy model is compared for different time steps Δt against the real data in the seasonal vessel during a 2-months cycle when the system was undisturbed, i.e. no heat was added or extracted (note that the simulation was done after estimating the model parameters; for the details of the parameter estimation we refer to Section 7.4). As can be observed, both $\Delta t = 30$ min and $\Delta t = 2$ h are small enough and a single mixing iteration per time step suffices to correct the buoyancy effect: the larger heat losses of the top layer cannot be noticed as the model correctly represents the mixing of the lower layers. However, as Δt increases to 4-6 h, the effect starts to be noticeable: the rate of heat losses in the top layer is larger than the update frequency of the buoyancy effect and the upper layers appear to have a lower temperature than the layers below it.

While the proposed model is very accurate for slow buoyancy effects, it is not for the faster buoyancy effects that appear due to charging/discharging. As indirect/direct charging introduces a much larger heat rate than heat losses, the corresponding buoyancy effects take place in a much shorter time span. This can be observed in Figure 7.5, where the simulated slow model is compared for different time steps Δt against the real evolution of the vessel during a 3-week charging period. During that period, the vessel is charged via its heat buffer 3, i.e. right in the middle of the vessel; during that time there is a moment where the temperature in the middle of the tank is equal to the temperature of the layers above and mixing due to buoyancy starts to occur. Although the proposed slow model still works, it only does so accurately for small time steps Δt. Particularly, while $\Delta t = 5$ min can correctly represent the buoyancy effects, $\Delta t = 30$ min already leads to an inconsistent system state where the middle temperatures are higher than the top ones. For even larger Δt the situation worsens: not only does the difference in temperature between the middle and top layers increase, but numerical artifacts such as oscillations appear in the simulated trajectory.

7.3.3. MODELING FAST BUOYANCY EFFECTS

Based on the previous results, it is clear that to model fast buoyancy effects a different approach is needed. In this section, we propose a model for the buoyancy effects that appear when charging and discharging the vessel. For the sake of simplicity, the model is first derived for the case of indirect charging. Then, it is extended to the case of indirect discharging and direct charging/discharging. Similarly, during the model derivation, we will assume that all layers have the same volume; nonetheless, at the end of the section, we will briefly explain the extension to layers with different volumes.



(a) $\Delta t = 30\,\text{min}$. **(b)** $\Delta t = 2\,\text{h}$.

(c) $\Delta t = 4\,\text{h}$. **(d)** $\Delta t = 6\,\text{h}$.

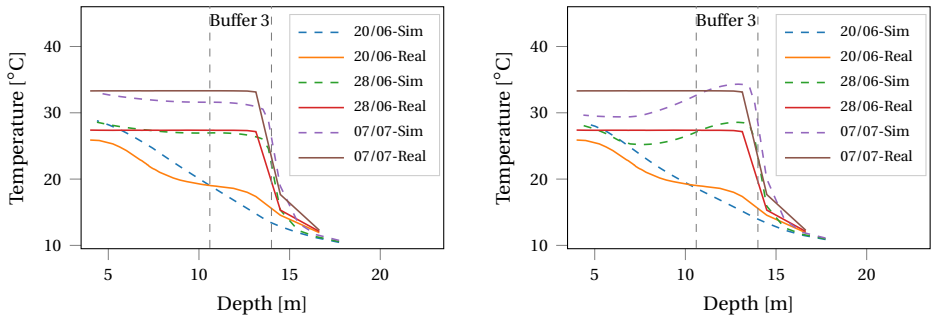
Figure 7.4: Effect of Δt when using the slow buoyancy model for slow buoyancy effects. The plot compares real vs. simulated trajectories of the Ecovat vessel during a 2-month cycle where the vessel is undisturbed. As buoyancy is slow, a medium-size Δt, e.g. 30 min or 2 h, can model the buoyancy effect. However, as Δt increases, to e.g. 4–6 h, the effect starts to be noticeable.

EMPIRICAL OBSERVATION

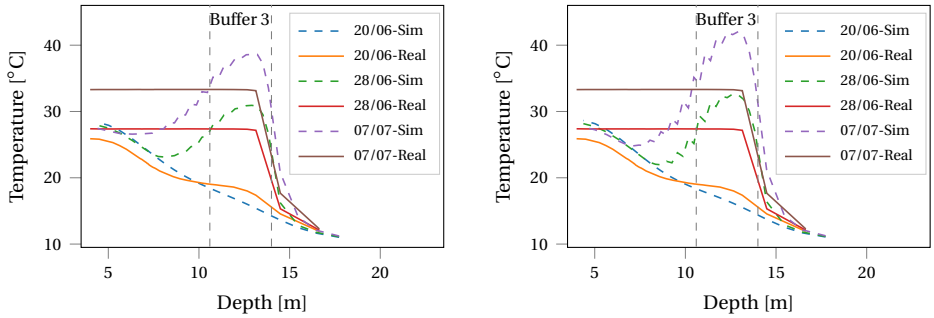
When observing the buoyancy effects due to indirect charging, what we can effectively see is that any layer on top of the charged layer with a lower or equal temperature is also charged. Particularly, we empirically observe that, when applying heat to a specific layer, the heat is homogeneously distributed across the charged layer and any layer above it with an equal or lower temperature.

This concept is better understood with a simple example. Consider a simple vessel with 4 layers as represented in Figure 7.6. Let us analyze what the above effect means in terms of heating the third layer: if T_3 is equal or lower than T_1 or T_2, any heat applied to the first or second layer will be equally distributed to the third layer; similarly, if T_3 is equal or larger than T_4, any heat applied to the third layer will be effectively divided between the third and the fourth layer. Mathematically, this means the third layer is indirectly heated by the following amount:

$$\alpha_3 \dot{Q}_3 + \alpha_2 \dot{Q}_2 + \alpha_1 \dot{Q}_1 \tag{7.11}$$



(a) $\Delta t = 5$ min. **(b)** $\Delta t = 30$ min.



(c) $\Delta t = 2$ h. **(d)** $\Delta t = 6$ h.

Figure 7.5: Effect of Δt when using the slow buoyancy model for fast buoyancy effects. The plot compares real vs. simulated trajectories of the Ecovat vessel during a 3-week cycle where the third heat buffer is charged. In this scenario, the slow model only captures buoyancy effects when using a small Δt, e.g. 5 min.

with

$$\alpha_3 = \begin{cases} 1 & \text{if } T_3 \leq T_4 \\ 1/2 & \text{if } T_3 > T_4 \end{cases} \tag{7.12}$$

$$\alpha_2 = \begin{cases} 0 & \text{if } T_2 \leq T_3 \\ 1/2 & \text{if } T_3 < T_2 \leq T_4 \\ 1/3 & \text{if } T_3 < T_2 \text{ and } T_4 < T_2 \end{cases} \tag{7.13}$$

$$\alpha_1 = \begin{cases} 0 & \text{if } T_1 \leq T_3 \\ 1/3 & \text{if } T_3 < T_1 \leq T_4 \\ 1/4 & \text{if } T_3 < T_1 \text{ and } T_4 < T_1 \end{cases} \tag{7.14}$$

MATHEMATICAL MODELING

Formally, the above observation means that, at any time step k, the i^{th} layer in the vessel is heated by the following amount:

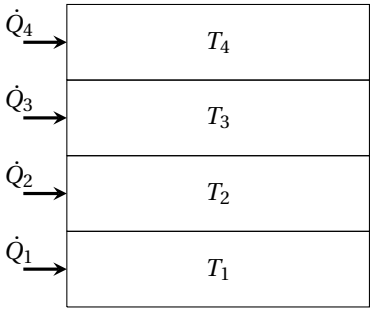


Figure 7.6: Simplified heat storage vessel with 4 layers and indirect charge/discharge.

$$\sum_{l=1}^{i} \dot{Q}_{k,l} \cdot \frac{\mathbb{1}_{T_{k,l} \geq T_{k,i}}}{\sum_{j=l}^{n_L} \mathbb{1}_{T_{k,l} \geq T_{k,j}}} \tag{7.15}$$

with $\mathbb{1}_S$ defining the indicator function:

$$\mathbb{1}_S = \begin{cases} 1 & \text{if } S \text{ is true,} \\ 0 & \text{otherwise.} \end{cases} \tag{7.16}$$

Using this empirical observation, we can now extend (7.2) to include the buoyancy effects due to indirect charging. In particular, by replacing $\dot{Q}_{k,i}$ by (7.15), these effects can be included in the model.

SMOOTH APPROXIMATION

The model derived so far is non-smooth and non-continuous. As with the initial model proposed for the slow buoyancy effects, an approximation is needed to transform (7.15) into a smooth and continuous expression. Analyzing (7.15), it can be seen that its non-smoothness and its non-continuity come from the inclusion of Heaviside step functions:

$$H(T_1 - T_2) = \begin{cases} 1 & \text{if } T_1 - T_2 \geq 0 \\ 0 & \text{else.} \end{cases} \tag{7.17}$$

Therefore, the only requirement to obtain a smooth and continuous model is to replace the step functions by a smooth and continuous approximation. One of the most popular approximations of the step function is the *logistic* or *sigmoid* function [100]:

$$S(T_1 - T_2) = \frac{1}{1 + e^{-\sigma(T_1 - T_2)}}, \tag{7.18}$$

where σ is a scaling parameter that indicates how sharp the logistic function is, i.e. the larger the σ the closer the logistic function is to the real step function. In all the experiments carried out in this research, $\sigma = 1$ was selected as it was empirically shown to be an appropriate value.

Using this approximation, we can finally establish the smooth and continuous approximation of the buoyancy effects due to indirect charging. Particularly, by replacing $\dot{Q}_{k,i}$ in (7.15) by:

$$\dot{Q}'_{k,i} = \sum_{l=1}^{i} \dot{Q}_{k,l} \cdot \frac{S(T_{k,l} - T_{k,i})}{\sum_{j=l}^{n_L} S(T_{k,l} - T_{k,j})}, \tag{7.19}$$

the buoyancy effects due to indirect charging can be included within the discrete system dynamics.

MODELING INDIRECT DISCHARGING

While the proposed model was derived for the case of indirect charging, it can be easily extended to the case of indirect discharging. When discharging the vessel, the effect is the opposite: any layer below the layer that is discharged that has a temperature higher than or equal to the discharged layer will also be discharged in an equally distributed manner. This effect can be easily modeled using the same approximations as for the charging case, i.e. for the indirect discharging, $\dot{Q}_{k,i}$ should be replaced by:

$$\dot{Q}'_{k,i} = \sum_{l=i}^{n_L} \dot{Q}_{k,l} \cdot \frac{S(T_{k,i} - T_{k,l})}{\sum_{j=1}^{l} S(T_{k,j} - T_{k,l})}, \tag{7.20}$$

MODELING DIRECT CHARGING AND DISCHARGING

In the case of direct charging, the empirical observation is very similar: the heat of the incoming flow will be distributed across any layer on top of the input layer that has a temperature equal to or lower than the temperature of the incoming flow. Therefore, defining by $T^{\text{in}}_{k,i}$ the temperature of the incoming flow at time step k and layer i, the approximation for direct charging can be included within the discrete system dynamics by replacing $\dot{m}_{k,i}\,(T^{\text{in}}_{k,i} - T_{k,i})$ in (7.15) by:

$$\dot{m}'_{k,i}\,\Delta T^{\text{in}\,\prime}_{k,i} = \sum_{l=1}^{i} \dot{m}_{k,l} \cdot (T^{\text{in}}_{k,l} - T_{k,i}) \cdot \frac{S(T^{\text{in}}_{k,l} - T_{k,i})}{\sum_{j=l}^{n_L} S(T^{\text{in}}_{k,l} - T_{k,j})}. \tag{7.21}$$

It is important to note the difference w.r.t. indirect charging: the temperature of layer l at time step k, i.e. $T_{k,l}$, is replaced by the temperature of the incoming flow at layer l at time step k, i.e. $T^{\text{in}}_{k,l}$. Using the same reasoning, the approximating for direct discharging can be similarly derived by replacing $\dot{m}_{k,i}\,(T^{\text{in}}_{k,i} - T_{k,i})$ by:

$$\dot{m}'_{k,i}\,\Delta T^{\text{in}\,\prime}_{k,i} = \sum_{l=i}^{n_L} \dot{m}_{k,l} \cdot (T_{k,i} - T^{\text{in}}_{k,l}) \cdot \frac{S(T_{k,i} - T^{\text{in}}_{k,l})}{\sum_{j=1}^{l} S(T_{k,j} - T^{\text{in}}_{k,l})}, \tag{7.22}$$

LAYERS WITH DIFFERENT VOLUME

For the sake of simplicity, all the derivations have been performed assuming constant volume across the layers, i.e. assuming that $A_i \Delta z_i$ was constant for each layer i. However, the model can be easily extended to the case where the volumes are not constant. In this case, it can be shown that (7.19) is equivalent to:

$$\dot{Q}'_{k,i} = \sum_{l=1}^{i} \dot{Q}_{k,l} \cdot \frac{A_i \Delta z_i \, S(T_{k,l} - T_{k,i})}{\sum_{j=l}^{n_L} A_j \Delta z_j \, S(T_{k,l} - T_{k,j})}. \tag{7.23}$$

Similarly, the three other cases can be derived using the same modification, i.e. rescaling the functions $S(\cdot)$ by the corresponding volumes.

7.4. PARAMETER ESTIMATION AND VALIDATION OF THE MODEL

IN order to validate the proposed model, we use it to estimate the parameters of the real stratified STESS introduced in Section 7.2. First, we estimate the model parameters using a training dataset and evaluate the model performance in an out-of-sample dataset. Next, we compare the obtained parameters with their ideal value considering the structure and materials of the vessel.

7.4.1. DATA

The employed data for validating the model consist of a period of 7 months divided in two different cycles:

1. A 5-month cycle (15/04/2017-10/09/2017) where the buffer 3, 4, and 5 (the top 3 buffers of the vessel) are charged interchangeably.
2. A 2-month cycle (10/09/2017-15/11/2017) where the system is left in a steady state and natural discharge occurs.

From this period, there are two types of measurements available:

1. The temperature T inside the vessel at different depths; these are sampled with a frequency of 1 week.
2. The heat $\dot{Q}$ introduced in each of the 5 heat buffers; these are sampled with a frequency of 15 minutes.

These measurements of the input heats and the temperatures are respectively depicted in Figure 7.7 and Figure 7.8.

7.4.2. MODEL PARAMETERS

Unlike most heat storage vessels, the considered vessel has its heat exchangers embedded into the concrete wall. As a result, the storage medium for the heat is a mixture of fluid and concrete and, unlike regular heat storage tanks, the parameters that define the dynamics of the vessel are not specific to the fluid but to that mixture of fluid and concrete. More specifically, the density ρ, the specific heat c_p, and the diffusivity α are

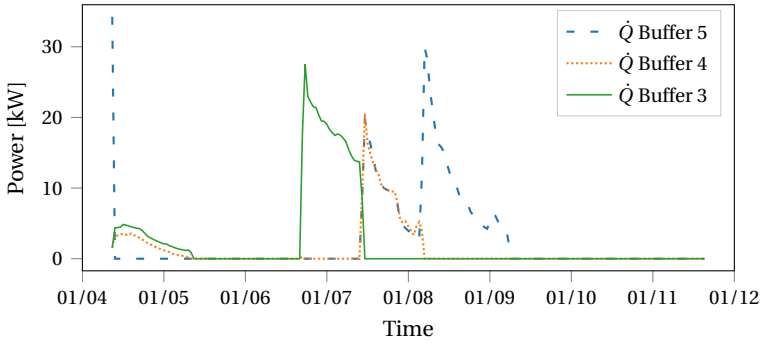


Figure 7.7: Heat introduced in the vessel during the 7-month period considered. In this experiment, the top three buffers are charged interchangeably.

Table 7.2: Model parameters to be estimated.

Parameter	Description
α [m^2/s]	Diffusivity of the storage medium.
λ [m K/J]	Coefficient of the input heat. As the cross-sectional area A does not vary along the tank, the same $\lambda_i = \frac{1}{\rho c_p A_i}$ applies to each layer i.
β [1/s]	Coefficient of heat losses of the inner layers. As the cross-sectional A and the perimeter P do not vary along the tank, the same $\beta_i = \frac{P_i k}{\rho c_p A_i}$ applies to each inner layer i.
β_1 [1/s]	Coefficient of heat losses of the bottom layer, which differs from the general β as the bottom layer has a larger surrounding wall area.
β_{n_L} [1/s]	Coefficient of heat losses of the top layer, which differs from the general β as the top layer has a larger surrounding wall area.
T_∞ [°C]	Temperature of the surrounding terrain. While it is known that the surrounding ground has a temperature of 11-13 °C, the exact value is unknown.

defined by the properties of the fluid (water in the case of the considered vessel) and the concrete in the walls. Considering this observation, equations (7.1) and (7.3), and the fact that the considered vessel works with indirect charge/discharge, it can be shown that there are 6 unknown parameters defining the system dynamics; these are listed and described in Table 7.2.

7.4.3. ESTIMATION PROBLEM

Define by $\mathbf{T}_k = [T_{k,1}, \dots, T_{k,n_L}]^\top$ the temperature distribution at time step k, by $\dot{\mathbf{Q}}_k = [\dot{Q}_{k,1}, \dots, \dot{Q}_{k,n_L}]^\top$ the input heat at the same time step, by $\boldsymbol{\varphi} = [\alpha, \lambda, \beta, \beta_1, \beta_{n_L}, T_\infty]^\top$ the vector of unknown parameters, and by n_L the number of layers used to model the vessel dynamics. Let us also define the discrete system dynamics by $T_{k+1,i} = G_i(\mathbf{T}_k, \dot{\mathbf{Q}}_k, \Delta t, \boldsymbol{\varphi})$, with Δt the length of the discretization interval and with $G_i(\cdot)$ the numerical integration of (7.1) including the two proposed buoyancy models; i.e. $G_i(\cdot)$ is equivalent to (7.10) but

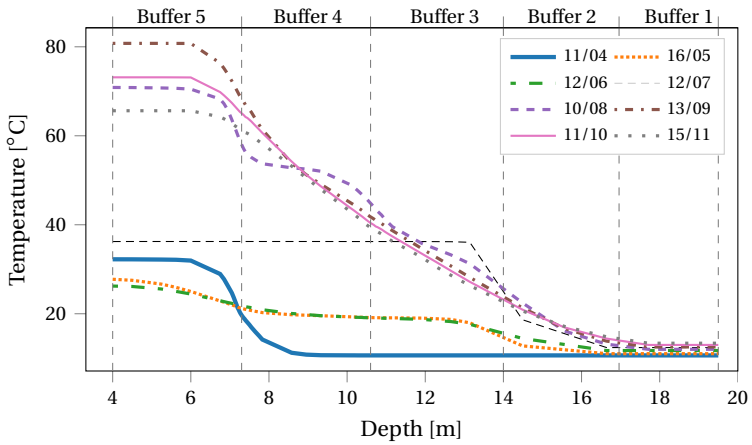


Figure 7.8: Temperature evolution in the vessel at intervals of 1 month during the 7-month period considered. The black dashed lines represent the division between the 5 heat buffers within the vessel. The depth is measured using the reference system defined in Figure 7.3.

replacing $\dot{Q}_{k,i}$ by (7.19) and (7.20). Then, the numerical optimization problem that is solved to estimate the model parameters is given by:

$$\underset{\boldsymbol{\varphi}, \mathbf{T}_1, \ldots, \mathbf{T}_{N+1}}{\text{minimize}} \quad \sum_{j \in \mathcal{S}} \|\bar{\mathbf{T}}_j - \mathbf{T}_j\|_2^2 \tag{7.24a}$$

subject to

$$T_{k+1,i} = G_i(\mathbf{T}_k, \dot{\mathbf{Q}}_k, \Delta t, \boldsymbol{\varphi}) \quad \text{for } k = 1, \ldots, N, \quad \text{for } i = 1, \ldots, n_{\mathrm{L}} \tag{7.24b}$$

where $\mathcal{S}$ represents the set of time indices when temperature measurements are available[1], i.e. $\mathcal{S} = \{\, k \mid k = 1, \ldots, N+1,\ \bar{\mathbf{T}}_k \text{ exists}\}$, $\bar{\mathbf{T}}_k$ the temperature measurement at time step k, and $N+1$ the number of discrete time points.

Note that the above formulation could also include the input heat $\dot{Q}$ as optimization variable. However, for the sake of simplicity, it is assumed that the measurements of $\dot{Q}$ are error-free. In addition, to avoid interpolations in $\dot{Q}$, Δt is selected as a multiple of the sampling period of $\dot{Q}$, i.e. $\Delta t = a \cdot 15\,\text{min}$ with $a \in \mathbb{N}$.

7.4.4. EXPERIMENTAL SETUP

To estimate and validate the model, the experimental setup is divided into two distinguishable parts: model validation and parameter estimation.

MODEL VALIDATION

In a first part, the model is estimated using a training dataset and validated with an out-of-sample test dataset. The goal of this part is to ensure that the estimated model does not overfit the training dataset and to ensure that its accuracy is in agreement with the

[1] Note that the temperature sampling period of one week will always be larger than Δt.

existing literature. To verify that the model validation is independent from the training/test dataset split, the parameter estimation is done 50 times and each time a random training/test dataset split is considered. Then, the average performance is compared with the performance of the existing models from the literature. To perform the splits, the training/test datasets are built using the following convention:

1. Training dataset: A dataset that spans 5.5 months and includes part of the charging period and part of the natural discharging period.

2. Test dataset: An out-of-sample dataset of 1.5 months that includes three weeks of the charging period and three weeks of the natural discharging period.

To randomize the splits, we consider that there are 25 weeks during the charging period and we randomly select an integer i between 1 and 23. Then, weeks $i, i+1$, and $i+2$ are used for the test dataset and the remaining 22 for the training dataset. Next, the process is repeated for the natural discharging period considering that in this case the number of weeks is 9, i.e. the randomly selected integer varies between 1 and 7. The results of this experiment are described in detail in Section 7.4.5.

PARAMETER ESTIMATION

After validating the model, the parameter estimation is performed on the full dataset to ensure that the parameters are estimated using as much information as possible. In addition, as the estimation problem (7.24) is non-convex, the problem is solved using multi-start optimization. In particular, the optimization problem is solved multiple times using different initial guesses for the optimization variables and the optimal parameters are selected from the optimization run that obtains the best fit, i.e. the lower optimal cost. For this application, the optimization problem is solved 30 times[2] and the initial guesses are randomly generated using Gaussian distributions. For the six model parameters, i.e. $\alpha, \lambda, \beta, \beta_1, \beta_{n_L}, T_\infty$, the means of their Gaussian distributions are selected as their theoretical values assuming water as storage medium (see the first column of Table 7.4). For the temperature variables, at those time points where measurements are available, the means of the Gaussian distributions are selected as the measured temperatures; for the time points where measurements are not available, the means are selected by a linear interpolation using the closest measurements in time. The standard deviations of the distributions are selected as half the value of the means. The results of this experiment are described in Section 7.4.6.

IMPLEMENTATION DETAILS

To estimate the model, we consider a discrete time step Δt of 2 hours and an explicit Euler scheme to perform the numerical integration of the dynamics. In terms of the spatial discretization, we consider different thicknesses for the different heat buffers of the vessel. Particularly, as the top buffers are more often charged and discharged, they employ a coarser spatial discretization. The discretization uses 23 layers distributed as follows:

[2] For the considered application, it was empirically observed that after 20-30 iterations the best optimal solution did not vary much anymore.

- 6 layers of 550 mm for the top heat buffer (buffer 5).
- 6 layers of 550 mm for the heat buffer 4.
- 6 layers of 550 mm for the heat buffer 3.
- 3 layers of 967 mm for the heat buffer 2.
- 2 layers of 1450 mm for the bottom heat buffer (buffer 1).

The problem is modeled in `python` using the mathematical modeling framework `CasADi` [11] and solved using the interior point solver `Ipopt` [237].

7.4.5. MODEL VALIDATION

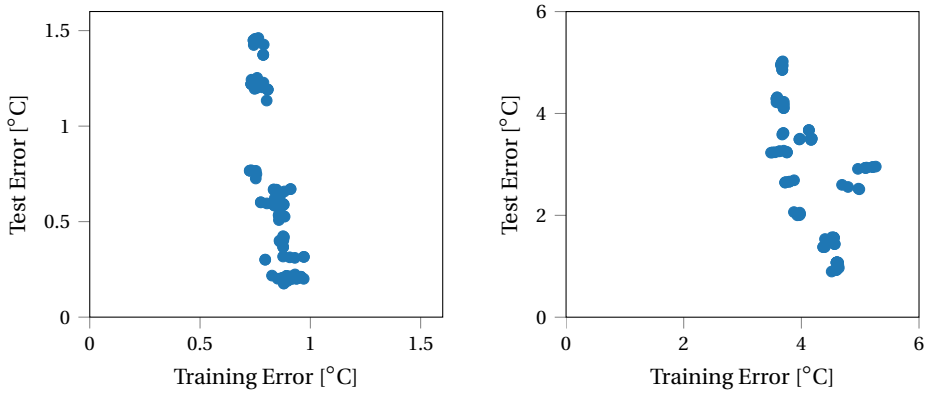
To validate the model, the parameters are estimated for 50 different training/test dataset splits and, for each split, the model is evaluated in terms of the mean and maximum absolute errors for both the training and the test dataset. Then, the model is considered to be valid if its performance is independent of the training and test datasets and if its accuracy is in agreement with the existing literature. A summary of the results is listed in Table 7.3, which presents the average and standard deviation of the mean and maximum absolute errors across the 50 splits. In addition, the distribution of the two error metrics is depicted in Figure 7.9. As can be observed, the average of both metrics in the training and test datasets is of the same order of magnitude, with the average errors in the test datasets being smaller. This last result is expected as the test datasets comprise a shorter time span and should therefore have lower errors.

Another observation that can be made is that the variance of the errors in the training datasets is smaller than the variance in the test datasets. This effect is also expected as the size of the training datasets is much larger than the size of the test datasets (a training dataset spans 5.5 months while a test dataset only 1.5 months). More specifically, as two training datasets can be at most 27% different (over the 5.5 months two training datasets can at most differ by a month and a half), it is normal for them to have similar errors. By contrast, two test datasets can show more variable errors as they can be 100% different (a test dataset only contains 1.5 months of the total 7 months of data).

Table 7.3: Comparison between training and test errors across the 50 training/test dataset splits in terms of the average and standard deviation of the mean absolute errors and the average and standard deviation of the maximum absolute errors.

Metric	Training Set	Test Set
Average Mean Absolute Error	0.84 °C	0.64 °C
Std. Mean Absolute Error	0.07 °C	0.39 °C
Average Maximum Absolute Error	4.14 °C	2.88 °C
Std. Maximum Absolute Error	0.50 °C	1.24 °C

Based on the obtained results it is clear that no significant differences exist between the performance of the model in the training datasets and the test datasets. Moreover, considering that over a period of 5.5 months the mean absolute errors are below 1 °C and



(a) Mean absolute error

(b) Maximum absolute error

Figure 7.9: Mean and maximum absolute errors of the training and test datasets for the 50 different dataset random splits. As the average of both metrics in the training and test datasets is of the same order of magnitude, it is clear that no significant differences exist between the performance of the model in the training and test datasets.

the maximum errors are below 5 °C, it can be stated that the accuracy of the proposed model is in agreement with the existing literature [62]. Using these two observations it can be concluded that: (i) the proposed model is valid; (ii) the model is accurate; (iii) the model does not overfit the data.

7.4.6. PARAMETER ESTIMATION

After the model has been validated, the parameters of the model are estimated using the full dataset and multi-start optimization. In addition, to further validate that the obtained parameters are reasonable, two comparisons are done:

1. The estimated parameters are compared for two different spatial discretizations: 23 layers (as before) and 31 layers. As the parameters are independent from the number of layers, a valid model should have similar parameters in both estimations.

2. The estimated parameters are compared against the theoretical parameters in the case that the heat storage medium is only water or only concrete. As motivated in Section 7.4.2, since the considered vessel has its heat exchangers embedded in the concrete wall, the parameters of the dynamics are not specific to the fluid but to a mixture of fluid and concrete. Therefore, if the model is valid, the estimated parameter are expected[3] to be in the range of the parameter values in the case of having only water as storage medium and the parameter values in the case of having only concrete.

[3] Assuming that, if the model parameters depend on two materials, their value is a linear combination of the parameters defined for each material.

The results of this experiment are listed in Table 7.4. As can be observed, the estimated parameters satisfy the two conditions for the model to be valid:

1. The difference in the estimated parameters for different numbers of layers is very small. Particularly, except for the heat loss parameter β_1 in the lowest layer, all other parameters show very small variations. In the case of β_1, this variation can be easily explained. In detail, as can be observed from Figure 7.8, the temperature at the bottom of the vessel only varies between 11 °C and 13 °C, which in turn is in the same range as the ground temperature T_∞. As a result, the heat losses in the lower layer have an almost negligible effect on the overall accuracy of the model and, as the problem is non-convex, there might exist several local minima with very similar accuracies but very different β_1 values.

2. All the estimated parameters are within the range of theoretical parameters corresponding to the cases of having only water as storage medium and having only concrete.

Table 7.4: Results of the parameter estimation for different numbers of layers. The estimation is compared against the ideal theoretical parameters assuming that the construction dimensions are perfectly known and considering that the material for storing heat in the vessel is either only water or only concrete.

	Theoretical Values		**Estimated Parameters**	
Parameter	**Water**	**Concrete**	**23 Layers**	**31 Layers**
α [m^2/s]	$1.4{\cdot}10^{-07}$	$52.0{\cdot}10^{-07}$	$2.32\cdot10^{-07}$	$2.76\cdot10^{-07}$
λ [m K/J]	$2.4\cdot10^{-09}$	$5.7\cdot10^{-09}$	$3.49\cdot10^{-09}$	$3.48\cdot10^{-09}$
β [1/s]	$0.8\cdot10^{-08}$	$1.9\cdot10^{-08}$	$1.60\cdot10^{-08}$	$1.68\cdot10^{-08}$
β_1 [1/s]	$2.7{\cdot}10^{-07}$	$6.3\cdot10^{-07}$	$3.99\cdot10^{-07}$	$2.55\cdot10^{-04}$
β_{n_L} [1/s]	$5.7\cdot10^{-08}$	$13.5\cdot10^{-08}$	$9.62\cdot10^{-08}$	$10.15\cdot10^{-08}$
T_∞ [°C]	[11, 13]		13.03	13.09

7.4.7. DISCUSSION

Based on the obtained results, it can be claimed that the proposed model is a valid model that correctly identifies the system dynamics. In particular, the model accuracy in out-of-sample datasets is similar to the accuracy in training datasets and this accuracy is within the range of expected accuracy for a 1D model. In addition, the estimated parameters are within the range of values to be expected to correctly identify the physics behind the system dynamics, and they are independent of the spatial discretization employed. These four observations are clear indications that the model is correct and that it correctly represents the real physical behavior of heat storage vessels.

7.5. MODEL COMPARISON

As motivated in the introduction, the goal of the proposed model is to provide a smooth representation of the dynamics of heat storage vessels so that the model

can be employed in derivative-based optimization problems. In this section, we illustrate the benefits obtained in optimization problems when using the proposed smooth model instead of the non-smooth models from the literature. In detail, we compare the quality of the optimal solution and the computation cost of the smooth and non-smooth models in the same optimization setup: an optimal control application where a stratified STESS needs to satisfy a given heat demand over some time horizon while the electricity prices over the same period are known. The goal of the controller is to find the optimal charging strategy that minimizes the cost while satisfying the heat demand. As before, we consider as a case study the real stratified STESS introduced in Section 7.2.

7.5.1. SMOOTH VS NON-SMOOTH MODELS

As previously explained, because the 1D dynamical models proposed in the literature are non-smooth, they cannot be employed in optimization setups using derivative-based optimization methods that employ automatic differentiation. Instead, they either need to employ heuristic methods or derivative-based methods that use finite differences.

HEURISTIC METHODS

As heuristic methods can optimize black-box functions, they are a good fit for optimization problems where the gradient and Hessian matrix of the problem cannot be computed analytically. However, even though they can be used to integrate non-smooth models in optimization setups, they have two problems: (i) they provide no guarantee of finding a local minimum or even a feasible solution; (ii) they are usually computationally more expensive than derivative-based optimization methods.

FINITE DIFFERENCES

Another option for non-smooth models is to employ finite differences to compute the derivative-based information and then use the same derivative-based optimization algorithms that smooth methods can use. However, unlike smooth models for which automatic differentiation can be used, the cost of computing second-order and first-order derivative information via finite differences respectively grows quadratically and linearly with the number of optimization variables. As a result, finite differences can quickly become computationally infeasible for many optimization problems.

More specifically, denoting the cost of evaluating some function $f : x \in \mathbb{R}^n \rightarrow \mathbb{R}$ by c_f, computing the gradient $\nabla_x f$ and the Hessian matrix $\nabla_x^2 f$ via finite differences have approximate costs of $n \cdot c_f$ and $n \cdot (n+1) \cdot c_f$ respectively[4]. By contrast, using automatic differentiation, the cost of computing $\nabla_x f$ is not only independent of the number of variables n but lower than $4 \cdot c_f$; likewise, the cost of computing $\nabla_x^2 f$ only grows linearly with n and is bounded by $8 \cdot n \cdot c_f$ [93]. This comparison is summarized in Table 7.5.

A second disadvantage of using finite differences is the accuracy: computing the gradient and Hessian matrix via finite differences reduces the numerical precision w.r.t. to the original function f and introduces truncation errors [93].

[4] Note that the cost of computing the Hessian is derived for the general case of having a non-symmetric Hessian, i.e. the Hessian of a non-smooth function, as this represents the real cost of the non-smooth models from the literature.

Table 7.5: Comparison of the computation cost of using finite differences versus using automatic differentiation.

Function	Finite Differences	Automatic Differentiation
$\text{Cost}(f : x \in \mathbb{R}^n \to \mathbb{R})$	c_f	c_f
$\text{Cost}(\nabla_x f)$	$(n+1)\cdot c_f$	$< 4 \cdot c_f$
$\text{Cost}(\nabla_x^2 f)$	$(n+1)^2 \cdot c_f$	$< 8 \cdot n \cdot c_f$

7.5.2. OPTIMAL CONTROL PROBLEM

To evaluate and compare the performance of the non-smooth models vs. the proposed smooth model, we consider an *optimal control problem (OCP)* where a stratified thermal storage vessel needs to satisfy a given heat demand over a horizon while ensuring that its costs are minimized. For this, the spatial discretization of 23 layers described in Section 7.4 is considered again. In addition, it is assumed that only the top 4 buffers (buffer 2 to buffer 5) can be charged and discharged and that each buffer has independent heat exchangers for charging and discharging[5]. The definition of the different OCP variables and parameters are listed in Table 7.6. Using these definitions, the OCP can then be defined as:

$$\underset{\substack{\mathbf{T}_1,\dots,\mathbf{T}_{N+1},\dot{\mathbf{Q}}_1^{\text{in}},\dots,\\ \dot{\mathbf{Q}}_N^{\text{in}},\dot{\mathbf{Q}}_1^{\text{out}},\dots,\dot{\mathbf{Q}}_N^{\text{out}}}}{\text{minimize}} \quad \sum_{k=1}^{N} p_k \cdot \Delta t \cdot \sum_{j=2}^{5} \dot{Q}_{k,j}^{\text{in}} + \gamma \cdot \|\mathbf{T}_1 - \mathbf{T}_{N+1}\|_2^2 \tag{7.25a}$$

subject to

$$\mathbf{T}_1 = \bar{\mathbf{T}}_1, \tag{7.25b}$$

$$T_{k+1} = \mathbf{G}\big(\mathbf{T}_k, \dot{\mathbf{Q}}_k^{\text{in}}, \dot{\mathbf{Q}}_k^{\text{out}}, \Delta t\big), \qquad \text{for } k = 1,\dots,N, \tag{7.25c}$$

$$T_\infty \le \mathbf{T}_k \le T_{\max}, \qquad \text{for } k = 1,\dots,N, \tag{7.25d}$$

$$\sum_{j=2}^{5} \dot{Q}_{k,j}^{\text{out}} = \hat{\dot{Q}}_k^{\text{d}}, \qquad \text{for } k = 1,\dots,N, \tag{7.25e}$$

$$0 \le \dot{Q}_{k,j}^{\text{in}} \le \dot{m}_{\max} c_{\text{p}} \Big(T_{\text{c}} - \frac{1}{n_{\text{L}j}} \sum_{i \in \mathcal{L}_j} T_{k,i}\Big)\Big(1 - e^{\frac{-k_{\text{he}}}{\dot{m}_{\max} c_{\text{p}}}}\Big), \qquad \text{for } k = 1,\dots,N,$$

$$\text{for } j = 2,3,4,5, \tag{7.25f}$$

$$0 \le \dot{Q}_{k,j}^{\text{out}} \le \dot{m}_{\max} c_{\text{p}} \Big(\frac{1}{n_{\text{L}j}} \sum_{i \in \mathcal{L}_j} T_{k,i} - T_{\text{d}}\Big)\Big(1 - e^{\frac{-k_{\text{he}}}{\dot{m}_{\max} c_{\text{p}}}}\Big), \qquad \text{for } k = 1,\dots,N,$$

$$\text{for } j = 2,3,4,5. \tag{7.25g}$$

From the optimization problem above, several facts are to be noted:

- The periodical penalty cost $\gamma \cdot \|\mathbf{T}_1 - \mathbf{T}_{N+1}\|_2^2$ ensures that the optimal solution does not leave the vessel discharged.

[5] This is the configuration in the real system: the top four layers have independent heat exchangers, but the bottom layer has none.

- The constraint (7.25e) ensures that at each time step k the heat extracted from the 4 buffers is equal to the heat demand.
- In (7.25f)–(7.25g), $\frac{1}{n_{\mathrm{L}j}} \sum_{i \in \mathcal{L}_j} T_{k,i}$ is simply the average temperature of heat buffer i.
- The constraint (7.25f) ensures that the input heat is lower than the maximum input heat, which is defined by the maximum temperature and flow in the heat exchanger and by the average temperature in the vessel.
- Equation (7.25g) ensures that the average temperature in a given heat buffer is enough to satisfy the heat demand required from that buffer.

Table 7.6: Parameters and variables of the optimal control problem considered for evaluating the proposed smooth model.

Parameter	Units	Description
N		Number of steps for the time horizon of the OCP.
$\mathbf{T}_k = [T_{k,1}, \dots, T_{k,23}]^\top$	K	Temperature distribution at time step k.
$\hat{Q}_k^{\mathrm{d}}$	W	Given heat demand at time step k.
p_k	€/J	Given heat price at time step k.
$\dot{\mathbf{Q}}_k^{\mathrm{in}} = [\dot{Q}_{k,2}^{\mathrm{in}}, \dots, \dot{Q}_{k,5}^{\mathrm{in}}]^\top$	W	Heat added to the four buffers at time step k.
$\dot{\mathbf{Q}}_k^{\mathrm{out}} = [\dot{Q}_{k,2}^{\mathrm{out}}, \dots, \dot{Q}_{k,5}^{\mathrm{out}}]^\top$	W	Heat extracted from the four buffers at time step k.
$\bar{\mathbf{T}}_1$	K	Initial observed temperature.
$T_{k+1} = \mathbf{G}(\mathbf{T}_k, \dot{\mathbf{Q}}_k^{\mathrm{in}}, \dot{\mathbf{Q}}_k^{\mathrm{out}})$	K	Discrete system dynamics with $\mathbf{G}(\cdot) = [G_1(\cdot), \dots, G_{n_{\mathrm{L}}}(\cdot)]^\top$, where $G(\cdot)$ represents the dynamics of each layer i as defined in (7.24b).
$\mathcal{L}_j$		Set of indices of the layers in heat buffer j
$n_{\mathrm{L}j}$		Number of layers in heat buffer j
$T_{\max}$	K	Maximum temperature in the vessel.
$\dot{m}_{\max}$	kg/s	Maximum water flow through the heat exchangers.
T_{d}	K	Minimum input temperature of the discharge heat exchanger.
T_{c}	K	Maximum input temperature of the charge heat exchanger.
c_{p}	J/(K·kg)	Specific heat of water.
k_{he}	W/K	Heat exchanger coefficient.

7.5.3. COMPARISON SETUP

To compare the performance of the smooth model vs the non-smooth models, the OCP is first solved using the proposed smooth model and a second-order Newton-based optimization method with automatic differentiation. Then, the same OCP is solved using the equivalent non-smooth model and the following optimization techniques:

1. The *particle swarm optimization (PSO)* [124], one of the most widely used heuristic algorithms that has often been used for optimizing different types of energy storage systems [98, 154, 250]. The algorithm is run for 3 different numbers of particles: 100, 1000, and 10000 particles.

2. The *Markov chain Monte Carlo (MCMC)* using the Metropolis–Hastings algorithm [101], a method used in the literature to estimate the parameters of a non-smooth heat vessel model [62], and also used for other energy demand applications [246].

3. The *tree-structured Parzen estimator (TPE)* [25], a black-box optimization algorithm previously used in energy-related applications [133, 135, 137].

4. A second-order Newton method where the Hessian is computed using finite differences.

5. A first-order Newton method where the Hessian is approximated using the gradient through the *Broyden–Fletcher–Goldfarb–Shanno (BFGS)* algorithm [153], and the gradient is computed using finite differences.

7

7.5.4. IMPLEMENTATION DETAILS

In this section, we provide the details regarding the implementation of the OCP, including how the optimization problem differs for each of the optimization techniques.

PRICE AND DEMAND DATA

The heat demand used in the OCP is the realistic heat demand profile from a cluster of buildings in The Netherlands from 01/09/2016 to 31/10/2016. Similarly, the price of buying heat is assumed to be the price of buying the equivalent amount of electricity in the Dutch day-ahead electricity market in the same time period. Both quantities are depicted in Figure 7.10.

TIME RESOLUTION AND OPTIMIZATION HORIZON

All the OCPs are solved using a time resolution of 1 hour, and to study the effect of the number of optimization variables, the OCP is solved for 4 different horizons: 1 day, 1 week, 1 month, and 2 months.

MODELING OF CONSTRAINTS

For the Newton-based method using the smooth model, the system dynamics are defined as in the OCP above: using a multiple shooting algorithm [27] where the system state (temperature profile) is an optimization variable and the dynamics are ensured via constraint (7.25c).

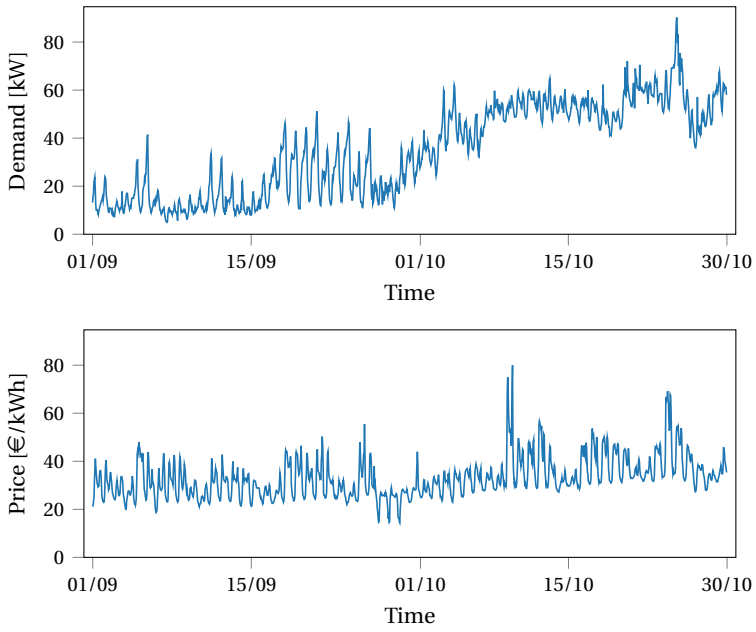


Figure 7.10: Considered heat demand (top) and heat prices (bottom) in the OCP. The heat demand is the real heat consumption of a group of buildings. The prices are the real market prices of the day-ahead electricity market in The Netherlands.

For the heuristic methods, in order to reduce the size of the search space, only the heat inputs $\dot{\mathbf{Q}}^{\text{in}}$ and $\dot{\mathbf{Q}}^{\text{out}}$ are considered as optimization variables. More specifically, the system dynamics are implicitly defined by modeling each temperature profile $\mathbf{T}_k$ as a function of the initial temperature $\mathbf{T}_1$ and the previous heat inputs. As the cost of computing the Hessian via finite differences scales quadratically with the number of variables, the reduced OCP formulation is also employed for the Newton-based method that uses finite differences. In addition, for the heuristic methods, as the constraints (7.25e-7.25g) cannot be explicitly included, they are modeled as penalty costs in the objective function.

SOFTWARE

All experiments are implemented in `python`. The OCP using the smooth model is solved using `CasADi` [11] and `Ipopt` [237]. The same applies for the OCP using the non-smooth model and finite differences. For the heuristic methods, we consider the `hyperopt` [24] library for the TPE algorithm, the `pyswarm` library [148] for PSO, and an in-house library for MCMC.

7.5.5. RESULTS

The comparison results of solving the OCP via the different methods are listed in Tables 7.7 and 7.8. In particular, Table 7.7 compares the quality of the optimal solution as the cost of buying heat and uses a baseline that represents the cost of buying heat without

using the heat buffer, i.e. buying the heat demand at the actual market price. Table 7.8 compares the computation time required for each of the methods. In both tables, the combinations of an algorithm and a horizon that did not find a solution were marked with an x; this issue always occurred for one of two reasons:

1. In the case of heuristic methods, the problem was always the same: the solver was unable to find a feasible solution as the constraint (7.25f) limiting the maximum input heat was usually violated.

2. In the case of finite differences, the algorithm was unable to find a solution within 3 days. It is important to note that this bound of 3 days was randomly selected to avoid situations where the algorithms would run indefinitely.

Table 7.7: Comparison of the OCP optimal solution, i.e. cost in EUR, for different time horizons using different optimization methods. The best solutions per time horizon are marked in bold. The first row represents the cost of buying directly the heat without the heat buffer, i.e. buying the heat demand at the actual market price. Cells with an x represent cases where the method was unable to find a feasible solution either because the algorithm converged to an infeasible solution (in the case of heuristic methods) or because the algorithm did not converge within 3 days (in the case of finite differences methods).

	OCP Horizon [h]			
Optimization Method	24	168	720	1440
No Buffer	10.1	73.3	397.0	1761.1
Smooth model	**0**	**16.2**	**215.9**	**1108.9**
Finite-differences: BFGS	**0**	**16.2**	235.1	x
MCMC	**0**	19.6	343.8	x
PSO 10000	**0**	38.1	2570.4	x
PSO 1000	**0**	361.6	3305.0	x
PSO 100	73.5	783.9	x	x
TPE	41.6	x	x	x
Finite-differences: 2^{nd} order	**0**	x	x	x

After analyzing the obtained results, the superiority of using a smooth model with a Newton-based optimization method and automatic differentiation becomes clear. Particularly, the following observations can be made:

- The OCP solved with the smooth model is able to obtain the best solution for all possible horizons.

- Not only is the proposed approach the one with the best optimal solutions, but also the only one that outperforms the baseline across all horizons. In terms of economic savings, the proposed approach respectively reduces the baseline cost by 100%, 77%, 46%, and 37% for the 1-day, 1-week, 1-month, and 2-months horizons.

- In terms of accuracy, the majority of the alternative methods do not perform well: except for the shortest time horizon of 24 hours, 5 of the 7 alternative methods are

Table 7.8: Comparison of the computation time (in minutes) required to solve the OCP for different time horizons and using different optimization methods. The most cost efficient solutions at each time horizon are marked in bold. Cells with an x represent cases where the method was unable to find a feasible solution either because the algorithm converged to an infeasible solution (in the case of heuristic methods) or because the algorithm did not converge within 3 days (in the case of finite differences methods).

	OCP Horizon [h]			
Optimization Method	24	168	720	1440
Smooth-model	**0.1**	**0.6**	**5.1**	**34.9**
Finite-differences: BFGS	0.3	47.6	708.3	x
MCMC	26.4	137.6	571.0	x
P-Swarm 10000	4.2	194.3	1203.6	x
P-Swarm 1000	0.8	18.8	165.6	x
P-Swarm 100	0.2	15.9	x	x
TPE	182.3	x	x	x
Finite-differences: 2^{nd} order	816.6	x	x	x

outperformed by the baseline and the remaining 2 directly fail to find any feasible solution.

- The best alternative methods are MCMC and BFGS, which can find better solutions than the baseline for 3 of the 4 horizons. However, they still fail to find a solution for the longest time horizon and they are outperformed by the proposed approach in terms of both computation time and the quality of the solution.

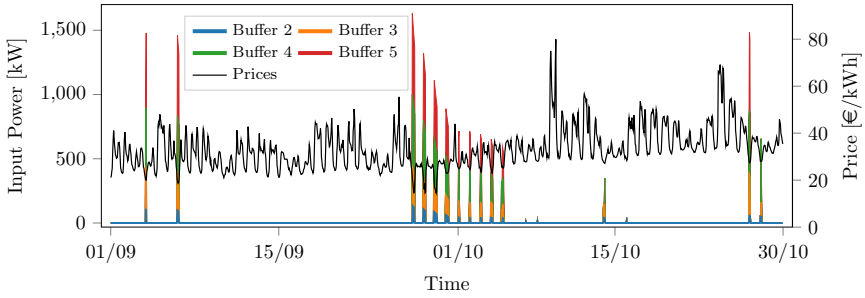
- In terms of computation cost, the method using the smooth model is by far the best: in comparison with all the other alternatives, the method using the smooth model finds the optimal solution between 10 and 100 times faster.

- As expected, as the number of optimization variables increases, all the methods using the non-smooth model struggle to find optimal solutions. In the case of a 1-month horizon, only MCMC and BFGS are able to find a solution. In the case of a 2-months horizon, none of these methods can.

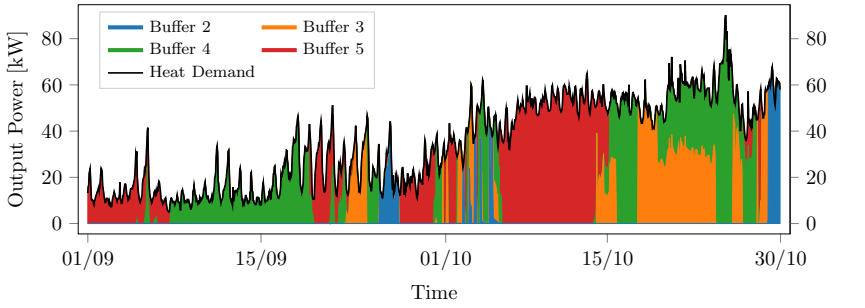
To further illustrate the good results of the smooth model, its optimal solution obtained for the longest horizon is depicted in Figures 7.11 and 7.12. Figure 7.11 illustrates the optimal charging and discharging strategy considering the heat price and the heat demand. Figure 7.12 depicts the optimal temperature evolution of vessel when applying this optimal charging/discharging strategy. As expected, the optimal solution is to fully charge the vessel when prices are lowest, i.e. between the last week of September and the first week of October, and then to discharge the system to follow the heat demand.

7.5.6. DISCUSSION

Based on the obtained results, it is clear that smooth models are very important if heat storage vessels are used in optimization contexts, e.g. if heat storage vessels are to be



(a) Charging strategy



(b) Discharging strategy

Figure 7.11: Optimal charging and discharging of the vessel over the 2-months period. As it is expected, the optimal solution charges the vessel when prices are low and discharges the system to follow the heat demand.

controlled optimally. In particular, when solving the OCP that provides the best charging strategy for the vessel, the smooth model provides the best yet fastest optimal solutions by using derivative-based optimization with automatic differentiation.

This gain becomes more significant for optimization problems with a large number of variables, i.e. for optimization problems that are critical for seasonal storage. In those situations, both heuristic methods and derivative-based optimization methods using finite differences struggle to solve the optimization problem and they use significant amounts of computational resources. Particularly, looking at the case of a 2-months horizon OCP, none of these methods is able to find a feasible solution either because the algorithm converges to an infeasible solution (in the case of heuristic methods) or because the algorithm does not converge within 3 days (the case of finite differences). Similarly, for the 1-month horizon, only MCMC and BFGS can find a solution; however, even in that case, the optimal solutions of MCMC and BFGS are worse and 100 times more computationally expensive than the one provided by the smooth model via derivative-based optimization.

Although they might seem surprising, these results are in fact not unusual as the search space for the heuristic methods becomes very large for the longer horizons: the number of variables to be optimized are 192, 1344, 5760, and 11520 respectively for the 1-day, 1-week, 1-month, and 2-months horizons. In the case of numerical optimization

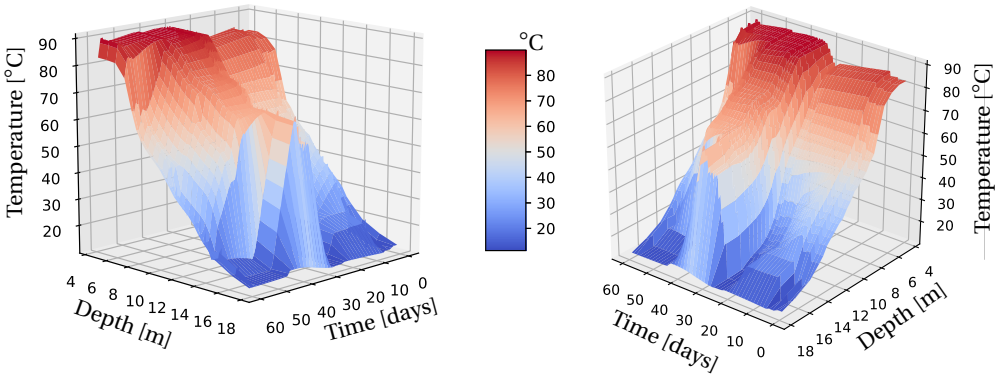


Figure 7.12: Optimal temperature evolution in the vessel over the 2-months period.

with finite differences, a similar problem occurs: the computation time quickly becomes unacceptable. This property can be seen from Table 7.9, which lists the time to compute the Hessian of the Lagrangian for the 4 different horizons and for the first-order and second-order methods: for the longest horizon, the computation cost difference between using finite differences and automatic differentiation is a factor of 10^6 for the second-order method and a factor of 10^3 for the first-order method. Considering that computing the Hessian is done at each iteration of the optimization process, it becomes clear why the finite differences approaches do not scale well with the OCP horizon.

Table 7.9: Comparison of the cost of using automatic differentiation versus using finite differences (via first-order and second-order methods) to compute the Hessian of the Lagrangian in the OCP.

	Computational Cost		
	Finite differences		Automatic
Horizon	Exact Hessian	Hessian-BFGS	Differentiation
24 h	7.4 s[a]	78 ms[a]	1.5 ms
168 h	43 min[a]	5 s[a]	11 ms
720 h	73 h[a,b]	32 s[a]	65 ms
1440 h	405 h[a,b]	130 s[a]	100 ms

[a] To reduce the computational cost of finite differences and list their best case scenario, these costs consider the single shooting formulation where only the control inputs are optimization variables.

[b] This cost is an approximation: it is computed as the cost of evaluating the Lagrangian times $(n+1)^2$, where n is the number of variables in the OCP.

In addition to the quality of the solution and the cheap computational cost, another clear advantage of the smooth model is that it provides the only feasible option to be run in real time, e.g. in a model predictive control setup. In particular, a real-time control application would require computation times below the time step $\Delta t = 1$ h, and as can be seen from Table 7.8, only the proposed model satisfies that.

7.6. CONCLUSIONS

IN this chapter, a new 1-dimensional model for stratified heat storage vessels has been proposed. The model overcomes the shortcomings of the existing models from literature by providing a smooth and continuous 1-dimensional representation of the system dynamics while including buoyancy effects. More specifically, this is the first model that, while remaining 1-dimensional, is able to model buoyancy using a smooth and continuous function. The combination of the smoothness property and the 1-dimensionality of the model is critical to efficiently integrate the model in optimization problems and to obtain better optimal solutions while using less computational resources. These properties allow the use of state-of-the-art derivate-based methods which, in comparison with the optimization methods available for the non-smooth methods from the literature, are computationally much more efficient and lead to more optimal solutions. In addition, the model further innovates the state-of-the-art in the field by explicitly distinguishing between slow and fast buoyancy effects, which results in a more accurate smooth representation of the buoyancy dynamics.

To show the benefits and the accuracy of the model, we have considered a real commercial stratified seasonal storage vessel so that the model is evaluated in a real and noisy environment. In detail, two experiments were carried out. First, the model was validated using real data from the seasonal vessel. During the estimation, the obtained parameters were shown to correctly identify the physical properties behind the system dynamics. Second, the benefits of using the smooth model in optimization problems were demonstrated: the performance of the proposed smooth model was compared against that of non-smooth models from literature considering an optimal control problem where the stratified thermal storage vessel was controlled to minimize its costs while satisfying a given heat demand. In this case study, it was shown that the model did not only result in the best optimal solutions, but it also led to computation costs that were 100 times lower.

In future research, other uses for the proposed model will be explored. A possible line of research will be the inclusion of the model in a model predictive control setup where the heat storage vessel has to interact with multiple markets and systems. In Chapter 8, we explore this line of research and propose multiple control algorithms for electricity trading that make use of this model.

8

CONTROL OF SEASONAL THERMAL ENERGY STORAGE SYSTEMS

It's easy to lie with statistics; it is easier to lie without them.

Frederick Mosteller

What I'd really like to control is not machines, but people.

Stephen Hawking

Seasonal thermal energy storage systems (STESSs) can shift the delivery of renewable energy sources and mitigate their uncertainty problems. However, to maximize the profits of STESSs and to ensure their long-term profitability, control strategies that allow them to trade on wholesale electricity markets are needed. While control strategies for STESSs have been proposed, none of them addressed electricity market interaction and trading. In this chapter, we develop the first control algorithms to control STESSs when interacting with different wholesale electricity markets. As different control solutions have different merits, we propose solutions based on model predictive control (MPC) and solutions based on reinforcement learning (RL). We show that this is critical since different markets require different control strategies: MPC strategies are better for day-ahead markets due to their flexibility whereas RL strategies are better for real-time markets because of fast computation times and better risk modeling. To study the proposed algorithms, we consider a real STESS interacting with the day-ahead and imbalance markets. We show that: (i) the developed controllers successfully maximize the profits of STESSs; (ii) the developed control strategies make STESSs important players in the energy transition: STESSs help to reduce grid imbalances while increasing their profits.

Parts of this chapter have been published in [141, 142].

8.1. INTRODUCTION

As the share of wind and solar energy is expected to reach very high levels by 2030, long-term energy storage [231] is crucial to reduce the seasonal fluctuations of RESs [194], to provide flexibility and ancillary services, and to obtain a smooth and reliable grid operation [194]. In the framework of long-term energy storage, *seasonal thermal energy storage systems (STESSs)* are arguably the most cost-effective technology: they are the most mature technology [194], have the advantage of having a significantly less expensive purchasing cost than electrical energy storage systems [194], and have at their disposal a large energy market that can provide a considerable amount of flexibility.

To guarantee the benefits of STESSs and ensure their widespread use, control strategies that can perform market trading and maximize the operational profit of STESSs are needed. However, despite the importance of such control strategies, existing control approaches are either limited to approaches for short-term storage with market interaction or seasonal storage without market interaction. In detail, while generic methods for storage systems model market interaction, they cannot cope with long optimization horizons. Particularly, all the existing methods [13, 43, 85, 120, 122, 168, 255, 260] provide trading approaches where storage systems trade energy with daily/weekly horizons. This poses a challenge for seasonal storage systems such as STESSs for two reasons:

1. STESSs require forecasts of electricity prices over yearly horizons. Although there are several forecast methods [133, 135, 137] for short-term horizons, i.e. days, there are no reliable forecast methods for long-term horizons.

2. Because of the long optimization horizons, the number of variables in the optimization problems grows very large and the existing methods become computationally intractable.

In terms of control algorithms for seasonal storage, while long horizons are considered, none of the existing methods is able to model electricity market interaction. This interaction is of primary importance for two reasons:

1. To maximize the profit of STESSs, they should be allowed to interact with markets. In particular, although controlling STESSs to satisfy heat demand and/or to maximize renewable energy usage are important objectives, these goals do not necessarily optimize the economic cost of STESSs. This is specially important to increase the number of storage systems in the electrical grid: if the time for return on investment of STESSs is too long, STESSs might become unattractive investments.

2. To help reduce grid imbalances, STESSs need to be able to arbitrage in multiple markets. In particular, to provide up-regulation in the imbalance markets, i.e. a real-time market, STESSs need to first buy that electricity in a market with an earlier gate closure time.

CONTRIBUTIONS AND ORGANIZATION OF THE CHAPTER

To fill this scientific gap, in this chapter we present four distinct contributions:

- First, we propose several control strategies for STESSs to interact with multiple electricity markets. Since there are different markets, we propose control approaches for two cases: trading in the day-ahead market, and simultaneous trading in the day-ahead and imbalance markets. Moreover, as different control approaches have different merits, for each market interaction we propose a *model predictive control (MPC)*-based solution and a *reinforcement learning (RL)*-based controller.
- Second, we propose the first control algorithms for storage systems with long optimization horizons. Unlike the existing literature, the proposed methods quantify the price variations and uncertainty over a horizon of a year, and exploit these variations to maximize its profit.
- Third, we assess the merits of each control solution for the different markets and show that, while MPC-based methods are most suitable for day-ahead markets, RL-based methods perform better when trading in the imbalance market.
- Finally, we empirically demonstrate that STESSs can play an important role in the energy transition by helping grid operators reduce grid imbalances. Particularly, we show that the economic incentives of STESSs are aligned with the regulatory duties of the grid operators.

As minor contribution, to deal with the problem of generating long-term scenarios, we propose a very simple method for generating such scenarios. In addition, considering that there not many papers on imbalance price forecasting, we also propose a forecasting method for imbalance prices.

The remainder of this chapter is organized as follows: Section 8.2 starts by motivating the selected methodology. Section 8.3 introduces and defines the framework of a general STESS interacting with electricity markets. Then, Sections 8.4 and 8.5 respectively present the proposed MPC and RL approaches. Section 8.6 studies the performance of the proposed control approaches under several case studies. Next, Section 8.7 discusses the obtained results. Finally, Section 8.8 concludes the chapter and discusses future research. As supplementary material, Appendix B.1 describes the proposed scenario generation method and Appendix B.2 explains the imbalance price forecasting method.

8.2. MOTIVATION FOR THE SELECTED METHODOLOGY

DESIGNING controllers for trading with STESSs in multiple markets is a challenging task as selecting the right control algorithms or the right electricity markets is not straightforward.

8.2.1. CONTROL ALGORITHMS FOR STESSS WITH MARKET TRADING

Considering the importance of market trading with STESSs, state-of-the-art control approaches, e.g. MPC [195] or RL [223], are highly desirable. However, in the case of MPC [195], several problems appear:

- MPC requires realistic forecasts and/or scenarios of electricity prices over yearly horizons. Although there are several forecast methods [133, 135, 137] and scenario

generation methods [84, 156, 183] for short-term horizons, i.e. days, there are no reliable methods to forecasts or generate scenarios for long-term horizons.

- In real-time electricity markets, e.g. imbalance markets, an action has to be taken within seconds. As the MPC works with a year horizon and the price resolution is typically 15 min, the number of variables in the optimization problem grows very large. As a result, MPC suffers from computational tractability problems to provide the optimal action within the available time frame.

While data-driven and RL techniques can mitigate or solve these two issues, they also have problems of their own:

- Even though they do not require forecasts or scenarios of electricity prices, they need to generate artificial time series of electricity prices to simulate the market conditions. Thus, a method to generate realistic prices is still needed.
- As they are trained offline, they do not have the real-time computation issues of MPC. Nonetheless, that comes at the cost of adaptability: if market conditions change or if the STESS suffers from a problem, e.g. a heat exchanger breaks, the controller has to be re-trained again. As the training can take several days, this limits the adaptability of RL to changes in the environmental conditions. In contrast, as MPC computes the solution online, any change in the environment can be directly included as a change in the optimization problem or by re-estimating the dynamical model with little impact on computation cost.
- The solutions of RL are at best a good approximation of the optimal solution. By contrast, MPC obtains an optimal solution by explicitly solving the problem that we care about.
- Unlike MPC, RL cannot explicitly model hard constraints (they can only be modeled as part of the reward). As such, RL cannot guarantee that the provided solutions do not violate constraints.

Based on these arguments, it becomes clear that the perfect method does not exist and considering RL or MPC involves several trade-offs. As a result, for this research, we will propose different methods based on the two families and analyze the performance of each.

8.2.2. ELECTRICITY MARKETS FOR TRADING WITH STESSS

Besides the challenges that these control strategies pose, another important point to consider is that not all electricity markets are the same. Although STESSs could in theory trade in any electricity market, there are two trading strategies that are especially relevant: trading only in the day-ahead market and trading in both the day-ahead and the imbalance market. Trading only in the day-ahead market is arguably the safest trading strategy for STESSs. In particular, as the day-ahead market is the electricity market with the largest volume of renewable energy trading, i.e. with low but volatile prices, and players incur no risks as they submit bidding curves, this market offers a great opportunity to charge STESSs with low risk.

While trading only in the day-ahead market is a low-risk and cost-effective trading strategy, it might still not be the most optimal economic strategy for STESSs. Although prices in the imbalance market are on average larger than in the day-ahead market, since the imbalance prices are much more volatile, there are periods of time where imbalance prices are much lower (sometimes becoming even negative). In addition, by participating in the imbalance market, STESSs might be able to help reduce grid imbalances: as during periods of positive imbalances, i.e. generation larger than consumption, prices are low, STESSs could wait for these periods to buy their energy; by doing so, they would not only reduce grid imbalances but also increase their own profits. Similarly, as prices are high during periods of negative imbalances, STESSs can make use of their charging flexibility to first buy energy in the day-ahead market, and then sell it in the imbalance market if imbalances are negative or use it if they are positive. By doing so, STESSs could potentially increase their profits whilst helping to reduce negative imbalances.

It is important to note that, despite all these potential benefits, trading strategies for the imbalance market have much higher risks. In the imbalance market, agents take an action for the next time interval without knowing the imbalance price. As imbalance prices are based on the grid imbalances during a period of time, the price is only known after the period is over. Thus, trading strategies for the imbalance market heavily rely on price forecasters and have an associated risk.

With that in mind, it becomes clear that, although trading in both the day-ahead and imbalance markets might potentially be a better strategy than just trading in the day-ahead market, it is necessary to study the viability of this approach and to analyze the newer economic risks. As a result, in this chapter, we will explore both trading strategies and analyze and study the benefits of each.

8.3. SEASONAL STORAGE SYSTEM FRAMEWORK

In order to introduce the control algorithms, we need to define the framework of a general STESS interacting with the electricity markets.

8.3.1. DYNAMICAL MODEL

A STESS can be defined as a general dynamical system with an internal state $\mathbf{x}(t)$, controls $\mathbf{u}(t) = (\dot{\mathbf{Q}}^{\text{in}}(t), \dot{\mathbf{Q}}^{\text{out}}(t))$, n_{units} storage units, and external disturbances $\mathbf{d}(t)$. The internal state $\mathbf{x}(t)$ represents the state of charge of the system and is usually given by the temperature $\mathbf{T}(t)$ in the tank. The controls $\dot{\mathbf{Q}}^{\text{in}}(t) \in \mathbb{R}^{n_{\text{in}}}$ and $\dot{\mathbf{Q}}^{\text{out}}(t) \in \mathbb{R}^{n_{\text{out}}}$ respectively represent the rate at which energy is injected and extracted into/from the system. The disturbance represents any uncontrollable input, e.g. the external temperature.

The dynamics of the system are defined by a *partial differential equation (PDE)*. For a sensible heat storage device with water stratification, the system can be divided into n_{L} layers acting as individual storage units, and the simplified dynamics of a layer i represented by the following PDE:

$$\frac{\partial x_i}{\partial t} = a_1 \frac{\partial^2 x_i}{\partial z^2} + a_2(d_i - x_i) + a_3(\dot{Q}_i^{\text{in}} - \dot{Q}_i^{\text{out}}), \tag{8.1}$$

where z represents the direction of stratification[1].

It is important to note that the definition above is very generic and fits the definition of any STESS system, e.g.:

- A large STESS composed of distributed storage units.
- A system with different charging devices or efficiencies.

As an example, Figure 8.1 illustrates an STESS with two independent storage units.

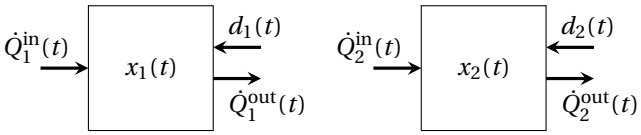


Figure 8.1: Generic representation of an STESS with two distributed storage units.

8.3.2. HEAT DEMAND AND PURCHASED ENERGY

In general, an STESS is required to supply an uncertain heat demand $\dot{Q}^{\text{d}}(t)$. To do so, it buys energy $\dot{Q}^{\text{m}}(t)$ from some market, stores it, and then delivers it to follow $\dot{Q}^{\text{d}}(t)$. To maximize the profits, it needs to consider the price of $\dot{Q}^{\text{m}}(t)$, the storage efficiency, and an estimation of the future heat demand $\dot{Q}^{\text{d}}(t)$. Using these definitions and the notations of the previous section, the following holds:

$$\dot{Q}^{\text{d}}(t) = \sum_{i=1}^{n_{\text{out}}} \dot{Q}_i^{\text{out}}(t), \qquad \dot{Q}^{\text{m}}(t) = \sum_{i=1}^{n_{\text{in}}} \dot{Q}_i^{\text{in}}(t), \tag{8.2}$$

i.e. the heat demand should equal the sum of the energy extracted from the STESS; similarly, the energy bought in the market should equal to sum of the energy introduced in the STESS.

It is important to note that an STESS works with periodic seasonal cycles where the heat demand $\dot{Q}^{\text{d}}(t)$ follows a similar pattern in each cycle. Usually, these seasonal cycles last a year and the demand is characterized by the season/time of the year.

8.3.3. TRADING IN THE DAY-AHEAD MARKET

Given a day-ahead market with unknown daily hourly prices $(p_1^{\text{dam}}, \ldots, \ldots p_{24}^{\text{dam}})$, the goal of any control algorithm for an STESS is to build optimal bidding curves to maximize the profit. In particular, the aim is to, one day in advance, build 24 optimal bidding curves $\dot{Q}_1^{\text{b}}(\cdot), \ldots, \dot{Q}_{24}^{\text{b}}(\cdot)$ such that, while the STESS always has enough energy to satisfy the demand $\dot{Q}^{\text{d}}(t)$, the cost of the purchased power $\dot{Q}^{\text{dam}}(t)$ is minimized. In this market structure, the purchased power $\dot{Q}^{\text{dam}}(t)$ at every hour h is defined by:

$$\dot{Q}^{\text{dam}}(t) = \dot{Q}_h^{\text{b}}(p_h^{\text{dam}}), \quad \forall\, t \in [h, h+1). \tag{8.3}$$

with p_h^{dam} being the market cleared price during hour h.

[1] Note that this equation does not model the buoyancy in the STESS. As explained in Chapter 7, the buoyancy can be modeled by adding nonlinear terms to the integral of the PDE.

8.3.4. TRADING IN THE IMBALANCE MARKET

For the imbalance market, the imbalance price p^{imb} is always unknown when purchasing/selling power as the price p^{imb} is determined in real time based on the costs of reserves activated by the TSO. In particular, at time step k, a market agent has to decide whether to sell, buy, or not trade without knowing the imbalance price p_k^{imb} for the interval $[k, k+1)$. As p_k^{imb} is usually known immediately at the next interval, the agent can take the decision based on past imbalance prices $p_{k-1}^{\text{imb}}, p_{k-2}^{\text{imb}}, \ldots$ or any other information available at time step $k-1$.

Defining as $\dot{Q}^{\text{imb}}(t)$ the energy traded in the imbalance market, with positive and negative values respectively representing energy that is bought and sold, it holds that:

$$-\dot{Q}^{\text{imb}}(t) \leq \dot{Q}^{\text{dam}}(t), \tag{8.4}$$

i.e. the energy sold in the imbalance market by an STESS is limited by the energy purchased on any previous market (the day-ahead market in the case of the proposed control algorithms). Particularly, because the STESS cannot effectively convert heat back to electricity, any energy sold is limited by the energy bought for the same time period in other markets. Similarly, it holds that:

$$\dot{Q}^{\text{m}}(t) = \dot{Q}^{\text{dam}}(t) + \dot{Q}^{\text{imb}}(t), \tag{8.5}$$

i.e. the total energy purchased for the STESS is the sum of the energy purchased in the day-ahead and imbalance markets.

Considering these definitions, a control algorithm for the imbalance market has to select the value of $\dot{Q}^{\text{imb}}(t)$ for each time step k so that, while the STESS has enough energy to to satisfy the demand $\dot{Q}^{\text{d}}(t)$, the total cost of trading $\dot{Q}^{\text{dam}}(t)$ and $\dot{Q}^{\text{imb}}(t)$ is minimized. To do so, the control algorithm receives as an input the energy $\dot{Q}^{\text{dam}}(t)$ purchased in the day-ahead, and selects the value of $\dot{Q}^{\text{imb}}(t)$.

8.4. MPC APPROACHES

IN this section, we derive and explain the two proposed MPC approaches: one to trade exclusively on the day-ahead market, and a second one to trade on both the day-ahead and the imbalance market.

8.4.1. BIDDING FUNCTIONS

In the case of the day-ahead electricity market, the goal of the MPC is to provide the 24 optimal bidding functions $\dot{Q}_h^{\text{b}}(\cdot)$, for $h = 1, 2, \ldots, 24$. Since standard MPC can only provide the optimal market power $\dot{Q}_{\bar{p}}^{\text{dam}}$ for a fixed price $\bar{p}$, an additional step is needed. For each hour h, we propose the following approach:

1. Predefine n_{p} discrete prices $\{p^1, p^2, \ldots, p^{n_{\text{p}}}\}$ for the price p^{dam} at hour h.
2. Fix the remaining 23 day-ahead prices using their expected value, e.g. a forecast $\{\hat{p}_j^{\text{dam}} \mid j = 1, 2 \ldots, 24,\ i \neq h\}$.
3. Solve the MPC for each of these n_{p} prices and obtain the associated optimal market powers $\{\dot{Q}_{p^1}^{\text{dam}}, \dot{Q}_{p^2}^{\text{dam}}, \ldots, \dot{Q}_{p^{n_{\text{p}}}}^{\text{dam}}\}$ at hour h.

4. Build the bidding function as a piecewise constant function based on the obtained solutions:

$$\dot{Q}_h^{\mathrm{b}}(p^{\mathrm{dam}}) = \begin{cases} \dot{Q}_{p^1}^{\mathrm{dam}}, & p^{\mathrm{dam}} \leq p^1 \\ \dot{Q}_{p^2}^{\mathrm{dam}}, & p^1 < p^{\mathrm{dam}} \leq p^2 \\ \vdots & \\ \dot{Q}_{p^{n_{\mathrm{p}}}}^{\mathrm{dam}}, & p^{n_{\mathrm{p}}-1} < p^{\mathrm{dam}} \leq p^{n_{\mathrm{p}}} \\ 0, & p^{n_{\mathrm{p}}} < p^{\mathrm{dam}} \end{cases} \tag{8.6}$$

This approach for building bidding functions is obviously only possible as long as the bidding functions within one day are independent of each other. However, since STESSs are very large storage devices, their internal state does not vary much within one day. As a result, the choice of one bidding function does not affect much the others and the assumption of independent bidding functions holds in practice.

Moreover, due to the day-ahead market structure and the long optimization horizons of STESS, the 24 bidding functions are very similar. In detail, as the 24 daily bids are submitted at the same time, all the bids are built based on the same information, e.g. the STESS state. In addition, because the state of the STESS barely changes with the action taken in a given hour and the optimal bidding functions only depend on the STESS state, it follows that the optimal bidding function for every hour of a given day are similar. As a result, in a given day, the difference in price distribution between hours is not important, and the STESS reacts almost equally to the market price independently of the hour, i.e.:

$$\dot{Q}_1^{\mathrm{b}}(p^{\mathrm{dam}}) \approx \dot{Q}_2^{\mathrm{b}}(p^{\mathrm{dam}}) \approx \ldots \approx \dot{Q}_{24}^{\mathrm{b}}(p^{\mathrm{dam}}), \quad \forall\ p^{\mathrm{dam}}. \tag{8.7}$$

Thus, to build the 24 bidding functions, it is only needed to obtain the bidding function $\dot{Q}_1^{\mathrm{b}}(\cdot)$ for the first hour.

8

8.4.2. MPC FOR DAY-AHEAD TRADING

As motivated in the previous section, we only need to estimate the bidding function $\dot{Q}_1^{\mathrm{b}}(\cdot)$ for the first hour of the day. However, instead of solving a single *optimal control problem (OCP)* like in standard MPC, we need to discretize the first price p_1^{dam} into a discrete set of prices $\{p^1, p^2, \ldots, p^{n_{\mathrm{p}}}\}$, and for each of these prices solve the relevant OCP.

For the sake of simplicity, in this section we will assume that each OCP is optimized using a discrete time grid $t_1, t_2, \ldots, t_{N+1}$, i.e. using an optimization horizon equal to $t_{N+1} - t_1$; the details of how the time grid is defined will be covered in Section 8.4.4. Similarly, we will assume that the expected day-ahead prices $\{\hat{p}_k^{\mathrm{dam}}\}_{k=1}^N$, the expected heat demand values $\{\hat{\dot{Q}}_k^{\mathrm{d}}\}_{k=1}^N$, and the expected disturbances $\{\hat{\mathbf{d}}_k\}_{k=1}^N$ are also provided; the method to obtain these values are explained in Appendix B.1.

Considering the previous definitions, at every day and for each price p^j, the MPC approach solves the following OCP:

OCP(p^j):

$$\underset{\substack{\mathbf{x}_1,\dot{\mathbf{Q}}_1^{\text{in}},\dot{\mathbf{Q}}_1^{\text{out}},\dot{Q}_1^{\text{dam}},\mathbf{x}_2,\ldots,\\ \dot{\mathbf{Q}}_N^{\text{in}},\dot{\mathbf{Q}}_N^{\text{out}},\dot{Q}_N^{\text{dam}},\mathbf{x}_{N+1}}}{\text{minimize}} \quad p^j \dot{Q}_1^{\text{dam}} \Delta t_1 + \sum_{k=2}^{N} \hat{p}_k^{\text{dam}} \dot{Q}_k^{\text{dam}} \Delta t_k \tag{8.8a}$$

$$\text{subject to} \quad \mathbf{x}_1 = \bar{\mathbf{x}}_1, \tag{8.8b}$$

$$\mathbf{x}_{k+1} = \mathbf{G}(\mathbf{x}_k, \dot{\mathbf{Q}}_k^{\text{in}}, \dot{\mathbf{Q}}_k^{\text{out}}, \hat{\mathbf{d}}_k, \Delta t_k), \quad \text{for } k = 1,\ldots N, \tag{8.8c}$$

$$\mathbf{x}_{\min} \le \mathbf{x}_k \le \mathbf{x}_{\max} \quad \text{for } k = 1,\ldots N, \tag{8.8d}$$

$$\sum_{i=1}^{n_{\text{out}}} \dot{Q}_{k,i}^{\text{out}} = \hat{\dot{Q}}_k^{\text{d}}, \quad \text{for } k = 1,\ldots N, \tag{8.8e}$$

$$0 \le \dot{\mathbf{Q}}_k^{\text{in}} \le \mathbf{g}_{\text{in}}(\mathbf{x}_k), \quad \text{for } k = 1,\ldots N, \tag{8.8f}$$

$$0 \le \dot{\mathbf{Q}}_k^{\text{out}} \le \mathbf{g}_{\text{out}}(\mathbf{x}_k), \quad \text{for } k = 1,\ldots N, \tag{8.8g}$$

$$\dot{Q}_k^{\text{dam}} \le \dot{Q}_{\max}^{\text{m}}, \quad \text{for } k = 1,\ldots N, \tag{8.8h}$$

$$\sum_{i=1}^{n_{\text{in}}} \dot{Q}_{k,i}^{\text{in}} = \dot{Q}_k^{\text{dam}}, \quad \text{for } k = 1,\ldots N, \tag{8.8i}$$

$$\mathbf{x}_{N+1} = \bar{\mathbf{x}}_1, \tag{8.8j}$$

where:

- The time step at every time point k is defined by $\Delta t_k = t_{k+1} - t_k$.
- The objective function represents the cost of purchasing energy considering that the first price is fixed and given by p^j and that the remaining prices in the horizon are the expected prices in the market.
- Equation (8.8b) fixes the initial state, which is assumed to be known and given by $\bar{\mathbf{x}}_1$.
- Equation (8.8c) ensures that the dynamics of the system are ensured at every time step. This equation is obtained by discretizing the continuous PDE (8.1) as explained in Chapter 7.
- The limits on the STESS state are defined by (8.8d).
- Through (8.8e) it is ensured that the heat demand is met.
- Equations (8.8f) and (8.8g) ensure the individual charging and discharging limits of each individual storage device. The upper limit is usually a function of the state as the maximum power that can be charged/discharged usually depends on the state of charge. An example of this dependency is provided by (7.25f–7.25g) in the case study of Chapter 7.
- The maximum power purchased from the market is limited by (8.8h). This can represent the maximum power of the electrical installation, i.e. maximum power of all charging devices, or a maximum amount to be purchased on the market based on some risk metric.

- Equation (8.8i) ensures that the input power equals the power purchased from the market.

- The OCP should avoid depleting the STESS at the end of the horizon. To do so, as the optimization horizon is usually a seasonal periodic cycle (see Section 8.4.4 for details), (8.8j) constrains the STESS to have the same state of charge at the beginning and at the end.

- This OCP is similar to the OCP (7.25) defined to evaluate the model proposed in Chapter 7 but with four differences: (i) avoiding depletion is modeled with the constraint (8.8j) instead of a penalty in the cost function; (ii) the maximum power from the market is modeled with an extra variable $\dot{Q}_k^{\text{dam}}$ instead of the sum of input power $\sum_{i=1}^{n_{\text{in}}} \dot{Q}_{k,i}^{\text{in}}$; (iii) the maximum power from the market is now limited by (8.8h); (iv) for the sake of generality, the state is defined by a general variable $\mathbf{x}$ instead of the temperature $\mathbf{T}$ and the dynamics include disturbances $\mathbf{d}$.

After solving an OCP for each discrete price p^{j}, the optimal bidding function $\dot{Q}_1^{\text{b}}(\cdot)$ can be estimated using (8.6), where the optimal market power $\dot{Q}_{p^{\text{j}}}^{\text{dam}}$ equals $\dot{Q}_1^{\text{dam}}$.

8.4.3. MPC FOR DAY-AHEAD AND IMBALANCE TRADING

The MPC-based approach to trade in both the day-ahead and the imbalance market consists of two separate MPC algorithms that run one after the other:

- A first MPC algorithm that trades in the day-ahead market but, unlike the MPC algorithm defined in the previous section, it considers that there is also a possible future interaction with the imbalance market.

- A second MPC algorithm that trades in the imbalance market and that considers that there is also possible future interactions with the day-ahead market. Nevertheless, unlike the MPC algorithm for the day-ahead market, it runs in real time and it does not build bidding functions. Instead, at time step $k-1$, it considers a forecast $\hat{p}_k^{\text{imb,acc}}$ of the next imbalance price and then solves a single OCP to obtain the optimal power $\dot{Q}_k^{\text{imb}}$ to trade in the imbalance market.

MPC FOR THE DAY-AHEAD MARKET

To define the OCP of the first MPC algorithm, we will again consider that the discrete time grid $t_1, t_2, \ldots, t_{N+1}$, the expected day-ahead prices $\{\hat{p}_k^{\text{dam}}\}_{k=1}^N$, imbalance prices $\{\hat{p}_k^{\text{imb}}\}_{k=1}^N$, heat demand values $\{\hat{\dot{Q}}_k^{\text{d}}\}_{k=1}^N$, and disturbances $\{\hat{\mathbf{d}}_k\}_{k=1}^N$ are given. In addition, we will simplify the vector of input controls as $\mathbf{u}_k = (\dot{\mathbf{Q}}_k^{\text{in}}, \dot{\mathbf{Q}}_k^{\text{out}}, \dot{Q}_k^{\text{dam}}, \dot{Q}_k^{\text{imb}})$. Then, at every day and for each discrete price in $\{p^1, p^2, \ldots, p^{n_{\text{p}}}\}$, the MPC solves the following OCP:

OCP(p^j):

$$\underset{\substack{\mathbf{x}_1,\mathbf{u}_1,\mathbf{x}_2,\ldots,\\ \mathbf{u}_N,\mathbf{x}_{N+1}}}{\text{minimize}} \quad p^j\,\dot{Q}_1^{\text{dam}}\,\Delta t_1 + \sum_{k=2}^{N} \hat{p}_k^{\text{dam}}\,\dot{Q}_k^{\text{dam}}\,\Delta t_k + \sum_{k=1}^{N} \hat{p}_k^{\text{imb}}\,\dot{Q}_k^{\text{imb}}\,\Delta t_k \tag{8.9a}$$

$$\text{subject to} \quad \mathbf{x}_1 = \bar{\mathbf{x}}_1, \tag{8.9b}$$

$$\mathbf{x}_{k+1} = \mathbf{G}(\mathbf{x}_k, \dot{\mathbf{Q}}_k^{\text{in}}, \dot{\mathbf{Q}}_k^{\text{out}}, \hat{\mathbf{d}}_k, \Delta t_k), \quad \text{for } k = 1,\ldots N, \tag{8.9c}$$

$$\mathbf{x}_{\min} \le \mathbf{x}_k \le \mathbf{x}_{\max} \quad \text{for } k = 1,\ldots N, \tag{8.9d}$$

$$\sum_{i=1}^{n_{\text{out}}} \dot{Q}_{k,i}^{\text{out}} = \hat{\dot{Q}}_k^{\text{d}}, \quad \text{for } k = 1,\ldots N, \tag{8.9e}$$

$$0 \le \dot{\mathbf{Q}}_k^{\text{in}} \le \mathbf{g}_{\text{in}}(\mathbf{x}_k), \quad \text{for } k = 1,\ldots N, \tag{8.9f}$$

$$0 \le \dot{\mathbf{Q}}_k^{\text{out}} \le \mathbf{g}_{\text{out}}(\mathbf{x}_k), \quad \text{for } k = 1,\ldots N, \tag{8.9g}$$

$$\dot{Q}_k^{\text{dam}} + \dot{Q}_k^{\text{imb}} \le \dot{Q}_{\max}^{\text{m}}, \quad \text{for } k = 1,\ldots N, \tag{8.9h}$$

$$\sum_{i=1}^{n_{\text{in}}} \dot{Q}_{k,i}^{\text{in}} = \dot{Q}_k^{\text{dam}} + \dot{Q}_k^{\text{imb}} \quad \text{for } k = 1,\ldots N, \tag{8.9i}$$

$$\dot{Q}_k^{\text{dam}} \ge 0, \quad \text{for } k = 1,\ldots N, \tag{8.9j}$$

$$-\dot{Q}_k^{\text{dam}} \le \dot{Q}_k^{\text{imb}}, \quad \text{for } k = 1,\ldots N, \tag{8.9k}$$

$$\mathbf{x}_{N+1} = \bar{\mathbf{x}}_1. \tag{8.9l}$$

Although the main structure is very similar to (8.8), there are some important differences:

- The algorithm minimizes the cost of purchasing energy like in (8.8a) but includes the future transactions in the imbalance market.
- The constraints that contain the power purchased from the market, i.e. (8.9h) and (8.9i), consider now the sum of the power purchased in both markets.
- Unlike the case of trading only in the day-ahead market, the STESS can now sell energy on the imbalance market if it has previously bought it in the day-ahead market. This is modeled by (8.9j) and (8.9k), which respectively guarantee that in the day-ahead market energy can only be bought, and that the energy sold in the imbalance market is limited to the energy bought in the day-ahead market.

MPC FOR THE IMBALANCE MARKET

To define the second MPC algorithm, let us first make the following assumptions and definitions:

- The MPC algorithm for the imbalance market considers a new time grid $t'_1, t'_2, \ldots, t'_{N_1+1}$ with $t'_{N_1+1} \le t_{N+1}$, i.e. a shorter horizon and a different discretization. The details on this discretization are provided in Section 8.4.4.
- The optimal state at time t_{N_1+1} is defined by $\mathbf{x}^\star_{N_1+1}$ and obtained from the solution of the MPC for the day-ahead market. In particular, this value can be obtained from the optimal solution of any of the n_{p} OCPs solved in the latest day-ahead market.

- An accurate forecast $\hat{p}_1^{\text{imb,acc}}$ of the next price in the imbalance market is available. The details of this forecast are explained in Appendix B.2.
- The time index t'_{n_d} defines the last time index of the first day, e.g. if $t'_1 = 21:00\,\text{h}$ and $\Delta t'_k = 1\,\text{h}$ then $t'_{n_\text{d}} = t'_3 = 23:00\,\text{h}$.
- As the day-ahead market has already been cleared for the first day, i.e. for time indices $t'_1, \dots, t'_{n_\text{d}}$, the day-ahead prices $\{p_k^{\text{dam}}\}_{k=1}^{n_\text{d}}$ and allocated day-ahead power $\{\dot{Q}_k^{\text{dam}}\}_{k=1}^{n_\text{d}}$ are predefined. As a result, the vector of input controls $\mathbf{u}_k$ is defined as $(\dot{\mathbf{Q}}_k^{\text{in}}, \dot{\mathbf{Q}}_k^{\text{out}}, \dot{Q}_k^{\text{imb}})$ for $k = 1, \dots, n_\text{d}$ and as $(\dot{\mathbf{Q}}_k^{\text{in}}, \dot{\mathbf{Q}}_k^{\text{out}}, \dot{Q}_k^{\text{dam}}, \dot{Q}_k^{\text{imb}})$ for $k = n_\text{d}+1, \dots, N_1$.
- The expected day-ahead prices $\{\hat{p}_k^{\text{dam}}\}_{k=n_\text{d}+1}^{N_1}$, imbalance prices $\{\hat{p}_k^{\text{imb}}\}_{k=1}^{N_1}$, heat demand values $\{\hat{\dot{Q}}_k^{\text{d}}\}_{k=1}^{N_1}$, and disturbances $\{\hat{\mathbf{d}}_k\}_{k=1}^{N_1}$ are again provided. (See Appendix B.1 for details).

Based on these definitions, before each imbalance market clearance, MPC solves the following OCP and trades the optimal solution $\dot{Q}_1^{\text{imb}}$ in the imbalance market:

$$\underset{\substack{\mathbf{x}_1,\mathbf{u}_1,\mathbf{x}_2,\dots,\\ \mathbf{u}_{N_1},\mathbf{x}_{N_1+1}}}{\text{minimize}} \quad \hat{p}_1^{\text{imb,acc}}\,\dot{Q}_1^{\text{imb}}\,\Delta t'_1 + \sum_{k=2}^{N_1} \hat{p}_k^{\text{imb}}\,\dot{Q}_k^{\text{imb}}\,\Delta t'_k + \sum_{k=n_\text{d}+1}^{N_1} \hat{p}_k^{\text{dam}}\,\dot{Q}_k^{\text{dam}}\,\Delta t'_k \tag{8.10a}$$

$$\text{subject to} \quad (8.9b) - (8.9k), \tag{8.10b}$$

$$\mathbf{x}_{N_1+1} = \mathbf{x}^\star_{N_1+1}. \tag{8.10c}$$

The new MPC scheme is very similar to the previous MPC for the day-ahead market but with some differences:

- The time step at every time point k is now defined by $\Delta t'_k = t'_{k+1} - t'_k$.
- As a bidding function is not needed, instead of solving the OCP multiple times for different possible prices, this MPC algorithm solves a single OCP considering the most likely imbalance price $\hat{p}_1^{\text{imb,acc}}$ in the next market clearance. Then, it trades directly the optimal solution $\dot{Q}_1^{\text{imb}}$ in the imbalance market.
- A distinction is made between the future expected imbalance prices $\{\hat{p}_k^{\text{imb}}\}_{k=2}^{N_1}$ and the forecast price $\hat{p}_1^{\text{imb,acc}}$ in the next time step. This distinction is made because the accuracy of the forecast is better than that of the method used to generate the expected future values[2].
- The cost for the day-ahead market starts at index $n_\text{d}+1$, which represents the first time step of the second day. In particular, as the cost $\sum_{k=1}^{n_\text{d}} p_k^{\text{dam}}\,\dot{Q}_k^{\text{dam}}\,\Delta t'_k$ incurred in the day-ahead market during the first day has already been determined, it is not included in the objective function as it represents a constant term.

[2] The difference between $\hat{p}_1^{\text{imb,acc}}$ and $\hat{p}_1^{\text{imb}}$ is that the latter represents a general estimation of the expected imbalance price and the former is a forecast using the most recent available information. Moreover, $\hat{p}_1^{\text{imb,acc}}$ is (in general) more accurate and is based on a point forecast; by contrast, $\hat{p}_1^{\text{imb}}$ is not res be estimated using other methods.

- As this MPC algorithm runs in real time, the computation time should be as small as possible. To reduce the computation time, a smaller horizon $t'_{N_1+1} < t_{N+1}$ is considered.
- As the optimization horizon t'_{N_1+1} is now smaller than a periodical seasonal cycle, it is suboptimal to constrain the final state to be equal to the initial state. However, not constraining the final state leads to an OCP that does not account for what happens after t'_{N_1+1}. To solve this problem, (8.10c) constrains the final state to be equal to the optimal state $\mathbf{x}^\star_{N_1+1}$ at time t'_{N_1+1}, which is obtained from the solution of the latest day-ahead MPC algorithm.

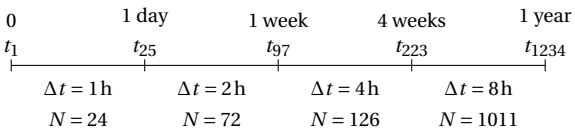
8.4.4. TIME GRID AND OPTIMIZATION HORIZON

In the previous sections, we assumed that the discrete time grids where the OCPs were defined were given. In this section, we explain the methodology to define these timegrids. In general, to define a discrete-time grid, we also need to define the optimization horizon T and the discrete time step Δt; then, based on T and Δt, the number of time intervals N is defined. For an STESS, T represents its seasonal horizon, which is typically a year. While most applications consider a constant Δt along the optimization horizon, we argue that for an STESS this not necessary and should in fact be avoided:

- As day-ahead markets have a different price every hour, the largest time step at the beginning of the horizon is limited to $\Delta t = 1\,\text{h}$. However, due to the long optimization horizons, it is not possible to accurately estimate with hourly resolutions the price and demand distributions at the end of the horizon. Instead, at the end of the horizon, it is better to estimate the distributions over larger intervals, e.g. several hours, where due to noise averaging the uncertainty can be better quantified.
- Another reason to consider a variable Δt is the computational cost: by increasing Δt towards the end of the horizon, we reduce the number of optimization points N and the computational complexity of the OCP.
- As MPC only needs the optimal control at the first time point, it can be argued that lowering the time resolution at the end of the horizon has little impact on the first optimal control.

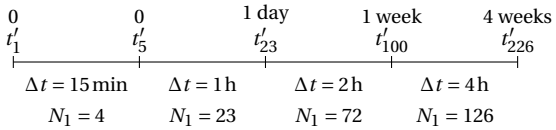
DAY-HEAD MARKET

Considering that the day-ahead electricity market is cleared every day, the hourly resolution should only be needed for the first day. Based on this and the arguments above, for the day-ahead market MPC we consider a time grid $t_1, t_2, \ldots, t_{N+1}$ with a year horizon, using four different Δt, and containing $N = 1233$ time intervals:



IMBALANCE MARKET

For the case of the imbalance market, the minimum Δt is 15 minutes. Moreover, considering the large uncertainty in imbalances prices, we argue that the 15 min resolution is only needed for the first hour. Finally, as the MPC algorithm for the imbalance market runs in real time, the computation time should be as small as possible. Based on these arguments, we consider a time grid $t'_1, t'_2, \ldots, t'_{N_1+1}$ for the imbalance market with a horizon of four weeks, using four different Δt, and containing $N_1 = 225$ time intervals:



It could be argued that considering a horizon of four weeks instead of a year (the standard seasonal cycle) leads to suboptimal solutions, i.e. the MPC cannot account for what happens during a full seasonal cycle. As explained in Section 8.4.3, MPC avoids this by constraining the state at the end of the four weeks to be equal to the optimal state $\mathbf{x}^\star_{226}$ at that time point. As the MPC algorithm for the day-ahead market always runs before the MPC algorithm for the imbalance market, $\mathbf{x}^\star_{226}$ is easily obtained from the day-ahead MPC optimal solution.

8.5. RL APPROACHES

IN this section, we present the two proposed RL approaches: one to trade in the day-ahead market, and a second one to trade in both the day-ahead and the imbalance markets.

8.5.1. RL FOR DAY-AHEAD TRADING

As with MPC, any RL control algorithm for the day-ahead market needs to estimate the bidding functions $\dot{Q}^{\mathrm{b}}_h(\cdot)$, for $h = 1, 2, \ldots, 24$. While in the case of MPC that required discretizing prices and solving multiple OCPs, for RL the bidding functions can be directly obtained from the optimal policy $\pi^\star(\mathbf{s}_k)$. In detail, if the RL agent is set up so that:

- The reward represents the negative of the cost of purchasing energy.
- The RL state $\mathbf{s}$ contains the day-ahead price p^{dam}.
- The action $\mathbf{u}$ includes the power $\dot{Q}^{\mathrm{dam}}$ purchased from the market.

Then, by definition, the bidding function $\dot{Q}^{\mathrm{b}}(p^{\mathrm{dam}})$ is implicitly defined by the optimal policy $\mathbf{u}^\star = \pi^\star(\mathbf{s}) = \pi^\star(p^{\mathrm{dam}}, \ldots)$. Below we provide further details on the proposed RL algorithm.

STATE AND CONTROL SPACES

The first step to define the RL algorithm is to define its state and control spaces. For the proposed algorithm, the state $\mathbf{s} = (\mathbf{x}, \tau, p^{\mathrm{dam}})$ is defined by three different features:

1. The state $\mathbf{x}$ of the STESS.
2. The time position τ within the periodic seasonal cycle, e.g. the day of the year.
3. The market price p^{dam}.

The reason for selecting these three features is twofold:

- The optimal action $\mathbf{u}^\star = \pi^\star(\mathbf{s})$ can be selected based on both the state of the STESS and the environment.
- As we will show, given a fixed time point $\bar{\tau}$ and STESS state $\bar{\mathbf{x}}$, the bidding function $\dot{Q}^{\text{b}}(p^{\text{dam}})$ is by definition given by the optimal policy $\bar{\pi}^\star(\bar{\mathbf{x}}, \bar{\tau}, p^{\text{dam}})$.

To define the action space $\mathcal{U}$, we consider that a single action $u \in \mathbb{R}^{n_{\text{in}}+1}$ has the following format:

$$\mathbf{u} = (u_1, u_2, \dots, u_{n_{\text{in}}}, j). \tag{8.11}$$

In detail, we consider that each input control u_i can take $n_{\text{dis}} + 1$ discrete values uniformly separated between 0 and 1 and that the real power $\dot{Q}_i^{\text{in}}$ into the storage i is obtained by multiplying u_i by the maximum power $\dot{Q}_i^{\max}$, i.e. $\dot{Q}_i^{\text{in}} = u_i \, \dot{Q}_i^{\max}$. This scaling is done because $\dot{Q}_i^{\max}$ might depend on the system state and can change throughout time. For the output control, a single storage unit j is selected to provide the demand $\dot{Q}^{\text{d}}$, i.e. $\dot{Q}^{\text{d}} = \dot{Q}_j^{\text{out}}$. The action space is then defined by the possible combinations of all these values:

$$\begin{aligned}\mathcal{U} = \big\{&(0,\dots,0,1), (\tfrac{1}{n_{\text{dis}}}\dot{Q}_1^{\max},0,\dots,0,1), (\tfrac{2}{n_{\text{dis}}}\dot{Q}_1^{\max},0,\dots,0,1), \dots, (\dot{Q}_1^{\max},\dot{Q}_2^{\max},\dots,\dot{Q}_{n_{\text{in}}}^{\max},1),\\ &(0,\dots,0,2), \dots, (\dot{Q}_1^{\max},\dot{Q}_2^{\max},\dots,\dot{Q}_{n_{\text{in}}}^{\max},2),\dots,\\ &(0,\dots,0,n_{\text{dis}}), \dots, (\dot{Q}_1^{\max},\dot{Q}_2^{\max},\dots,\tfrac{n_{\text{dis}}-1}{n_{\text{dis}}}\dot{Q}_{n_{\text{in}}}^{\max},n_{\text{out}}), (\dot{Q}_1^{\max},\dot{Q}_2^{\max},\dots,\dot{Q}_{n_{\text{in}}}^{\max},n_{\text{out}})\big\}.\end{aligned} \tag{8.12}$$

8

REWARD FUNCTION

The reward r_k at time step k is defined as the negative of the cost of the energy purchased. Thus, assuming that the agent is at state $\mathbf{s}_k = (\mathbf{x}_k, \tau_k, p_k^{\text{dam}})$ and takes an action $\mathbf{u}_k = (u_{1,k}, \dots, u_{n_{\text{in}},k}, j)$, r_k is defined as $-p_k^{\text{dam}} \sum_{i=1}^{n_{\text{in}}} (u_{i,k} \cdot \dot{Q}_{i,k}^{\max})$. In addition, if the agent depletes the system and the demand $\dot{Q}_k^{\text{d}}$ cannot be satisfied, the reward penalizes this situation with a cost 10 times larger than the cost of instantaneously buying $\dot{Q}_k^{\text{d}}$ in the market[3]. Finally, as with standard RL algorithms, the reward at the last point in an episode is 0:

$$r_k = \begin{cases} 0, & \text{if } k = T_{\text{e}} \\ -p_k^{\text{dam}}\left(\sum_{i=1}^{n_{\text{in}}} (u_{i,k}\, \dot{Q}_{i,k}^{\max}) + 10\, \dot{Q}_k^{\text{d}}\right), & \text{if the system is depleted} \\ -p_k^{\text{dam}} \sum_{i=1}^{n_{\text{in}}} (u_{i,k} \cdot \dot{Q}_{i,k}^{\max}), & \text{otherwise.} \end{cases} \tag{8.13}$$

[3] Selecting a factor of 10 is a design choice. The agent just needs a large penalty cost whenever it depletes the STESS.

EPISODE LENGTH AND TIME GRID

Another critical point when designing an RL algorithm is to select the episode length T_e. For an STESS, it can be argued that, to avoid optimal policies that deplete the STESS, the minimum T_e should be two seasonal periodic cycles. In particular, if the episode length equals the cycle length, the agent would know the time position within an episode as the agent knows the time position τ within a seasonal cycle. Using that information, the agent could potentially deplete the STESS at the end of the episode/cycle to reduce the cost. This behavior would be undesirable as the STESS needs to provide energy for more than a seasonal periodical cycle.

For the size of the discrete time grid, we consider that a time step transition $k \to k+1$ spans a day. In particular, as with MPC, it is assumed that the state of charge does not change dramatically from one day to another and that the optimal bidding curves within a day are very similar. It is important to note that selecting this time step size is just a design choice and that it is equally possible to consider time steps of one hour at the expense of increasing the computation load.

SIMULATION ENVIRONMENT

During training, the RL agent needs to interact with the system and environment. However, unless interacting with the real ones is cheap and risk free, building a simulator is usually needed. For STESSs, considering the cost of trading, we argue that a simulation environment is the best solution and we propose a simulation environment that recreates the world an STESS lives in. In detail, this environment consists of two modules:

- **STESS simulator**: An environment that simulates the dynamical model of the STESS, i.e. $\mathbf{x}_{k+1} = \mathbf{G}(\mathbf{x}_k, \dot{\mathbf{Q}}_k^{\text{in}}, \dot{\mathbf{Q}}_k^{\text{out}}, \hat{\mathbf{d}}_k)$.
- **Environment simulator**: A simulator that produces realistic day-ahead market prices p^{dam}, heat demand $\dot{Q}^{\text{d}}$, and disturbances $\mathbf{d}$. To obtain a simulator that generates realistic time series, the method for scenario generation explained in Appendix B.1 is considered.

TRAINING ALGORITHM

The last step before training the agent is to select the specific RL algorithm to estimate the optimal policy $\pi^\star(\mathbf{s})$. For the case of STESSs, we propose using fitted Q-iteration [73] with boosting trees [42]. The reason for selecting this algorithm is that we empirically observed (using the real system presented in Section 8.6) that this algorithm performed as good as more advanced RL algorithms but without the additional computational complexity. For more details about the algorithm see Section 6.4.2, [223], or [73].

BUILDING BIDDING FUNCTIONS

After the RL agent is trained, the optimal bidding functions $\dot{Q}^{\text{b}}(\cdot)$, which define the power purchased from the market as a function of the price p^{dam}, are directly obtained. In particular, given a fixed time point $\bar{\tau}$ and STESS state $\bar{\mathbf{x}}$, we have an optimal policy $\mathbf{u}^\star = \pi^\star(\bar{\mathbf{x}}, \bar{\tau}, p^{\text{dam}}) = \bar{\pi}^\star(p^{\text{dam}})$ that selects the optimal power purchased from the market as function of the market prices. Therefore, by definition, $\dot{Q}^{\text{b}}(p^{\text{dam}})$ is directly defined by $\bar{\pi}^\star(p^{\text{dam}})$.

8.5.2. RL FOR DAY-AHEAD AND IMBALANCE TRADING

As with MPC, the RL-based approach to trade in both markets consists of two separate RL algorithms:

- A first RL algorithm that trades with the day-ahead market. This is the algorithm proposed in Section 8.5.1 and it is agnostic of what happens in the imbalance market.
- A second RL algorithm that trades in the imbalance market and that considers the interaction with the day-ahead market. This algorithm runs in real time and it does not build bidding functions.

TRAINING MULTIPLE RL AGENTS

As each electricity market has its own rules and working principles, it is clear that a different RL agent for each market is needed. As an example, an RL agent for the imbalance market has a different state $\mathbf{s}$ as it knows more information than the agent for the day-ahead market, e.g. it knows the prices and allocations of the day-ahead market. Based on this premise, when using RL to trade in two electricity markets, the problem becomes a multi-agent RL problem [36]. More specifically, as both agents are trying to minimize the economic cost, it becomes a collaborative multi-agent RL problem [48, 161].

While the literature has several methods for collaborative RL, e.g. join-action learners [48], the available methods are not very suitable for the case of STESSs. In particular, when training several agents at the same time, the environment becomes non-stationary [36], i.e. as each agent improves and changes its own policy the environment that the other agents perceive changes as well. This non-stationary condition invalidates the convergence properties of most single-agent RL algorithms [36]. Although there are methods that address this by allowing every agent to observe the state and actions of the other agents, these are not applicable to STESSs. In particular, due to the sequential decision making nature of electricity markets, while the imbalance agent can know the state of the day-ahead agent, the opposite is not true, i.e. the information of the imbalance market is unknown at the time bids need to be submitted to the day-ahead market.

Based on this argument, we propose a RL approach for trading in the two markets where agents are not trained simultaneously. Instead, the day-ahead agent is trained first using the algorithm proposed in Section 8.5.1, and the imbalance agent is trained afterwards including in its state information from the day-ahead market. This scheme has two benefits:

- **Convergence**: Since the two RL agents are independently trained in two stationary environments, standard RL algorithms have guarantees of convergence.
- **Flexibility**: As the imbalance market is highly volatile, STESSs owners could potentially want to stop the trading in the imbalance market during periods of high volatility. As the agent for the day-ahead market is independent, STESSs could simply use the controls of this agent and be optimal in the more stable day-ahead market.

RL FOR THE IMBALANCE MARKET

As the RL agent for the day-ahead is the same as the one described in Section 8.5.1, we only need to define the RL agent that trades in the imbalance market. The new agent, albeit similar to the one described for the day-ahead, has some modifications.

For the state space, besides the three values included in the state of the day-ahead agent, the new state includes the day-ahead energy allocation $\dot{Q}^{\text{dam}}$, past imbalance prices p^{imb}, and past imbalance volumes v^{imb}. In detail, the state $\mathbf{s}_k$ is defined by:

$$\mathbf{s}_k = (\mathbf{x}_k, \tau_k, p_k^{\text{dam}}, \dot{Q}_k^{\text{dam}}, p_{k-1}^{\text{imb}}, v_{k-1}^{\text{imb}}, \dots, p_{k-n_{\text{hrl}}}^{\text{imb}}, v_{k-n_{\text{hrl}}}^{\text{imb}}), \tag{8.14}$$

where the number of historical past values n_{hrl} is defined by the most recent lagged price $p_{k-n_{\text{hrl}}}^{\text{imb}}$ that is uncorrelated to the imbalance price p_k^{imb}. As an example, for The Netherlands, we observed $n_{\text{hrl}} = 3$ to be a good choice.

For the action space $\mathcal{U}$, a single action $u \in \mathbb{R}^{n_{\text{in}}}$ has a similar format as before, i.e. $\mathbf{u} = (u_1, u_2, \dots, u_{n_{\text{in}}})$, with each input control u_i taking $n_{\text{dis}} + 1$ discrete values. However, unlike the input controls for the day-ahead market that were defined in the range $[0, 1]$, each input control u_i can take $n_{\text{dis}} + 1$ discrete values uniformly separated between -1 and 1. Defining by $\dot{Q}_i^{\text{in,dam}}$ the energy purchased for storage device i in the day-ahead market, a value of $u_i = -1$ represents selling all the energy $\dot{Q}_i^{\text{in,dam}}$ in the imbalance market, i.e. $\dot{Q}_i^{\text{in}} = 0$. By contrast, a value of $u_i = 1$ represents buying all the energy that is still possible, i.e. $\dot{Q}_i^{\text{max}} - \dot{Q}_i^{\text{in,dam}}$, for storage device i, i.e. $\dot{Q}_i^{\text{in}} = \dot{Q}_i^{\text{max}}$. The selection of the output power is not considered as it is already selected by the day-ahead agent.

Besides the reward r that includes now the combined cost/income obtained in the imbalance and day-ahead markets, and the simulation environment that also generates imbalance prices, the other parts of the RL agent remain the same.

MARKET INTERACTION

In terms of the interaction with the agent for the day-ahead market, the STESS is controlled with both agents acting sequentially. First, one day-ahead, the day-ahead agent builds the bidding functions for the next day's day-ahead market. Next, the day-ahead market is cleared and the energy is allocated. Then, in real time, the imbalance agent uses the existing information of the day-ahead and imbalance markets to select the optimal power to buy/sell.

Unlike the agent for the day-ahead market, the imbalance agent does not build bidding functions as the imbalance market requires direct selection of the power $\dot{Q}^{\text{imb}}$ to buy/sell. As a result, the optimal policy $\pi^\star(\mathbf{s}_k)$ at time k directly selects the power to be traded based on available data $\mathbf{s}_k$ at that time step k, but not on the imbalance market price p_k^{imb}.

8.6. CASE STUDY

To study the quality of the proposed control strategies, and in order to analyze the merits and disadvantages of each one of them, we consider a real SSTES in eight case studies. First, we consider the STESS needing to satisfy an uncertain heat demand during one year while minimizing the cost when purchasing energy in the day-ahead market. Second, we consider the STESS needing to supply the same heat demand but

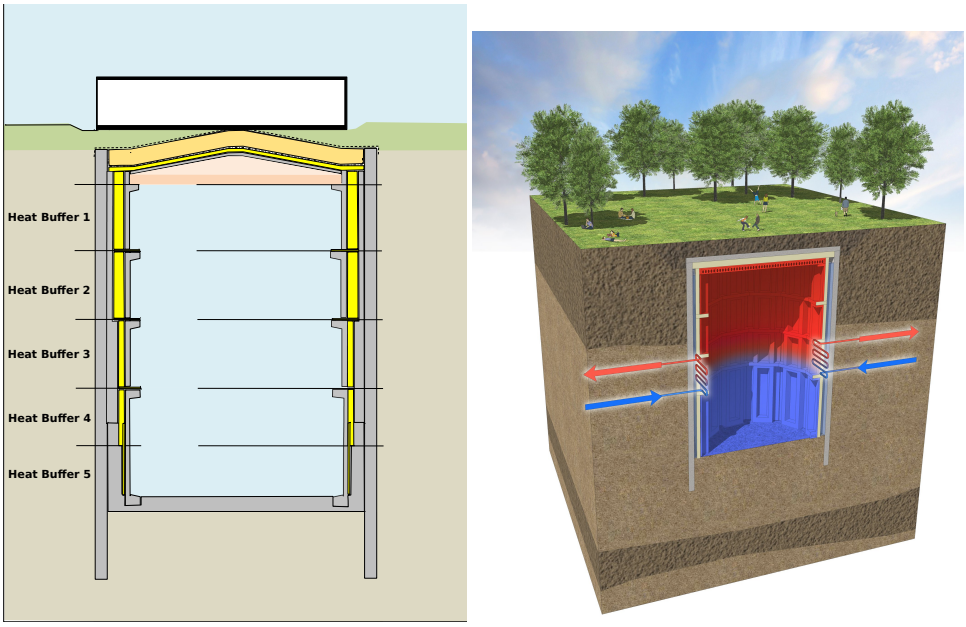


Figure 8.2: Schematic representation of the STESS. **Left**: technical scheme representing the five heat buffers in the real system. **Right**: scheme representing the underground installation of the STESS.

interacting with both the day-ahead and the imbalance market. Then, for each of the two scenarios, we consider two different heat demand profiles. Finally, for each of the two market scenarios and the two heat demand profiles, we consider two different countries.

8.6.1. REAL STESS

The considered STESS is a large subterranean thermal stratified storage vessel with the ability to store heat for seasonal periods and to supply heat demand to a cluster of buildings. The system is divided into different segments or heat buffers that can be charged and discharged separately; the system has 5 thermal buffers with the top 4 buffers (see Figure 8.2) being able to be charged and discharged independently. Figure 8.2 provides a schematic representation of the vessel and Figure 8.3 illustrates the real system when it was under construction. For further details on the system we refer to Chapter 7.

8.6.2. DATA

To set up the study, we consider the day-ahead and imbalance prices between 2015–2017 in The Netherlands[4], and the heat demand of a cluster of 5 buildings with a yearly-average heat demand of 220 MWh during the same time period[5]. As a second case study, we consider the day-ahead and the imbalance markets in Belgium during the same period, and the same heat demand.

[4] Collected from https://transparency.entsoe.eu/.
[5] Obtained from one of our research partners.

Figure 8.3: Construction of the STESS. **Left**: installation of the last heat buffer. **Right**: STESS almost completely sealed.

The data of 2015 and 2016 are used as training data for the RL agents, and as the historical data for generating scenarios. The data of 2017 are used as out-of-sample data to evaluate the performance of the different algorithms.

8.6.3. EXPERIMENTAL SETUP

In this section, we provide the details regarding the implementation of the case study. Particularly, we describe the evaluation method, the dynamics of the STESS, the implementation details of the control approaches, as well as the software used in the case study.

EVALUATION

To compare and study the control approaches, we evaluate their performance in terms of the economic cost that they incur when controlling the STESS for the full 2017 year. As a baseline, we consider the economic cost of directly buying the instantaneous heat demand $\dot{Q}^{\mathrm{d}}$ at the day-ahead market price. This baseline serves us to establish whether a control approach learns to trade energy, i.e. whether a control approach can use the STESS to reduce the energy cost. Moreover, to compare the algorithm in different conditions, the demand data are multiplied by 2 and used to evaluate the algorithms in the case of having 10 buildings, i.e. a yearly-average demand of 440 MWh.

SYSTEM DYNAMICS

The state of the STESS at time step k is defined by $\mathbf{x}_k = (T_{1,k}, T_{2,k}, T_{3,k}, T_{4,k}, T_{5,k})$, i.e. by the temperature stored in each of the 5 buffers as it is proportional to the stored energy. Similarly, as the top 4 buffers can be charged and discharged independently, the input and output power are respectively defined by $\dot{\mathbf{Q}}^{\mathrm{in}}_k = (\dot{Q}^{\mathrm{in}}_{1,k}, \ldots, \dot{Q}^{\mathrm{in}}_{4,k})$ and $\dot{\mathbf{Q}}^{\mathrm{out}}_k = (\dot{Q}^{\mathrm{out}}_{1,k}, \ldots, \dot{Q}^{\mathrm{out}}_{4,k})$.

Using these definitions, the dynamical model used for the RL simulator and for the constraints in the MPC is based on the dynamical model for thermal stratified vessels proposed in Chapter 7. Particularly, the dynamics of each heat buffer i at time k are defined by:

$$T_{i,k+1} = F_i(\mathbf{T}_k, \dot{\mathbf{Q}}^{\mathrm{in}'}_k - \dot{\mathbf{Q}}^{\mathrm{out}'}_k, \Delta t, T_\infty), \tag{8.15}$$

where F_i is defined by (7.10), where T_∞ represents the ambient temperature and is the only disturbance $\mathbf{d}$, and where $\dot{Q}^{\text{in}'}_{k,i}$ and $\dot{Q}^{\text{out}'}_{k,i}$ are the modified version of $\dot{Q}^{\text{in}}_{k,i}$ and $\dot{Q}^{\text{out}}_{k,i}$ to model buoyancy according to (7.19) and (7.20). For further details on the model we refer to Chapter 7.

MPC IMPLEMENTATION DETAILS

The OCPs that are solved are defined by (8.8), (8.9), and (8.10) where:

- The dynamical constraints is represented by (8.15).
- The maximum power $\dot{\mathbf{Q}}^{\text{in}}_{\text{max}}$ to be traded in the market is defined by the electrical installation to charge the STESS. In our case $\dot{\mathbf{Q}}^{\text{in}}_{\text{max}} = 300\,\text{MW}$.
- The individual upper limits of charging and discharging, i.e. $\mathbf{g}_{\text{in}}(\mathbf{x}_k)$ and $\mathbf{g}_{\text{out}}(\mathbf{x}_k)$, are defined by the maximum heat transfer of the heat exchangers, which in turn is proportional to the temperature difference between the tank temperature and the temperature of the fluid in the heat exchangers. In detail, $\mathbf{g}_{\text{in}}(\mathbf{x}_k) = \kappa\,(T_{\text{in}} - \mathbf{T}_k)$ and $\mathbf{g}_{\text{out}}(\mathbf{x}_k) = \kappa\,(\mathbf{T}_k - T_{\text{out}})$, where T_{in} and T_{out} are respectively the temperatures of the fluid in the heat exchangers when charging and discharging the STESS and κ is a heat exchanger coefficient.
- The limits on the STESS state are given by $\mathbf{x}_{\text{min}} = 286\,\text{K}$ and $\mathbf{x}_{\text{max}} = 363\,\text{K}$, where the lower limit is defined by the outer soil temperature and the upper limit by the safety margin to prevent water boiling in the tank.

Moreover, to use the MPC approaches, a discrete set of prices has to be defined to build the bidding functions. To do so, we selected 15 discrete prices equally spaced between 0 and 70 €/MWh. This selection was done based on the price distribution in 2015–2016 and considering the computation time of solving a single OCP; nonetheless, a coarser or finer discretization could be used to respectively decrease the computation time or to increase the accuracy of the bidding functions. For prices above 70 €/MWh the bidding function was set to 0 considering the seldom occurrence of prices above this threshold. For negative prices, the bidding function was defined as the solution at 0 €/MWh.

RL IMPLEMENTATION DETAILS

The RL control algorithms proposed in Section 8.5 can be directly applied to the current case study where:

- The time position τ is simply the day of the year.
- As the STESS has a seasonal cycle of a year, a RL episode length is defined as two years.
- The time-dependent constraints on the maximum power are implicitly enforced within the action space as the actions are normalized w.r.t. the maximum power.

SOFTWARE

The MPC algorithm is modeled using `CasADi` [11] and `python`, and then solved using `Ipopt`[6] [237]. For the RL approach, the fitted Q-iteration algorithm (see Section 8.5.1) is implemented in `python` using the `Xgboost` [42] library. The forecaster of imbalance prices is also done via the `Xgboost` library.

8.6.4. DAY-AHEAD MARKET TRADING

The main results of the first study, i.e. the comparison of MPC and RL when only trading in the day-ahead market, are listed in Tables 8.1 and 8.2. Table 8.1 displays the yearly economic cost when using both algorithms and the cost of not having an STESS, i.e. the cost of buying directly the heat demand in the day-ahead market; it also lists the economic savings of both algorithms w.r.t. the case of not having an STESS. Table 8.2 lists the offline costs, i.e. one-time computations, and online costs, i.e. real-time computations, of both algorithms.

Table 8.1: MPC and RL comparison in terms of their economic cost when only trading in the day-ahead market. The savings are computed w.r.t. the cost of not having an STESS. For each case study, the best method is indicated in bold.

		The Netherlands		**Belgium**	
		10 buildings	5 bldgs.	10 bldgs.	5 bldgs.
Cost [€]	No STESS	19384	9692	23490	11744
	MPC	**15206**	**6825**	**16826**	7033
	RL	15942	7465	17636	**7027**
Savings	MPC	**21.6**%	**29.6**%	**28.4**%	40.1%
	RL	17.8%	23.0%	24.9%	**40.2**%

Table 8.2: MPC and RL comparison in terms of their computation time when trading in the day-ahead market. The comparison is done in terms of online and offline computation time.

	Offline	**Online**
MPC	0	10–15 minutes
RL	1–2 days	<1 second

Independently of the country or heat demand level considered, the following observations can be made:

- Both algorithms can trade energy and make use of the STESS to reduce the economic cost. Particularly, using the STESS and trading optimally, the algorithms

[6] The solver `Ipopt` is selected because it is one of the best free nonlinear optimization solvers.

can reduce the economic cost by 20–40%.

- The performance of both algorithms is similar, but MPC can obtain slightly lower costs and larger profits.
- Although RL requires a long offline computation time, its cost online is almost negligible. In particular, as the optimal bidding functions are estimated offline, the computation time in real time is almost 0.
- By contrast, while MPC does not require offline computations, it needs 10–15 minutes in real time to build the bidding functions. However, as the bidding functions are submitted once per day and one day in advance, this large real-time computation cost does not represent a real disadvantage.

To illustrate the generated bidding curves of both methods, Figure 8.4 displays the generated bidding curves for the first day of the case study that considers 5-buildings and the day-ahead market in The Netherlands. As it could be expected based on the results in Table 8.1, both bidding curves are very similar.

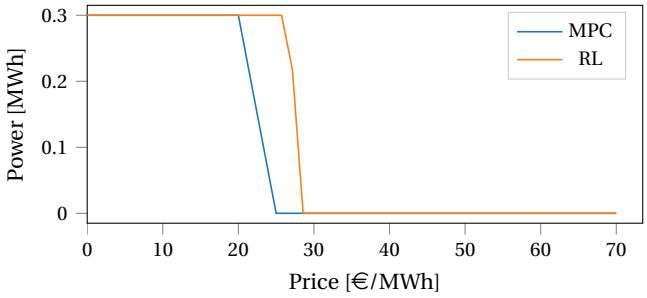


Figure 8.4: Generated bidding curves by the MPC and RL algorithms on 01/01/2017 in The Netherlands when supplying heat for 5 buildings.

8.6.5. DAY-AHEAD AND IMBALANCE MARKET TRADING

The main results of the second study, i.e. the comparison between MPC and RL when trading in both the day-ahead and imbalance markets, are listed in Tables 8.3, 8.4, and 8.5. Table 8.3 displays the yearly economical cost and economic savings of both algorithms. Table 8.4 lists their offline and online computation cost when trading in the imbalance market (the computation cost for trading in the day-ahead is the same as in Table 8.2). As an extra comparison, Table 8.5 summarizes the percentage of times that each algorithm correctly up-regulates and down-regulates the grid, i.e. the percentage of times that the algorithm sells (buys) energy in the imbalance market while the TSO tries to up-regulate (down-regulate) the system.

As before, independently of the case study considered, the following observations can be made:

Table 8.3: MPC and RL comparison in terms of their economic cost when trading in the day-ahead and imbalance markets. The savings are computed w.r.t. the cost of not having an STESS. For each case study, the best method is indicated in bold.

		The Netherlands		**Belgium**	
		10 buildings	5 bldgs.	10 bldgs.	5 bldgs.
Cost [€]	No STESS	19384	9692	23490	11744
	MPC	**9227**	3544	**10569**	4401
	RL	11176	**3437**	11468	**3872**
Savings	MPC	**52.4%**	63.4%	**55.0%**	62.5%
	RL	42.3%	**64.5%**	51.8%	**67.0%**

Table 8.4: Computation cost of the MPC and RL approaches when trading in the imbalance market. The comparison is done in terms of online and offline computation time.

	Offline	**Online**
MPC	0	30–45 seconds
RL	1–2 days	<1 second

- As for day-ahead trading, both algorithms perform very similar to each other. However, unlike in the case of only day-ahead trading, MPC no longer performs slightly better. Instead, RL performs slightly better for lower heat demand profiles (5 buildings), and MPC performs better for higher heat demand profiles (10 buildings).
- Trading in both markets is much more beneficial than trading only in the day-ahead market as the costs are halved w.r.t. day-ahead trading. In particular, while day-ahead trading reduces the economic cost by 20–40%, trading in the two market reduces the cost up to 60–70%.
- As before, RL requires large offline computation costs but negligible online computation costs. By contrast, MPC has no offline computation costs but requires 30–45 seconds to obtain the optimal trading strategy for the imbalance market. Since the imbalance market is cleared every 15 minutes and optimal decisions are made within seconds, it can be argued that the online computation cost of MPC might now represent a problem.
- When buying energy in the imbalance market, the RL algorithm helps the TSO to down-regulate the grid. In particular, approximately 80% of the times the RL algorithm buys energy, the TSO simultaneously tries to reduce the grid generation or to increase the grid consumption. Although the MPC algorithm also helps, this contribution is worse since it only helps to down-regulate 55–70% of the time.

Table 8.5: MPC and RL comparison in terms of the % of times that they correctly up-regulate or down-regulate the grid, i.e. % of times that they sell/buy energy in the imbalance market when the TSO up/down-regulates. For each case study, the best method is indicated in bold.

		The Netherlands		**Belgium**	
		10 buildings	5 bldgs.	10 bldgs.	5 bldgs.
Up-regul.	MPC	**51%**	47%	44%	47%
	RL	49%	46%	**50%**	**52%**
Down-reg.	MPC	70%	66%	68%	55%
	RL	**81%**	**81%**	**81%**	**80%**

- By contrast, when selling energy in the imbalance market, none of the algorithms help much to up-regulate: only 45–55% of the times an algorithm sells energy the TSO is simultaneously trying to up-regulate.

8.7. DISCUSSION

IN this section, based on the obtained results, we discuss the merits and disadvantages of the proposed control approaches, the benefits of using STESSs for energy trading, how to optimally operate STESSs to maximize their profits, and the generality and optimality of the proposed methods.

8.7.1. MERITS OF EACH CONTROL APPROACH

We start the discussion by analyzing the merits of the different proposed approaches in the two trading contexts.

DAY-AHEAD TRADING

When trading only in the day-ahead market, both approaches can trade energy with a similar performance despite their underlying differences. Therefore, although MPC obtains slightly lower economic costs than RL, it is necessary to consider other metrics in order to make a meaningful comparison.

When considering the online computation time, both algorithms are feasible for real-life applications. Thus, the largest difference between both approaches is the offline computation time. While this metric does not play a role most of the time, i.e. it usually represents one-time computation costs, it might be important when the system regularly goes under maintenance, something breaks down, or the market has a big change. If any of these events happens, MPC can easily adapt itself by a change in the OCP or by re-estimating the dynamical model (which does not take more than some minutes). By contrast, RL requires 1–2 days to re-estimate the optimal policy under the new conditions, which hinders the day-ahead trading. Thus, MPC has in general better adaptability to environmental conditions.

Based on this analysis, it becomes clear that MPC is a better approach when trading only in the day-ahead market. Particularly, slightly better optimal solutions together

with a better adaptability to environmental changes make the proposed MPC approach a better solution in this case.

DAY-AHEAD AND IMBALANCE TRADING

Similar to the case of only day-ahead trading, when trading in the day-ahead and imbalance market the two proposed approaches obtain good solutions. In particular, while RL performs slightly better for lower heat demand profiles (5 buildings) and MPC performs better for higher heat demand profiles (10 buildings), these difference are not very large and, as before, other metrics need to be considered.

Although the online computation time for day-ahead trading was not an issue, for the case of imbalance trading it becomes one. In detail, due to the real-time nature of the imbalance market, optimal decisions should be made in seconds. As the proposed MPC approach requires 30–45 seconds to compute an optimal solution, it can potentially fail to provide an optimal trading strategy.

As a result, even though the proposed MPC approach still has a better adaptability to environmental changes, it can be argued that is less appropriate control strategy than the proposed RL approach. The latter, with its negligible real-time computation cost, equal quality solutions, and better regulatory capabilities, is a better choice when it comes to trading in the imbalance market.

8.7.2. THE IMPORTANCE OF MARKET TRADING FOR STESSS

Based on the obtained results, it is clear that optimal control approaches, either MPC or RL, are key to maximize the profits of STESSs and to ensure their widespread use as optimal control strategies and can reduce the energy cost by 60–70%. In this context, the largest profits are obtained when the STESS trades in multiple markets. In particular, while a traditional STESS would restrict its trading to the day-ahead market to avoid unnecessary risks, in this chapter we show that STESSs can dramatically reduce their costs by using optimal control strategies and trading also in the imbalance market.

8.7.3. STESSS AS REGULATION TOOLS

Looking at the results of Table 8.5, it can be argued that the economic goal of STESSs is (partially) aligned with the regulatory duties of the TSO. In the case of RL, 80% of the times the STESS buys energy in the imbalance market, it helps the TSO to down-regulate the system. This behavior is seen for the various case studies considered, which included different imbalance markets and different heat demands. In the case of MPC, this effect is not so pronounced; nevertheless, it sill helps the TSO 55–70% of the times.

Although the same cannot be said about up-regulation, i.e. only 50% of the times the STESS sells energy in the imbalance market it is actually helping up-regulate the grid, it can be argued that wrongly up-regulating is less critical than wrongly down-regulating. In particular, if the STESS wrongly sells energy in the imbalance market, the TSO can always request somebody to reduce their generation, i.e. down-regulate. However, if the STESS wrongly buys energy in the imbalance market, the TSO has to request somebody to increase their generation; as the generation is limited, there might not be an available agent that can provide that service.

As additional remark, to further improve the regulatory services of STESSs, communication between the TSO and the STESS could be established. Particularly, in the current setup, the STESS simply optimizes its profit without considering the TSO. Thus, to improve this, the TSO could simply indicate the STESS whether it is allowed to buy or sell energy, i.e. whether the TSO plans to down or up-regulate, and the STESS could take its optimal action if it helps the TSO and its own profit.

8.7.4. GENERALITY OF THE METHODS

While the case study focused on a specific STESS, i.e. a latent heat storage via water stratification, the proposed methods are general and can be applied to any STESS. Indeed, with the proposed methods, the several challenges that prevent the development of efficient control solutions for STESS trading can be tackled, namely: scenario generation and quantification of price uncertainty for long horizons, small computation costs for real-time control, and adaptability to market changes.

8.8. CONCLUSIONS

WE have proposed several optimal control strategies for *seasonal thermal storage systems (STESSs)* when interacting with electricity markets. While in the literature there are control strategies for STESSs and there are optimal trading strategies for traditional storage systems, the former do not allow STESSs to trade in the markets and the latter are not suitable for STESSs. To fill that gap, we have proposed a *model predictive control (MPC)* and a *reinforcement learning (RL)* approach for the case of having an STESS trading in the day-ahead electricity market. In addition, we argued that trading in one market is not optimal, and proposed another MPC and another RL approach for the case of having an STESS trading in both the day-ahead market and the imbalance market.

To study the merits and disadvantages of the proposed approaches, we have considered a real STESS in eight different case studies. We have showed that, despite the similarity in the optimal solutions of the proposed algorithms, MPC is a better trading strategy for the day-ahead market due to its larger adaptability. In contrast, for trading in the imbalance market, the proposed RL approach is a more suitable control strategy as it has negligible real-time computation costs, leads to similar economic costs, and has better regulatory capabilities.

It has also been shown that STESSs are potential tools for grid regulation and that the economic incentive of STESSs are aligned with the regulatory duties of TSOs. Similarly, it has been demonstrated that optimal control strategies are needed to optimize the profit of STESSs and ensure their widespread use.

In future research, we intend to further explore the use of STESSs as regulation devices; more specifically, we will analyze the regulatory framework of the imbalance market in different countries and study changes to that framework to successfully use STESSs for grid regulation. As starting point, in Chapter 10, we propose a new market framework for imbalance trading and show the benefits of the framework by using STESSs to provide grid regulation.

III

NEW MARKET POLICIES TO IMPROVE GRID BALANCING

9

BACKGROUND: STRUCTURE OF ELECTRICITY MARKETS

All of the books in the world contain no more information than is broadcast as video in a single large American city in a single year. Not all bits have equal value.

Carl Sagan

Markets do very weird things because they react to how people behave, and sometimes people are a little screwy.

Alan Greenspan

This chapter provides background knowledge on the concepts explored in the third part of the thesis. In particular, the chapter starts by motivating the research on the structure of electricity markets. Then, it describes the working principle of wholesale electricity markets, the balancing market, and the imbalance settlement mechanism. Finally, it provides a brief introduction into seasonal storage systems, the energy systems used as case study in this part of the thesis.

9.1. INTRODUCTION

WHILE the energy transition [216] aims at considerably increasing the share of *renewable energy sources (RESs)* in the energy mix, there are several problems that need to be addressed [17] before achieving the desired goal of nearly 100% RES generation. In particular, electricity cannot be still stored efficiently and economically over long periods of time and electricity networks require constant balancing between generation and consumption. However, due to the uncertainty in the supply and demand, a perfect balance between generation and consumption is hardly possible and grid imbalances are unavoidable. To prevent grid instabilities, these imbalances have to be corrected in real time by the *transmission system operator (TSO)*.

In this context, due to the weather dependence of RES generation, electricity generation becomes more uncertain and grid imbalances become larger as the integration of RESs increases. Consequently, as we approach the 100% RES generation target, the grid becomes harder to control and TSOs struggle to keep the grid balanced. As proposed in Part I of the thesis, a potential solution to address this issue is to rely on accurate forecasting techniques to reduce uncertainty (see Chapters 3–5). A second option, as proposed in Part II of the thesis, is to reduce grid imbalances via seasonal storage systems that optimize their trading strategies in electricity markets (see Chapters 7–8).

In this part of the thesis, we propose a third solution to address this issue: changing the structure of the electricity markets to allow more systems to participate in grid balancing. In detail, while some RES systems could potentially contribute to grid balancing, they are not being used for this purpose due to the current regulations in grid balancing. Examples of such systems include solar photovoltaic installations, storage systems such as seasonal storage, or even—in some countries—wind farms. With that motivation, we argue that allowing them to trade in the imbalance settlement mechanism, i.e. the mechanism used by the TSO to financially settle imbalances, is a potential solution for those systems to assist the TSO in reducing grid imbalances. Then, we propose a new market framework for providing balancing services through the imbalance settlement mechanism. As we show, the new market framework can help reduce grid imbalances without comprising the safety of the grid operation.

As introductory material, this chapter provides background knowledge on the concepts studied in Part III of the thesis. In particular, Section 9.2 describes in detail the structure of wholesale electricity markets, the balancing market, and the imbalance settlement mechanism. Then, Section 9.3 briefly presents the energy systems, i.e. seasonal storage, used as case study in this part of the thesis.

9.2. ELECTRICITY MARKETS

TO help obtain a balanced grid, wholesale electricity markets have a very specific structure. Particularly, two specific features of the European electricity markets, namely self-dispatch and balancing responsibility, are relevant for grid balancing. Self-dispatch refers to the fact that market participants make their own decisions regarding the dispatch of their generators[1] but are obliged to submit their projected generation

[1] As opposed to systems with central dispatch (e.g. in the U.S.) where the system operator makes dispatch decisions.

and consumption schedules ahead of time [18]. Balancing responsibility refers to the fact that all market parties carry a balancing responsibility: they are financially responsible for deviations from their schedules as these deviations create grid imbalances [68, 235].

9.2.1. WHOLESALE ELECTRICITY MARKETS

In order to avoid deviations from their notified schedules, market participants can trade in different markets that are mainly distinguished by the time of execution. In particular, as market participants obtain more accurate information about their actual generation and consumption, they can adjust their schedules by trading in markets with execution times closer to real time [53]. In this context, besides bilateral trading, European actors have several organized marketplaces at their disposal [81]:

- Forward market: Electricity is traded weeks or months in advance.
- Day-ahead market: Electricity is traded up to one day in advance.
- Intraday market: Electricity is traded one day ahead of delivery to one hour or some minutes before delivery time. In addition, electricity is traded continuously, in hourly or quarterly auctions, or a mix thereof [71].

For these three markets, contracts between buyers and sellers are established in a market exchange and supervised by a market operator. Moreover, the intraday and the day-ahead markets are also sometimes referred to as spot markets.

In theory, by having all these markets with different gate closure times, market participants are provided with several opportunities to correct their imbalances. Particularly, due to generation and consumption uncertainty, it is nearly impossible for market actors to know in advance how much electricity they should trade, e.g. the electricity traded in a forward market (months in advance) is rarely the electricity that the actors would like to trade in real time. By having these different markets, actors can minimize their economic risks by trading the bulk of their energy in more stable markets (the ones with earlier execution times), and then continuously adjusting their trades to make sure that the sum of the traded electricity in all the markets matches their submitted schedule.

9.2.2. BALANCING MARKET

Despite this market structure, due to unplanned unit outages and uncertainty in the supply and demand of electricity, imbalances still occur as market agents rarely consume or generate what they have traded [178]. To avoid frequency deviations and grid failures, the TSO corrects the imbalances via the balancing market [188, 235]. In this market, participating actors offer their balancing capacity (for potential activation) to the TSO months to days ahead. Then, in real time, the TSO activates the required reserves to correct positive (generation exceeding consumption) and negative (consumption exceeding generation) grid imbalances[2].

In terms of balancing providers, the participation in the balancing market is currently fairly restricted. This is to a large extent explained by the main distinguishing feature

[2] See Figure 9.1 for exact time frame of the balancing and the wholesale electricity markets.

between the balancing market and the other markets. In detail, in order to participate, potential balancing providers are subject to a prequalification procedure. In this procedure, depending on the balancing product, i.e. up or down-regulation and primary, secondary, or tertiary reserve, balancing providers are required to satisfy certain technical requirements involving the speed and duration of activation, frequency of activation within a contracting period, or ramp-up and ramp-down rates (among others). As the prequalification requirements are fairly restrictive, only a handful of technologies are able to fulfill these criteria [188].

9.2.3. IMBALANCE SETTLEMENT

As a final step, after the TSO activates the reserves and corrects the imbalances, the costs of balancing are covered in the imbalance settlement mechanism [53] by financially penalizing the actors that caused the imbalances[3]. In this settlement, market participants are charged for the imbalance they produced within a defined time interval, which is known as the *imbalance settlement period (ISP)* and in most European markets equals to 15 minutes [81]. The unit price paid for having an imbalanced position is called the imbalance price.

TRADING WITH IMBALANCES

In some countries, e.g. Germany or France [35], it is discouraged or even forbidden to actively influence and trade with grid imbalances. Instead, market agents are expected to trade honestly[4] in the markets available before delivery time and they are expected to only generate unexpected imbalances. This rule, despite granting the TSO full control of the grid balance, is economically suboptimal as the economic incentives for imbalance trading of some market agents are in fact aligned with the balancing duties of the TSO. In detail, as prices are low during periods of positive imbalances, some market agents might be willing to buy cheap electricity during those periods; by doing so, they would indirectly help in reducing the imbalance [35]. Similarly, as prices are high during periods of negative imbalances, some market agents could be willing to reduce their consumption or to increase their generation in order to increase their profit. In both cases, not only would the market agents increase their profits, but the imbalances would be reduced and the imbalance price would decrease as the TSO would no longer have to activate more expensive balancing reserves.

Based on this argument, although some countries forbid creating imbalances for trading purposes, some others, e.g. The Netherlands [113] and the UK [35], allow this type of trading. Nevertheless, despite this consent, the TSOs in those countries have no mechanism in place to ensure that the imbalances created during trading do not harm the grid stability, e.g. they still face the risk of market actors potentially creating an imbalance that would aggravate the grid stability.

For the remainder of the thesis, we will refer to the imbalance settlement mechanism as the *imbalance market*. This is done because, since we will propose a market framework for trading with imbalances, we assume that the imbalance settlement mechanism

[3] The exact calculation of the imbalance price differs across the EU countries.

[4] We define honest trading as trading based on forecast and electricity needs, instead of trading to intentionally create imbalances to obtain economic benefits.

is simply a type of market. It is important to note that this term refers to the imbalance settlement and not to the balancing market.

9.2.4. SCHEMATIC REPRESENTATION OF ELECTRICITY MARKETS

A schematic representation of the electricity markets is displayed in Figure 9.1. The figure represents the time frame for the decision making process in each of the markets. In particular, the figure includes a timeline that spans from months ahead to real time. Over the timeline, the different markets are represented by gray boxes and their gate opening and closure times defined by the position of the vertical borders of the boxes.

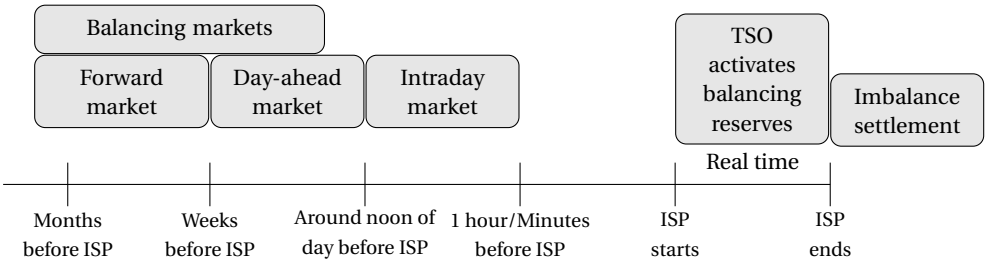


Figure 9.1: Representation of the time frame and decision making in the context of electricity markets. Grey boxes represent the different markets and their vertical borders their gate opening and closure times.

9.3. SEASONAL STORAGE SYSTEMS

THE availability of a reliable and profitable long-term energy storage is crucial for ensuring the success of the energy transition. In particular, since the penetration of solar and wind energy is expected to reach very high levels by 2030 (70-80% in some countries) [231], the uncertainty in energy supply is expected to increase. Similarly, as the generation of renewable sources is season dependent[231], e.g. the production of solar power is larger in summer than in winter, the generation of electricity is expected to be characterized by very strong seasonal fluctuations [194]. In this context, seasonal storage solutions [231], which can store energy across several months, are crucial to reduce the uncertainty and seasonal fluctuations of RESs [194].

While there are two seasonal storage technologies with capabilities to store electricity and feed it back into the grid, i.e. hydrogen storage and synthetic natural gas storage [231], in their current state they are economically non-viable. First, both technologies [180] are expensive and in early stages of development [1, 180]. In addition, synthetic fuels have a very low energy efficiency due to conversion losses [180].

An arguably better technology to store energy over seasons involves *seasonal thermal energy storage systems (STESSs)* [247]. Although these systems store electricity as heat and cannot transform it back to electricity, they have the advantage of being less expensive than electrical energy storage [194] and they represent a more reliable and mature technology that is already operational. Moreover, as 45% of the commercial and domestic energy usage corresponds to cooling and heating demand [166], STESSs have

the potential to become a key element in the energy transition as they have a large market to commercialize the stored heat. For further details on seasonal storage systems we refer to [194, 231] (see also Section 6.2).

9.4. CONCLUDING REMARKS

IN this chapter, we have provided background knowledge on the concepts used in the third part of the thesis. We have started the chapter by motivating the research on electricity markets. Then, we have described the working principles of wholesale electricity markets, the balancing market, and the imbalance settlement mechanism. Finally, we have provided a brief introduction to seasonal storage systems, the systems used as case studies in the third part of the thesis.

10

A NEW MARKET FRAMEWORK FOR GRID BALANCING SUPPORT

Understand well as I may, my comprehension can only be an infinitesimal fraction of all I want to understand.

Ada Lovelace

The dominance of short-term perspectives has led to routine decisions in the markets that sacrifice the long-term buildup of genuine value in pursuit of artificial, short-term gains.

Al Gore

To correct grid imbalances, the transmission system operator (TSO) deploys balancing reserves and settles the imbalances by penalizing the market actors that caused them. In general, it is forbidden to influence the grid imbalances to let the TSO retain full control of grid regulation. In this chapter, we argue that this approach is not optimal as it prevents market actors from balancing the grid more efficiently, e.g. some renewable sources cannot participate in the balancing market but they could have economic incentives to regulate the grid by trading with imbalances. Based on that, we propose a new market framework where any energy system is allowed to trade with imbalances. Then, we show that the new market can reduce the balancing costs and provide 10-20% of the balancing energy needed, while still granting the TSO full control of the grid balance. This is of primary importance as: (i) new approaches for grid balancing are needed since renewables are generally not used for grid balancing but its increasing integration creates higher imbalances; (ii) although seasonal storage is key in the energy transition, it needs to become profitable; as we show, the proposed market guarantees that.

Parts of this chapter are based on [140].

10.1. INTRODUCTION

ONE of the main pillars of the energy transition is the large integration of *renewable energy sources (RESs)*, a change of paradigm that is expected to increase grid imbalances [129]. In this context, as traditional power plants are taken off the grid, it is paramount that RESs contribute to grid balance if the grid is to remain stable. However, in the current market framework, many RES technologies are not allowed to participate in the balancing market due stringent prequalification requirements and/or due to the procurement time frames. Particularly, the prequalification requirements, which include aspects such as the speed and duration of activation, are largely based on the technical capabilities of traditional large-scale generators and are very hard or nearly impossible to fulfill for RESs[1]. In addition, since balancing services are often procured days or weeks in advance, RESs cannot properly quantify their balancing potential due to their inherent generation uncertainty.

Besides traditional power plants being taken off the grid, a second argument in favor of RESs contributing to grid balancing is the need for new mechanisms to ensure economic incentives for RESs. As the policies for supporting RESs are being phased out, it is necessary to identify new value streams to sustain investment in RESs and to advance the energy transition. In this context, allowing RES systems to participate in the balancing services, not only might ensure a more reliable grid operation, but also provide new sources of revenues for RESs.

These two issues are especially relevant for *seasonal thermal energy storage systems (STESSs)*. These systems, despite their potential for grid balancing support, cannot fully help to reduce grid imbalances due to the current electricity market rules. In particular, as STESSs cannot transform the stored heat back to electricity, they either cannot participate in the balancing market or they are limited to down-regulation[2]. As a result, despite their potential to adapt to the seasonal fluctuations of RESs and to correct grid imbalances (see Chapter 8), their capability to help is limited. This is especially hurtful for STESSs as their business model relies on trading in markets with large price volatility, i.e. using their flexibility in order to buy electricity during the periods of very low prices, and thus they would ideally trade in the balancing market.

Besides the prequalification requirements, a second issue with the current balancing markets is their high prices. In detail, the high technical requirements make the number of balancing service providers very limited, i.e. between a few and a few dozen depending on the country and balancing product (see e.g. [192]). In this context, it is easier for actors to exert market power and to lead to situations where balancing prices reach thousands of euros. As an example, if we consider the balancing market in Germany in 2018, the average balancing price was approximately 43 €/MWh, but the prices oscillated between 2013 and -1873 €/MWh. Similarly, if we look at the same balancing market but in 2017, the deviations are even larger: while the average price was 33 €/MWh, the prices oscillated between 24455 and -2558 €/MWh.

[1] For a comprehensive discussion of the barriers to RESs in the balancing market please see [188].

[2] Down-regulation might not even be profitable for STESSs if they have to bid too much in advance as markets with execution times closer to real time might offer more volatile prices.

CONTRIBUTIONS AND ORGANIZATION OF THE CHAPTER

To tackle the described problems, we propose a new market framework for providing balancing services through the imbalance settlement mechanism, a.k.a. imbalance market. The goal of the new framework is to incorporate systems that cannot participate in the traditional balancing market into the portfolio of balancing providers. The core idea is to allow trading in the imbalance market under direct control of the TSO so that the grid stability is never compromised. In detail, market actors send their bids to the TSO stating their availability to deviate from their schedule. Then, the TSO automatically activates the available units in real time using a process that is similar to the one employed for standard balancing products. As the deviations are controlled by the TSO, participants cannot worsen the frequency regulation. Hence, the grid imbalances are reduced without comprising the grid safety.

Besides increasing the number of balancing services, the proposed framework also has the advantage of ensuring competitive trading practices and avoiding the gaming practices of the balancing market. Particularly, as actors do not bid prices but just imbalance volumes, their ability to behave strategically and to drive market prices is very limited; similarly, as any actor can participate in the new market, the market power of a single actor becomes smaller.

To study the proposed market framework, we consider a real STESS as a case study. This selection is done because, in addition to not being able to participate in the traditional balancing market, STESSs are arguably one of the key elements to obtain a smooth and reliable grid operation [194]. Using the proposed framework and optimally controlling an STESS we show that: (i) the proposed market framework allows actors such as STESSs to efficiently assist the TSO in stabilizing the grid; (ii) STESSs can increase their profits while helping reduce grid imbalances; (iii) the TSO can reduce the cost of balancing the grid without losing control over the grid regulation. Despite using STESSs as a case study, it is important to note that the proposed market framework is very general and is valid for any type of technology with the same property: not being able to participate in the balancing market, but having economic incentives to trade in the imbalance settlement to reduce grid imbalances.

The remainder of the chapter is organized as follows: Section 10.2 introduces the proposed market framework. Section 10.3 defines different case studies and presents the obtained results. Then, Section 10.4 discusses and analyzes the results. Finally, Section 10.5 concludes the chapter.

10.2. METHOD

IN this section, we introduce the proposed market framework for trading in the imbalance market.

10.2.1. MARKET FRAMEWORK

The core idea of the new market framework is to allow trading in the imbalance market, coupled with a communication channel with the TSO to facilitate coordination and to prevent situations in which actors would negatively affect grid stability.

MARKET MECHANISM

In detail, the market framework can be divided in three sequential steps:

1. Before the beginning of each *imbalance settlement period (ISP)* (see Section 9.2.3 for details), the actors communicate to the TSO how much power they are willing to provide for upward and/or downward regulation. Particularly, they send a bid to the TSO indicating the availability of each of their units to create a positive or negative imbalance together with the maximum volume of that imbalance. It is important to note that these bids are not traditional bids as they do not include a price.

2. Then, at any time during the ISP, the TSO activates any unit whose bid helps to regulate the grid. To keep full control, the activation of the units is automatically done as with traditional balancing products: the TSO sends a direct signal to the unit and the unit automatically generates or consumes the requested power.

3. At the end, each actor pays or receives the imbalance price multiplied by the net imbalance volume created, or an economic penalty if they failed to provide the requested regulation.

It is important to note that the timeline to submit bids is not restricted by the market framework. Instead, it is a decision variable that the TSO defines given two considerations: (i) it should be possible to submit bids after the intraday gate closure time so that actors have an updated schedule; (ii) all bids must be submitted before the beginning of the ISP so that the TSO knows the available balancing energy in the imbalance market.

A timeline of the different electricity markets including the proposed imbalance market is displayed in Figure 10.1 (see Section 9.2 for details on the different electricity markets). As can be seen, bidding in the new market can be done after the intraday market closes, but all the bids are submitted before the beginning of the ISP. Then, the TSO activates in real time the participating actors similar to how it does for the regular balancing service providers.

A simplified schematic view of the framework is depicted in Figure 10.2, which represents a possible interaction between the TSO and an actor with three units. In the example, before the ISP starts, the actor sends a bid. Then, during the ISP, the TSO automatically activates units 1 and 2 and deactivates unit 2 before the end of the ISP. Finally, the actor gets paid the imbalance price.

MARKET BENEFITS

The advantages of this framework are fourfold:

- As the TSO automatically activates the actors' units to correct grid imbalances, it does not lose control of the grid balance.

- For market actors that cannot participate in the balancing market, e.g. seasonal storage systems or solar photovoltaic farms, this framework allows them to contribute to grid regulation and to increase their profit.

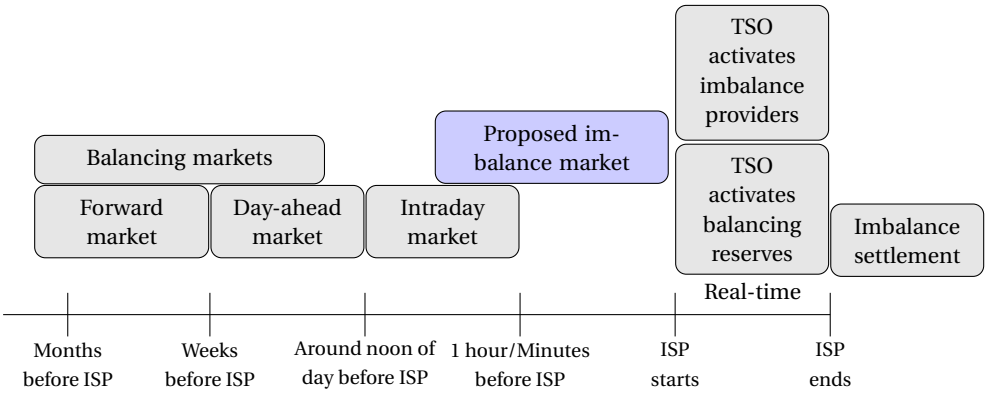


Figure 10.1: Representation of the time frame and decision making process in the context of electricity markets including the proposed market framework.

- The cost of balancing can only decrease as the actors only get paid the imbalance price. More specifically, if they are activated, in the worst case the price does not change; in the best case, the price decreases as their activation prevents a more expensive balancing technology from being used.
- The TSO can count on more balancing flexibility without losing control over the grid and use that flexibility to reduce the balancing cost.

DIFFERENCE WITH UNCONTROLLED IMBALANCE TRADING

As a final remark, it is important to note that the proposed imbalance market differs from the case of imbalance trading as considered in the previous chapter. In particular, in Chapter 8 we considered that STESSs were in a market where imbalances could be influenced, e.g. The Netherlands, but where there was no mechanism in place to prevent market actors from worsening the grid balance. As we showed in Chapter 8, such uncontrolled mechanism leads to market actors that create imbalances that worsen the grid balance. Therefore, in contrast with the proposed framework, trading with imbalances as it is done now in certain countries does not necessarily help to reduce grid imbalances but can instead increase them.

10.2.2. ADDED VALUE FOR THE EXISTING MARKETS

Although some arguments could be made against the proposed market, we argue that the new framework does not only complement but also adds value to the existing markets. In this section, we discuss the added value of the proposed market as well as the possible arguments against it.

ADDED VALUE FOR THE BALANCING MARKET

A potential argument against the proposed framework is that if there already exists a liquid and working balancing market, the proposed market framework might be unnecessary. However, in the current balancing market, there are two issues that prevent some

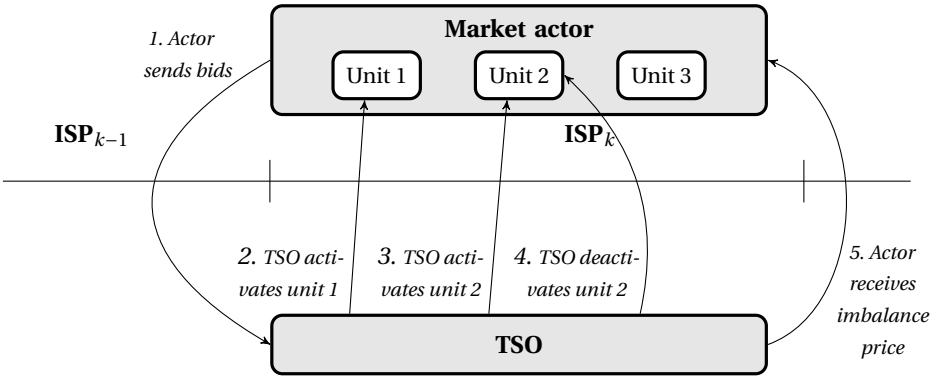


Figure 10.2: Representation of the proposed market framework for a single ISP and a single actor with 3 units. The arrows cross the timeline at the time intervals during which the action described takes place. In this example, the TSO activates unit 1 until the end of the ISP, and unit 2 for a short period of time.

actors to balance the grid and that arguably makes the procurement of balancing actors economically inefficient [119, 188]:

- **Prequalification:** As explained in the introduction, prequalification requirements leave out of the balancing market actors such as thermal seasonal storage systems, solar photovoltaic farms, or in some countries even wind turbines. Considering the increasing integration of these systems into the electricity grid, it is important to find a way to also integrate them into the balancing portfolio of the TSO.
- **Time frame:** In most balancing markets, participants are expected to send their bids hours or days in advance. Due to the uncertainty of RES generation, this deadline constrains the participation of RES systems as they are unable to accurately quantify their regulatory power in advance. Similarly, due to the same deadline, systems that cannot generate power are limited to down-regulation. Since these systems could in theory provide balancing services, their current use might be economically inefficient. Particularly, by allowing them to balance the grid, the cost of balancing the grid can only decrease.

With the proposed market framework, these issues would be solved. As prequalification would not be needed, more systems could take part in balancing the grid. In addition, as bids are submitted closer to real time, the uncertainty of RESs would decrease and RES systems would be able to provide balancing services. Similarly, since actors balance the grid by deviating from their schedule, they are not limited to down-regulate the grid as they can purchase energy in other markets, e.g. the day-ahead market, and use that energy to provide up-regulation.

Despite these benefits, it could still be argued that the time frame problem will be solved in the near future as European countries will have to adjust their balancing markets in order to incorporate free bids[3] and bids close to real time [51]. However, there are two issues with this argument:

[3] Free bids are bids that are submitted by actors whose capacity was not reserved ahead of time.

- Bidders are still required to pass the prequalification process, which limits the use of important RES systems, e.g. solar farms, for grid balancing. By contrast, the proposed imbalance market framework allows integrating many participants into the balancing portfolio of the TSO and has the potential of lowering balancing costs.
- The proposed market framework has an advantage over the balancing market with free bids: it avoids gaming practices. In detail, with free bids, market actors can submit strategic bids and drive the balancing price if they have market power. By contrast, in the proposed market framework, market actors only bid the imbalance volume but not the price and have to accept the unknown imbalance price without driving the balancing cost. As a result, the proposed market framework has the potential to ensure competitive trading practices and to lower the balancing costs.

ADDED VALUE FOR THE INTRADAY MARKET

Another possible argument against an imbalance market could be that STESSs and other technologies could use the intraday market instead of the imbalance market for reducing imbalances. More specifically, as actors usually participate in the intraday market as their last resort to correct imbalances, RES technologies could already trade on the intraday market and help other actors to reduce their imbalances. However, this approach is not sufficient if it is intended for reducing the grid imbalances and incentivizing the use of RESs:

1. The energy traded by RESs in the intraday market does not necessarily ensure that the grid imbalances are reduced. In particular, while the grid imbalance might be negative, a given actor might have an individual positive imbalance[4] and correct it in the intraday market by selling the excess energy to a STESS. In that scenario, it is clear that it would be more beneficial for the grid balance if the STESS would create an individual positive imbalance, i.e. sell electricity in the imbalance market, instead of buying electricity in the intraday market.
2. In terms of the economic profits of RESs, it could more beneficial to trade with imbalances. If we compare the intraday market to the imbalance market, there is a price spread that favors imbalance trading. For example, in Germany, the imbalance price during up-regulation is on average 20 EUR higher than the price in the intraday market [129]. In those scenarios, it is clear that for RESs it is more beneficial to sell energy in the imbalance market rather than in the intraday market. Similarly, in Germany, the imbalance price during down-regulation is on average 37 EUR cheaper than the price in the intraday market [129]. In those situations, storage systems like STESSs would buy energy in the imbalance market rather than in the intraday market.

In short, the proposed market framework can be seen as an additional tool in the balancing portfolio of the TSO that complements the existing balancing market. This new tool can lower the balancing costs, allow larger participation of market actors in balancing services, and ensure competitive trading practices.

[4] The grid imbalance is the cumulative imbalance of all the actors. Thus, it is possible for the imbalance of a given actor to have a different direction than the cumulative grid imbalance

10.2.3. CONTROL ALGORITHM

The proposed market framework is based on the idea that, while there are systems that cannot participate in the traditional balancing market, these systems may have economic incentives that are aligned with the balancing responsibilities of the TSO. Therefore, any system that participates in the proposed market framework needs a control algorithm that optimizes its profits, i.e. that exploits its economic incentives. In particular, the system requires an algorithm that can decide, before the beginning of each ISP, whether up or down-regulation is economically beneficial for the system during that ISP.

In this chapter, as the framework is analyzed in the context of STESSs, we consider a modified version of the optimal control algorithm for STESSs proposed in Chapter 8 (see Section 10.3.3 for details).

10.3. CASE STUDY

As a case study, we consider a real SSTES from the company Ecovat [69] trading in the proposed market. Based on experimental results we show that, while the STESS increases its profits by trading in the new market, the proposed market improves grid stability as it reduces grid imbalances.

10.3.1. GOAL OF THE STUDY

The aim of the study is to quantify how much balancing energy the TSO can save by using the new market and a single actor, i.e. an STESS. To perform the study, we have built a simulation environment of the STESS and of the different markets. For the markets, we have built a simulator that replicates the day-ahead market, the proposed imbalance market, and the balancing actions of the TSO. For the STESS, we have considered its dynamic model as described in Chapter 7.

As a secondary goal, we also study how profitable it is for the STESS to participate in the proposed market. In particular, we compare the profits of the STESS when trading only in the day-ahead market to its profits when trading in both the day-ahead and the proposed imbalance market.

10.3.2. REAL STESS

The considered STESS is similar to the one employed in Chapters 7 and 8. In particular, it is a large subterranean thermal stratified storage vessel with the ability to store heat for seasonal periods and to supply heat demand to a cluster of buildings. The system is divided into different segments or heat buffers that can be charged and discharged separately. The size of the system and the number of heat buffers depends on the heat demand and the use case of each particular STESS. In this case study, in comparison with the STESS considered in Chapters 7 and 8, we consider an STESS that is 13 times larger. In particular, we consider an STESS that is being built in Arnhem (The Netherlands) to cover the heat demand of 500 houses. The system contains 20000 m^3 of water, can store 1.3 GWh of energy, can supply a yearly heat demand of 2.8 GWh, and has a maximum electrical power of 1 MW. Figure 10.3 provides a schematic representation of one of these vessels and illustrates one of them during the construction phase.

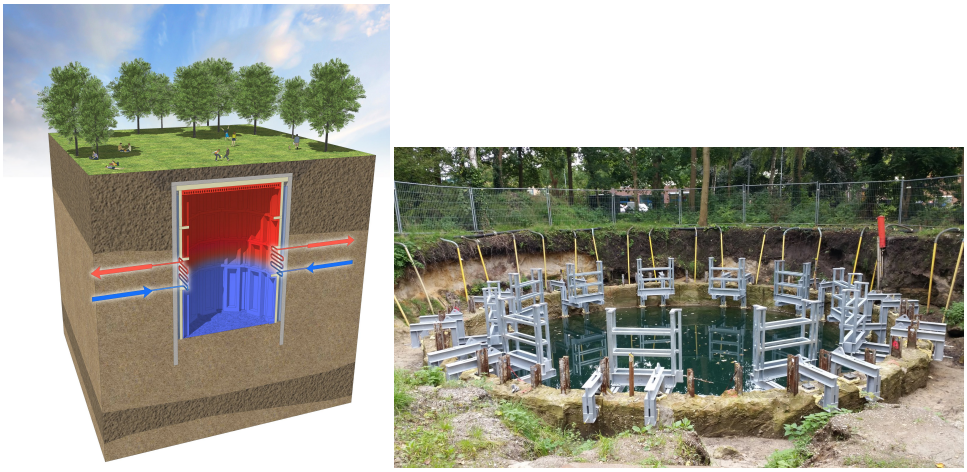


Figure 10.3: Representation of an STESS. **Left**: scheme representing the underground installation. **Right**: real STESS under construction.

10.3.3. CONTROL ALGORITHM

Any system that participates in the proposed market framework needs a control algorithm that can decide, before the beginning of each ISP, whether up or down-regulation is economically beneficial for the system during that ISP. For this case study, we consider a modified version of the optimal control algorithm for STESSs proposed in Chapter 8.

The original algorithm consists of two collaborative *reinforcement learning (RL)* [223] agents that trade in the day-ahead and in the imbalance market. In detail, given a stochastic heat demand that the STESS needs to satisfy, the goal of the algorithm is to maximize the profits of the STESS while satisfying the heat demand. To do so, the first RL agent buys electricity in the day-ahead market and the second RL agent chooses between selling in the imbalance market the energy purchased in the day-ahead market or buying more energy in the imbalance market. If the second agent sells the day-ahead energy and the overall grid imbalance is negative, the STESS helps the TSO to up-regulate the grid. Similarly, if the STESS buys energy in the imbalance market and the grid imbalance is positive, the STESS helps the TSO to down-regulate the grid. A scheme of the control algorithm is depicted in Figure 10.4. It is important to note that in this scheme the RL agent simply takes a decision, i.e. creates an imbalance, without knowing the actual imbalance price or imbalance volume. Thus, unlike with the proposed framework where the TSO controls the created imbalances, the decisions of the RL agent do not necessarily help to regulate the grid.

In this chapter, in order to fit the control algorithm to the new market framework, the control algorithm has two modifications:

- Instead of directly taking the desired action in the imbalance market (desired imbalance position), the agent sends a bid to the TSO before the beginning of the ISP. Then, the TSO choses to activate that actor if needed.

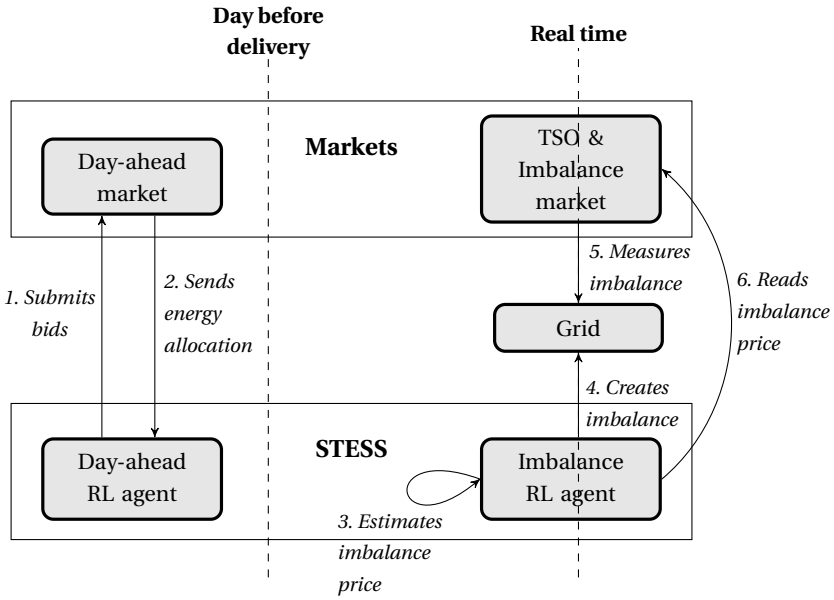


Figure 10.4: Representation of the original optimal control algorithm for trading in the day-ahead and imbalance markets proposed in Chapter 8.

- Instead of taking the desired action for an entire ISP, the agent is activated by the TSO only for the time within the ISP when the action is really needed (the TSO might down and up-regulate within the same ISP).

The new control algorithm is depicted in Figure 10.5. For the sake of simplicity, further details about the control algorithm are not provided as the focus of the chapter is the proposed market framework and the control algorithm is one of many trading strategies to interact with the new market. However, the details of the algorithm can be found in Chapter 8.

It is important to note that, to fully maximize the profits, it could be argued that trading in the intraday and forward markets should also be considered. However, the proposed imbalance market framework would work exactly the same if the STESS would trade in the forward, day-ahead, and intraday markets. The only difference in that case would be that the STESS would up-regulate using the net power purchased in the three markets. As the goal of the chapter is to show the potential of the new market framework for grid regulation, a simpler control strategy is thus employed.

10.3.4. EXPERIMENTAL SETUP

In this section, we describe the specific experimental setup used to analyze and study the proposed imbalance market. In particular, we explain how the proposed market is simulated and which data and software are used.

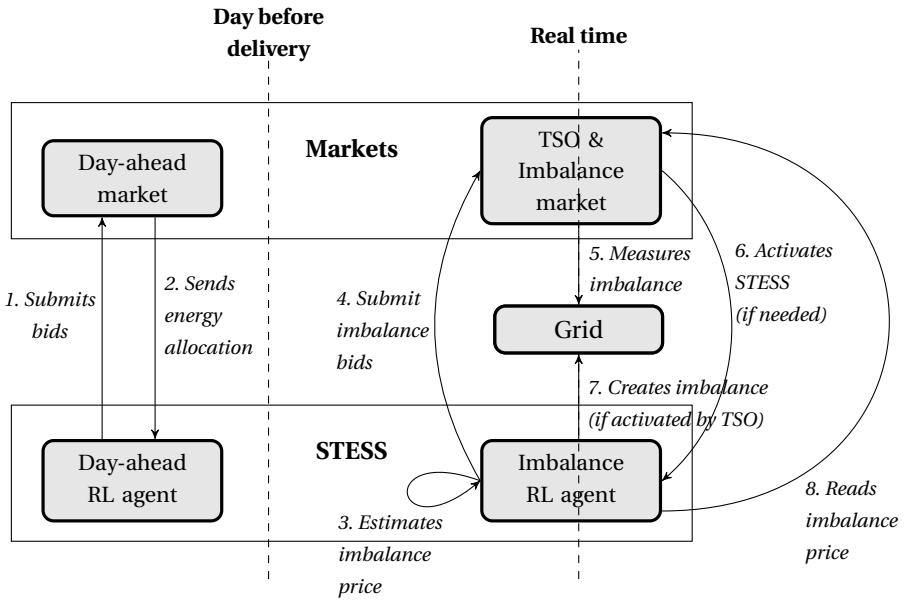


Figure 10.5: Representation of the modified optimal control algorithm for trading in the proposed market framework.

DATA

To build a simulator of the new market framework, real data regarding day-ahead market prices, imbalance prices, the activation of balancing products by the TSO, and the heat-demand that the STESS satisfies are required.

For the prices, we consider the day-ahead and imbalance prices in The Netherlands (as the STESS is located there). For the behavior of the TSO, we consider the activated volume of secondary balancing energy reserve, a.k.a. automatic frequency restoration reserve, in The Netherlands in 1-minute intervals. The prices are collected using the ENTSO-E transparency platform [230] and the activated balancing products using the TenneT transparency platform [92].

To simulate the behavior of the STESS, we consider the real heat demand that the STESS supplies: the heat demand of a cluster of 500 buildings (with a yearly-average energy consumption of 2.8 GWh)[5] using the same time resolution as the imbalance market, i.e. 1 minute.

Regarding the periods considered, the data are collected for the years 2015 to 2017. Then, the data of 2015 and 2016 are used as training data for the control algorithm, and the data of 2017 are used as out-of-sample data to evaluate the performance of the proposed market framework.

[5] Obtained from one of our research partners.

MARKET FRAMEWORK SIMULATOR

To simulate the market framework, we assume that the STESS is a price taker, i.e. that its bids do not modify the market price[6]. Then, we allow the STESS to purchase electricity in the day-ahead market if it bids below the historical market price. For the imbalance market, we allow the STESS to submit imbalance bids up to 3 minutes before the start of the ISP. Then, we simulate the clearing of the imbalance market as follows: for each of the 1-minute intervals, we activate the STESS if the submitted bid reduces the original activated volume. In this way, this simulation provides the yearly balancing energy that the new market and a small sized actor, i.e. an STESS, can potentially save the TSO.

It could be argued that using 1-minute intervals limits the activation of the STESS as the STESS can only be activated at the beginning of each of these 1-minute intervals. Instead, one would ideally consider the exact time when individually products are activated and replace these products with the balancing energy provided by the STESS. However, the data regarding individual activated products are not available, and the data with the smallest time resolution that are available are the cumulative activated products in 1-minute intervals. Nonetheless, this limitation only underestimates the full potential of the new market framework as the STESS cannot be activated as often as it would be in real life[7].

VARIABILITY IN THE EXPERIMENTAL SETUP

Due to the nature of the STESS and its controller, the obtained results are not deterministic:

- The performance of the RL agent depends on the initial conditions of the training algorithm. Thus, the results will vary depending on how the RL control algorithm is estimated.
- The maximum balancing energy that the STESS can provide depends on the initial energy in the STESS.

To account for this variability, the experiment is repeated 100 times by taking into account these two sources of uncertainty. In particular, the experiment is repeated for 10 different RL agents that are trained using 10 different random initializations, and for 10 STESSs that have 10 different initial states[8].

SOFTWARE

The entire market simulator is developed in `python` and the model of the STESS is implemented using `CasADi` [11]. As in Chapter 8, the optimal control algorithm is implemented as a RL agent using the fitted Q-iteration algorithm [73] and the `Xgboost` [42] library.

[6] This is a reasonable assumption considering the large energy volume traded in the market in comparison with the maximum energy that the STESS can trade.

[7] This holds as long as the start-up time of the STESS is below 1 minute. For this case study, this is true as the STESS can be activated within seconds.

[8] The RL agents are randomly initialized by using the first 10 positive integers (1…10) as random seeds for the library [42] that models the agents. The initial states are randomly initialized by uniformly sampling from a dataset of real measurements.

10.3.5. RESULTS

The obtained results are listed in Tables 10.1 to 10.4. In detail, Table 10.1 shows the balancing energy that would have been provided by the STESS in 2017 using the new imbalance market. Similarly, Table 10.2 lists the amount of time that the STESS is used for balancing the grid. From these two tables, the following observations can be made:

- The STESS is able to provide 674 MWh of balancing energy for up-regulation and 1426 MWh for down-regulation.
- While the energy for up-regulation and down-regulation can vary by 24 % depending on the control algorithm and the initial state, the total amount of balancing energy only varies by 18 %. A possible reason for this is that, depending on the initial state and/or the RL agent, either up-regulation or down-regulation could be more profitable.
- The STESS is activated to balance the grid during 2099 hours, which is equivalent of being activated 24 % of the year.

Table 10.1: Balancing energy provided by the STESS during 2017.

Energy	
Up-regulation	673.6 ± 161.6 MWh
Down-regulation	1425.8 ± 347.6 MWh
Total regulation	2099.4 ± 383.3 MWh

Table 10.2: Amount of time (in 2017) where the STESS is used to balance the grid.

	Time
Total	2099.4 ± 383.3 hours
Percentage over the year	24.0 ± 4.4 %

Besides analyzing the total amount of balancing energy provided by the STESS, we also study the relative contribution of that energy to the total energy activated by the TSO. This metric is provided in Table 10.3 for up-regulation and down-regulation. From this table, it can be observed how the STESS can approximately supply the TSO between 0.25 % and 0.6 % of the total balancing energy needed.

Finally, it is also important to analyze how profitable it is for the STESS to participate in the new market framework. For that, Table 10.4 lists the economic costs of the STESS when it trades optimally on the day-ahead market[9], i.e. the standard case for these type

[9] To control the STESS to trade optimally in the day-ahead market, we consider the optimal control algorithm proposed in Chapter 8.

Table 10.3: Balancing energy used by the TSO during 2017 and percent contribution of the STESS to that amount.

	TSO Balancing Energy	STESS contribution
Down-regulation	281323 MWh	0.51 ± 0.12 %
Up-regulation	224623 MWh	0.30 ± 0.07 %

of systems, and when it does so but in both the day-ahead and the imbalance markets. It can be observed how, by trading in the new market framework, the STESS can reduce its costs by 40 % and the variability on these savings is minimal, i.e. only 0.5 %.

Table 10.4: Comparison in terms of economic cost and savings during 2017 between trading in the day-ahead market and trading in both the day-ahead and the imbalance market.

	Cost	Savings
Day-ahead market	93661 €	
Day-ahead + imbalance market	56252 ± 580 €	39.9 ± 0.5 %

10.4. DISCUSSION

IN this section, we discuss the obtained results and analyze the benefits of the proposed market framework. To do so, we first discuss the results in the context of the TSO and the grid balance, and then in the context of the energy transition.

10.4.1. POTENTIAL OF THE IMBALANCE MARKET

The TSO clearly benefits from the new market framework as it obtains extra balancing energy that can only reduce the overall balancing costs. Even from a single relatively small STESS, the TSO is able to extract 2000 MWh of balancing energy in a given year. Although this amount represents only 0.35 % of the total balancing energy needed, the considered STESS is a relatively small system. In particular, the current STESS only serves the heat demand of a small neighborhood, i.e. 500 houses. If we consider 25-50 of these systems (something very reasonable in a country like The Netherlands), STESSs alone could potentially reduce the current balancing demand by 10-20 %.

Based on these results, it becomes clear that the proposed market framework has the potential to provide a large share of the energy required to keep the grid balanced. Particularly, besides STESSs, the proposed market framework is designed to integrate more systems into this new portfolio of balancing resources. In this context, while accurately estimating the potential contribution of the new framework is nearly impossible (it would require a simulation including all possible systems that would participate in this market), we can consider the relative contributions of STESSs and make some qualitative inferences. In detail, STESSs are not very large systems, i.e. the considered STESS had

a maximum power of 1 MW. Therefore, if STESSs can already provide 10-20 % of the total balancing energy needed, it is clear that the market framework has the potential to revolutionize the balancing market, provide a large share of the balancing energy needed, and reduce the total balancing cost.

It is important to note that all these benefits come without any additional operational costs or risks for the TSO: even though the new market is based on imbalance trading, the TSO has full control over the grid stability. Moreover, as the market actors do not bid prices but only volumes, the new market can only drive the balancing costs down.

10.4.2. BENEFITS FOR THE ENERGY TRANSITION

Besides benefiting the TSO and its balancing duties, the new market framework promotes the use and expansion of long-term energy storage solutions. By participating in this new market, STESSs can reduce their operational costs significantly. Thus, the importance of the new market framework in promoting a widespread use of STESS is evident.

This is of great importance as the availability of reliable and profitable long-term energy storage is crucial for ensuring the success of the energy transition. As the new market framework encourages the use of these type of storage systems, it is a potentially very valuable tool to further advance the energy transition.

10.5. CONCLUSIONS

IN this chapter, a new market framework to aid the transmission system operator (TSO) to balance the grid has been proposed. The framework is based on an adaptation of the current imbalance settlement mechanism to explicitly allow trading with imbalances under the supervision of the TSO. The goal of the new framework is to allow *renewable energy sources (RESs)* and other systems to actively contribute to grid regulation. In detail, RES integration and seasonal storage solutions that are economically viable are key for the energy transition. However, the design characteristics of the traditional balancing market prevent RESs and seasonal storage systems from participating in it. This limitation poses two problems for the energy transition. First, as RESs are a common source of grid imbalances, they cannot be effectively integrated into the energy mix without allowing them to contribute to system stability. Second, as the business case of seasonal storage systems is to exploit price differences in volatile markets, their lack of access to the balancing market limits their profitability and their economic viability.

The proposed market framework solves these issues by integrating RESs and seasonal storage systems into the portfolio of balancing resources. Besides permitting these technologies to balance the grid, the proposed imbalance market has three main advantages: (i) it increases the number of balancing resources available to the TSO; (ii) it ensures competitive trading practices and avoids the gaming practices of the balancing market; (iii) it allows the actions of market actors to be controlled by the TSO so that grid stability is never compromised.

To demonstrate and quantify the benefits of the proposed market, a real *seasonal thermal energy storage system (STESS)* trading in the new market was considered as a case study. Based on the obtained results we show that:

- The proposed market framework can provide a large share of the balancing energy needed and reduce the total balancing cost of the TSO. Particularly, by using 25-50 small-scale STESSs, the new market can provide 10-20 % of the total balancing energy needed.
- Even though the new market is based on trading with imbalances, the grid stability does not worsen.
- STESSs can significantly reduce their operational costs by participating in the imbalance market. This is key to ensure their economic viability and, in turn, to guarantee the widespread use of seasonal storage solutions needed in the energy transition.

As future research, we will evaluate the performance of the new market using other RES systems. In addition, we will further assess the proposed market using the data from other European countries and we will consider more complex trading strategies to quantify the maximum potential of the proposed market.

11

CONCLUSIONS AND RECOMMENDATIONS

So comes snow after fire, and even dragons have their ending!

John R. R. Tolkien

I'd put my money on the sun and solar energy. What a source of power!
I hope we don't have to wait until oil and coal run out before we tackle that.

Thomas Alva Edison

In this thesis, we have proposed different approaches to reduce grid imbalances while increasing the profitability and integration of *renewable energy sources (RESs)*. The proposed approaches were based on optimal market trading and on exploiting the win-win interactions between market actors and grid balance. In detail, we first proposed novel forecasting techniques that help to reduce RES uncertainty and to increase their profitability. Then, we developed modeling and control strategies for seasonal storage systems that grant more energy flexibility and help to absorb grid imbalances. Finally, we proposed a new market framework to overcome the limitations of the existing balancing markets and to reduce the cost of balancing the grid. In this last chapter, we first present the main contributions and results of the thesis. Then, we discuss the impact that this research has for society. Finally, we suggest some promising topics for future research.

11.1. CONCLUSIONS

THIS PhD dissertation contributes to the energy transition by developing approaches that help incentivize the use of *renewable energy sources (RESs)* and mitigate the negative effects of their integration, i.e. grid imbalances. In detail, the main contributions of this thesis can be summarized as follows:

- **Modeling market integration in price forecasting models**
  Motivated by the increasing integration among electricity markets, in Chapter 3 we have proposed the first techniques to include market integration in electricity price forecasting models. Then, we have shown how market integration plays a large role in price dynamics, and demonstrated that the proposed approaches are key to improve the accuracy of forecasting prices. As improvements in forecasting accuracy increase the profits of RESs, this contribution helps incentivize the use and integration of RESs. In addition, as better forecasts also lead to a more predictably behavior of RESs, this contribution also helps to obtain a grid that is easier to balance.

- **Deep learning techniques for forecasting electricity prices**
  Considering that several energy applications have recently benefited from the field of *deep learning (DL)*, in Chapter 4 we have investigated DL techniques for electricity price forecasting. In particular, we have proposed four different DL forecasting models and shown that the new techniques improve the state-of-the-art methods from the literature. As before, this contribution helps incentivize the use and integration of RESs and helps to obtain a grid that is easier to balance.

- **Forecasting solar irradiance without local data**
  In Chapter 5, we have proposed a generalized model for solar irradiance that can forecast the irradiance in any location without the need of ground measurements. Using a case study, we have shown that the method is not only location independent but more accurate than existing forecasting techniques. As solar generators are geographically dispersed and ground measurements are not easy to obtain, the proposed method is paramount to forecast solar irradiance everywhere. Consequently, this is key to reduce the uncertainty of solar generation and in turn the imbalances in the grid due to RES generation.

- **Low-complexity model for thermally stratified seasonal storage**
  Long-term (seasonal) energy storage is key to absorb the seasonal fluctuations of RESs. In this context, thermally stratified storage tanks are one of the most widely used seasonal storage technologies. With that motivation, in Chapter 7 we have proposed a new dynamical model for stratified tanks that tackled the problems of the existing models, i.e. being too complex for optimization problems. In particular, the low complexity of the proposed model allows stratified tanks to be integrated in real-time control and optimization. This contribution is vital to maximize the profits of these systems, ensure its widespread use, and thus guarantee that there is enough seasonal storage available for the energy transition.

- **Optimal control of seasonal storage to reduce grid imbalances**
  In Chapter 8, we have proposed the first control solutions for seasonal storage

systems that perform market trading. The algorithms are not only designed to maximize profits but also to reduce grid imbalances. To do so, they exploit the win-win opportunities between electricity markets and grid balance. To study the algorithms, we have considered a real seasonal storage tank interacting with several electricity markets and shown that the proposed methods do indeed maximize profits and reduce grid imbalances. In addition, as different algorithms have different merits, we have proposed solutions based on *model predictive control (MPC)* and on *reinforcement learning (RL)*. Then, we have demonstrated that while MPC is better for day-ahead trading, RL is more suitable for imbalance trading. This contribution is important for two reasons. First, it further helps incentivize the use of seasonal storage technologies and ensures that there is enough seasonal storage available for the energy transition. Second, it provides a direct solution to reduce the grid imbalances and in turn to mitigate the negative effects of RES integration.

- **Imbalance market for grid balancing support**
  As traditional power plants are being closed now, RESs should start contributing to balancing the grid if the grid is to remain stable. Yet, the current market framework prevents RESs from participating in the balancing market. With that motivation, in Chapter 10 we have proposed a novel imbalance market that grants access to any type of systems (including RESs) to balancing the grid. The benefits of the new market are that it reduces the cost of balancing the grid and integrates RESs into the portfolio of balancing providers. To illustrate these benefits, we have considered a case study involving seasonal storage systems trading in the new imbalance market. Then, we have shown how the proposed imbalance market can potentially reduce by 10-20% the activated balancing energy in the balancing markets. This contribution is critical to help maintain the grid stability as the increasing integration of RESs creates higher imbalances but RESs cannot participate in the balancing market.

11.2. IMPACT OF THIS RESEARCH ON SOCIETY

BESIDES the technical contributions, the research performed in this PhD dissertation will have multiple positive impacts on society. Namely, it helps to mitigate climate change, to lower the energy bill of end consumers, to reduce the likelihood of energy interruptions and blackouts, and to democratize the access to electricity markets.

CLIMATE CHANGE

The first and most obvious impact is that it helps society to fight climate change and to reduce carbon emissions. Particularly, as part of the plans to address climate change, policy makers are trying to shift the energy sector towards 100% RES generation. In this context, the shift towards green energy poses several unsolved problems. Among them, there is an electrical grid that becomes more unstable or insufficient economic incentives to ensure the widespread use of RESs. In this thesis, we provide solutions for these two problems: we develop approaches that help to reduce grid imbalances and approaches that increase the profits and incentives for RESs.

ENERGY BILL

In the electricity sector, to protect end consumers against price fluctuations, wholesale markets are decouples from retail markets. In particular, end consumers usually enjoy fixed price rates and do not face the price volatility associated with electricity trade. However, as electricity retailers have to absorb all the economic risks, they end up overcharging end consumers to protect themselves. Hence, the electricity price that end consumers pay is usually higher than the actual cost of electricity.

In this context, as RESs have negligible marginal costs, the integration of RESs should in theory lower the energy bill of end consumers. Nonetheless, since the integration of RESs also increases the volatility in prices, retailers face higher risks and end up overcharging end consumers even more.

With the contributions of this PhD dissertation, this issue is partially mitigated. In particular, the forecasting methods proposed in this thesis lower the uncertainty in RES generation. In addition, the proposed forecasting and control algorithms increase the profits of market agents. Together, these contributions lead to retailers with lower economic risks and higher profits, and thus to end consumers that pay less for their energy.

BLACKOUTS

To guarantee the supply of energy, the generation and consumption of electricity should be balanced at all times. In particular, the electrical grid is supposed to operate at a certain fixed frequency, e.g. 50 Hz in Europe. However, when imbalances between the generation and consumption occur, the frequency has to shift to accommodate the lack or excess of power. In this context, when the changes in frequency cross a certain threshold, the grid fails, entire regions experience shortages of energy, and blackouts occur.

To prevent these grid failures, the *transmission system operator (TSO)* monitors the grid and actives energy reserves that correct the grid imbalances. However, as the integration of RESs increases and as traditional power plants are being closed down, imbalances become more severe and TSOs face the risk of not having enough energy reserves. In this situation, the probability of blackouts increases and societies are at risk of experiencing energy shortages.

To help mitigate this issue, this PhD dissertation has two key contributions. First, the forecasting algorithms reduce the uncertainty in grid imbalances and thus the risk of blackouts. Second, the proposed market framework increases the amount of balancing providers, and thus the amount of balancing energy TSOs have at their disposal.

DEMOCRATIZATION OF ELECTRICITY MARKETS

One of the fundamental principles for well-behaved markets is to ensure that market actors do not have strong market power. Particularly, if market actors have power, they can manipulate the market and influence the prices. In the electricity sector, due to the recent liberalization of the market and the large entry barriers, the number of market actors is still limited. As a result, market power can sometimes still be observed in the more restrictive markets, e.g. the balancing market occasionally displays very high prices that can be partially explained by the existing market power.

In this thesis, we aim at correcting this situation and democratize the access to electricity markets. To do so, we provide approaches to improve the economic profits of market actors and to grant newer actors more incentives to participate in the electricity

market. In addition, we provide a new market framework to ensure that even small and uncertain systems can participate in one of the most restrictive electricity markets, i.e. the balancing market.

In short, the research performed in this PhD dissertation has four positive impacts for society. First, it facilitates the energy transition and provides opportunities to revert climate change. Second, it helps society by lowering the energy bill of end consumers and making affordable energy available to a larger set of the population. Third, it lowers the risk of blackouts and energy shortages. Finally, it aids society by democratizing electricity markets, improving the economy and competitiveness of society, and ensuring that greener technologies, i.e. RESs, can enter the marker under fair conditions.

11.3. SUGGESTIONS FOR FUTURE RESEARCH

IN this section, we introduce open problems that still have to be addressed as well as the possible directions for future research.

FORECASTING METHODS

As shown in Part I of the thesis, accurate forecasting methods can help to reduce the uncertainty of RESs and to increase their profitability. In this context, there are still several open problems that require further investigation:

- **New techniques to model market integration**
  In Chapter 3, we have proposed two different techniques to model market integration and we have shown that such techniques were paramount to obtain more accurate forecasts. However, our study was limited to a single market, a single forecasting model, and two different modeling techniques. To further characterize the level of integration, it is necessary to study how important market integration is using other models and considering other markets. In detail, the most important questions that still have to be answered are: (i) Can this effect be observed between countries with markets of similar size[1]? (ii) Can market integration be captured by other forecasting models beyond neural networks, i.e. autoregressive models? (iii) Is this effect observable in other markets besides the day-ahead markets? (iv) Can market integration be modeled with a better technique than the two proposed? To answer the first question, we should first analyze which countries have a large border interconnection. For the second question, a useful analysis could be to study whether both linear and nonlinear models can capture market integration. For the third question, it would be important to quantify the integration of the balancing market and the intraday markets. To answer the fourth question, an interesting topic of research could be to study the use of advanced DL techniques such as transfer learning [229] (see also the next topic of future research).

- **Transfer learning for forecasting electricity prices**
  Considering the success of the proposed DL techniques in Chapter 4, it is interesting to explore more advanced DL techniques that can potentially improve the forecasting accuracy. One of the DL techniques that in recent years has gained a lot

[1] In our case study one market was larger than the other.

of attention is transfer learning [229]. The idea behind transfer learning is to use the knowledge of a learned task to improve a related task that still has to be learned. This is for example very useful where a lot of data exist for an application but not so much for another, e.g. for medical diagnosis tasks where a lot of data exist for a disease but not enough for another one. One of the most common approaches of transfer learning is to use a deep neural network, estimate the model in a dataset, and then transfer the model to another dataset by simply retraining the last layer of the network. In the context of electricity prices, transfer learning could potentially improve the forecasting accuracy via several approaches. The first and more obvious is to transfer knowledge between markets in cases where a forecaster is needed in a market but not enough data exist for that market. A second possibility could be to train a model using data from different markets, and then individually retrain the learned model for each market. As the first model learns features that are common across markets, it could potentially generalize, reduce overfitting, and improve the forecasting accuracy in each individual market.

- **Self-supervised learning for forecasting electricity prices**
  Another DL method that has also become very popular is self-supervised learning [208], which is based on learning auxiliary related tasks to either pre-train the model or to improve training. As with transfer knowledge, self-supervised learning can either be used in cases where the amount of data is not enough or to improve the accuracy of an application. In the context of electricity price forecasting, it would be interesting to answer whether self-supervised learning can improve the forecasting accuracy and to study which auxiliary tasks can be used for it. For the latter, a compelling auxiliary task that could be tried is the prediction of related time series data, e.g. the grid load or weather data.

- **Forecasting techniques for the intraday markets**
  Despite the extensive literature in electricity price forecasting, most of the work has focused on the day-ahead market. While this work has been very important, other markets like the intraday market have been largely ignored. In comparison with the day-ahead market, the intraday markets act as an interactive pool where bids can be submitted and executed: agents send their bids at any time, a bid includes the power to be exchanged and the time of the exchange, and the bid is executed if another agent accepts it. Because of this structure, forecasting models not only have to predict the electricity prices but also the available power in the pool. As a result, the proposed methods for the day-ahead market cannot be applied and new forecasting methods are required. Hence, considering the importance of the intraday market (a lot of RESs are traded on it), future research in the field of electricity price forecasting should aim at developing accurate models for this type of market. Two lines of research that should be prioritized are to identify which factors determine the dynamic of prices and which forecasting models are suitable for such market.

- **Extended generalized model for solar irradiance**
  In Chapter 5, we have proposed a generalized model for solar irradiance that did

not require ground data. While the model successfully improves upon the state-of-the-art literature, the improvements were not large. Therefore, based on the importance that generalized models can have in reducing solar power uncertainty, future research should aim at studying new techniques to build and improve such models. Considering the limited amount of inputs employed for the model that we proposed, a promising line of research is to find other non-local inputs that are relevant for forecasting the irradiance, e.g. other weather data.

- **Scenario generation methods for electricity prices**
  In order to employ forecasting techniques in stochastic settings, e.g. stochastic optimization, it is necessary to go beyond point forecasts and to generate possible scenarios of the variables of interest. While several scenario generations methods for wind forecasting and economic time series have been proposed, there are no methods specifically designed for price forecasting. Considering the uncertainty and volatility of prices and its role in the energy transition, future research should focus on developing accurate scenario generation methods tailored to electricity prices. In this context, a possible line of research is to analyze the different family of copulas [169] and study which copula family is more suitable to characterize prices.

- **Analyze the importance of model inputs versus forecasting model**
  In the context of the energy transition, most of the forecasting research is devoted to propose new methods and compare them with existing ones. However, the inputs of a forecasting model can potentially be more critical than the model itself. Therefore, an important topic of future research is to analyze whether the inputs of forecasting models in the energy domain have a larger impact than the model itself. If the answer is positive, researchers should start spending more time looking at explanatory variables rather than coming up with new forecasting methods.

- **Comparing and deriving properties between scenario generation methods**
  As previously mentioned, scenario generation methods are key for stochastic optimization and control. In this context, there are three different families of methods that exist in the literature[2]: (i) sampling-based method based on stochastic models and forward simulation/sampling [80]; (ii) optimization-based methods that estimate the statistical properties of the scenarios and then solve optimization problems to generate scenarios satisfying these properties [108, 121]; (iii) copula-based methods that use Sklar's theorem [214] to generate scenarios that follow some marginal distributions [183]. Despite the size of the field and the variety of methods, it is still unclear how these methods and their properties compare with each other. In particular, in our experience, it would seem as if each family of methods would belong to a specific field of research and as if researchers in one field would barely refer to the methods in the other. Therefore, considering the importance of scenario generation methods, we believe that it is paramount

[2] This classification between three families has been observed by us but we did not see it anywhere else in the literature. In particular, to the best of our knowledge, no formal classification exists for the different families of scenario generation methods.

that future research derives the differences between the families of methods and studies their relative accuracy.

SEASONAL STORAGE SYSTEMS

In Part II of the thesis, we have proposed algorithms and techniques so that seasonal storage systems could provide energy flexibility, absorb grid imbalances, and help grid regulation. Within this area of research, there are still several research directions that need further study:

- **Low-complexity models for seasonal storage technologies**
  Obtaining models that are accurate yet simple is paramount to incorporate dynamical systems in optimization and control problems. This requirement is particularly important in the context of seasonal storage systems where, due to the long optimization horizons, the optimization problems have to use a large number of variables. In Chapter 8, we have proposed such a model for one of the most widely used seasonal storage systems, i.e. stratified tanks. Nonetheless, there are likely other promising seasonal technologies, e.g. hydrogen storage or borehole thermal energy storage, with complicated dynamics that might also require simplified models for optimal control. Therefore, we believe that an important line of future research to further incentivize the use of seasonal storage could be to develop low-complexity models for other seasonal storage technologies.

- **Stochastic control for seasonal storage with market interaction**
  In Chapter 8, we have proposed control approaches for seasonal storage systems that helped to reduce grid imbalances. Despite their contributions to grid balancing, they were all based on deterministic control strategies and none of them could optimize stochastic metrics, e.g. minimize economic risks. Considering the volatility of electricity prices, minimizing risks is part of the natural behavior of market agents. Therefore, a promising line of future research in the context of seasonal storage is to develop stochastic control solutions. To obtain methods suitable for real-time and methods that can be quickly re-adapted, the research should explore both stochastic MPC solutions and RL approaches.

- **Scenario generation methods for long-term horizons**
  Generating scenarios is paramount to employ forecasts in stochastic settings and, as shown in Chapter 8, to train RL algorithms. In this context, despite long-term scenarios being paramount for seasonal storage systems, the literature on scenario generation for long-term horizons is very scarce. Although in Appendix B.1 we propose a first approach to do so, the derived method is rather simplistic, e.g. the method does not sample scenarios from accurate probability distributions and it is simply based the scenarios on historical data. Therefore, an important line of future research could be to investigate methods to accurately generate scenarios for long-term horizons.

- **Actor-critic and policy gradient algorithms for seasonal storage systems**
  As shown in Chapter 8, RL algorithms might be the only control solutions that are fast enough for real-time control of systems with long-optimization horizons,

e.g. seasonal storage systems. However, all the existing RL approaches for controlling this type of systems are based on Q-learning algorithms, a RL family that is limited to discrete control spaces. Thus, the existing approaches cannot compute the exact optimal solution and do not scale well for large input spaces. In this context, there is a different category of RL algorithms, namely policy gradient and actor critic methods, that are able to work in continuous control spaces. Therefore, considering that seasonal storage systems work with continuous actions, an important line of future research could be to develop actor-critic and policy gradient methods for controlling seasonal storage systems. These new family of algorithms can potentially increase the quality of the optimal solutions, i.e. obtain larger profits, and scale better with larger input spaces.

- **Multi-agent RL algorithms for trading in multiple markets**
  In Chapter 8, we have proposed a simplified multi-agent RL scheme for trading in two markets. Although the approach did consider two RL agents, the agents were trained sequentially and did not have much room for collaboration. As a result, it is possible that this property is limiting the economic profits as the agents cannot take combined actions. Therefore, to fully maximize the economic profits during market trading, an interesting line of future research is to explore collaborative RL algorithms that model agents interacting with each other. A potential difficulty that might appear is the fact that multi-agent RL algorithms have convergence issues as their environment is often non-stationary.

- **Quantify the effect of uncertainties in market trading**
  In Chapter 8, we have proposed multiple control algorithms for electricity trading that made used of price forecasts to maximize the market profits. The motivation behind forecasting prices but not considering other sources of uncertainty was that price uncertainty is usually much larger than any other disturbance and thus the control algorithm can account for most of the uncertainty by simply considering price disturbances. However, as further sources of uncertainty exist, e.g. model mismatches or uncertainty on the initial state of the system, an interesting topic of future research is to consider all sources of uncertainty and validate whether or not the other sources of uncertainty play a role in market trading.

BALANCING MARKETS

In Part III of the thesis, we have investigated possible modifications in the structure of electricity markets so that more energy systems could contribute to grid balancing. In this context, there are still some open problems that require further research:

- **Quantify the total flexibility of the imbalance market**
  In Chapter 10, we have proposed a new market framework for providing balancing support through the imbalance settlement mechanism. Although the proposed framework was shown to reduce the balancing needs of the TSO, we have only quantified the contributions made by seasonal storage systems. In this context, as seasonal storage systems could already reduce the balancing energy by 20%, it is paramount to quantify the full potential of the new imbalance market when

including other RES systems. Such an analysis, albeit very valuable, is a very challenging topic of research for several reasons: (i) it requires very complex models involving all the systems that participate in the market; (ii) for each of these systems, tailored control algorithms are needed to ensure that the grid imbalances are reduced and their profits maximized.

- **Improving trading strategies for the imbalance market**
  Not only was the proposed framework in Chapter 10 limited in terms of market actors, but the trading strategies were also simple. Particularly, market actors were limited to day-ahead trading and did not have many opportunities to acquire energy flexibility. To maximize the benefits of the proposed approach, i.e. to further increase the profit of market agents and the flexibility that they provide, more complex strategies are needed. As an example, it could be worthwhile for seasonal storage systems to buy electricity in the intraday market and then sell it in the imbalance market. Hence, an important area of future research is to study more complex trading strategies (including all electricity markets) for the proposed imbalance market.

FUTURE RESEARCH BEYOND THE SCOPE OF THE THESIS

Going beyond the scope of this thesis, there are other interesting open problems that are important for society:

- **DL methods to forecast complex phenomena**
  In Chapter 4, we have proposed DL methods for electricity price forecasting and showed that the proposed methods could improve upon the state-of-the-art forecasting techniques. One of the likely reasons behind these improvements were the specific properties of electricity prices: (i) highly nonlinear dynamics; (ii) exogenous inputs that are not limited to numerical values but that include categorical inputs; (iii) inputs that need to be processed as sequences; and (iv) inputs that include forecast of other variables. Particularly, as these properties make the dynamics of electricity prices very complex, DL models can arguably exploit this complexity to obtain more accurate predictions than existing forecasting solutions. Hence, an interesting topic for future research could be to extend the proposed DL models to other forecasting areas where the quantity to be forecasted has complex dynamics. A potential area of research could be to forecast traffic conditions; there, the variable of interest is highly nonlinear, depends on categorical and numerical inputs, and can even depend on other forecasts like weather forecasts. Another compelling area could be to forecast the generation and consumption within microgrids; as with traffic conditions, the variable is highly nonlinear and depend on a large variety of factors (including other forecasts).

- **Methods to forecast faults in distribution grids with limited data**
  In Chapter 5, we have developed a forecasting model that could make predictions with limited data. This ability to make predictions with limited data is also very valuable in other areas, e.g. fault detection. In detail, small generating RES units create current and phase voltage imbalances that are nearly impossible to distinguish from regular operating conditions. In turn, these phase imbalances may

lead to faults that are hard to detect and solve. Hence, to safely integrate distributed RES units into distribution grids, it is important to monitor the grid and to proactively manage potential faults and phase imbalances. In this context, existing methods rely on either having multiple sensors along the distribution grid or having enough data to estimate the parameters of the detection models [204, 205]. However, due to the characteristics of distribution grids, sensors and data are very scarce and the applicability of the existing models is rather limited, i.e. they are not designed to work with limited data. Thus, an important topic for future research is to study and develop reliable methods that can anticipate and forecast faults in distribution grids even with limited sensors and data. In this context, a possible solution direction could be to extend the DL model proposed in Chapter 5 so that, instead of generalizing across geographical locations, it generalizes across distribution grids that are within the same area, e.g. the same city. Particularly, the DL model could exploit similarities among neighboring distribution grids to generalize and to make predictions even for grids with limited data.

- **Market integration beyond electricity markets**
  In Chapter 3, we have proposed a forecasting method for electricity prices that exploited market integration in electricity markets. The proposed method, which was based on auxiliary tasks and transfer learning, considered the price dynamics in multiple markets to learn useful features that could be generalized across markets. As we showed, market integration was critical to improve the forecasting accuracy and to obtain reliable forecasts. As market integration is an effect that also appears for other goods and commodities, a very interesting topic for future research is to extend the proposed model to other types of markets. A possible research area could be to use market integration to predict movements in the stock markets: due to globalization, stock markets across the world are highly coupled. Another area that could also benefit from the proposed model is the retail sector. There, forecasting methods are widely employed to predict sales. However, such models are usually based on data from a single company. Hence, an interesting research topic could be to extend the proposed model so that it exploits the integration between the sales of multiple companies.

- **Reinforcement learning versus model predictive control**
  In Chapter 7, we showed that RL techniques are necessary for market trading with long optimization horizon as MPC methods have intractable computation times. This problem of MPC approaches might not be limited to energy systems with long optimization horizons only: as technologies evolve and systems become more complex, real-time control via MPC also becomes more complicated. In this context, although some research areas have already pointed out the computational advantages of RL, e.g. energy management of electrified vehicles [257] or optimization and control of bioprocesses [181], the application of RL is still rather limited and the existing approaches have no stability guarantees. Therefore, an attractive topic for future research is: (i) to investigate potential applications where MPC techniques might become obsolete and (ii) to develop RL solutions with stability guarantees for these type of applications. Arguably, one of the most interesting

areas to perform this research are highly complex networked systems, e.g. transportation networks, where the complexity of the resulting optimization problems is too large. Other areas where this research is also compelling are applications where the real-time computational cost is very critical, e.g. autonomous driving. To develop RL solutions with stability guarantees, a possible area of research could be to improve and extend the few RL algorithms for which stability guarantees have already been established, e.g. safe Lyapunov learning [26]. A second possibility to assess stability could be to perform extensive simulations and experiments to verify whether the developed RL control algorithms are stable.

- **Uncertainty quantification in medical predictions**
  Although in the context of time series forecasting uncertainty quantification is a rather mature area with multiple approaches, e.g. scenario generation (see Appendix B.1) via copulas or probability forecasting via quantile regression, its development in the machine-assisted medical domain is not so [20]. In particular, while artificial intelligence and DL techniques have been widely used to make medical predictions, for e.g. breast cancer [21], tuberculosis [143], or cardiomegaly [215] detection, the uncertainty over these predictions is often unknown. Since errors in medical diagnostics can have terrible consequences, relying on machine-assisted techniques for medical diagnostics will hardly be possible unless tailored uncertainty quantification methods are developed. Hence, a very promising and important area of future research is to develop uncertainty quantification methods for machine-assisted medical applications. A potential solution direction could be to extend the existing methods in time series forecasting, e.g. algorithms based on copulas, to classification problems like disease detection. Another interesting solution could be to develop accurate methods based on Bayesian inference, e.g. Bayesian neural networks, which implicitly model the predictive uncertainty.

- **Debiasing techniques for biases in human-based data**
  In the framework of automated *machine learning (ML)* systems, bias in human-based data is one of the most important issues. In particular, existing datasets often contain more samples of a given biological trait and lead to ML systems that are biased. As an example, we could consider an automated ML system to read CVs and to pre-select a range of candidates applying to a surgeon position. Since the percentage of female surgeons is much lower than that of male surgeons [49], existing datasets to training this ML system will likely have many more male samples than female ones. As a result, the ML automatic system might potentially end up having a positive bias towards male candidates (i.e. male candidates will be more likely to be selected). Besides gender, data can also be biased towards age or towards ethnicity. Therefore, an important topic for future research is to develop novel data debiasing techniques that can lead to unbiased ML systems. A possible approach to tackle this issue could be to develop novel data augmentation techniques to decrease the data imbalance, e.g. the ratio of males/females, by artificially generating data samples of the data classes with fewer samples.

- **Explainable deep learning for critical decision making processes**
  DL techniques in certain domains are often disregarded as potential solution be-

cause of their hard explainability. For instance, in the medical domain, although some explainable techniques tailored to specific diseases have been developed, e.g. acute intracranial haemorrhage [149], most of the existing approaches behave as black box models with an unknown decision making mechanism. Although this is not much of a problem in time series forecasting, e.g. forecasting electricity prices (see Chapter 4), errors in medical diagnosis are too important to neglect this problem. Hence, as with uncertainty quantification, unless explainable techniques are developed, the use of DL in the medical domain will be limited. This issue is not only restricted to medical applications but appears in several domains where understanding the decision making process is very important. For instance, in the context of autonomous driving, it is very important to understand the decision making process to analyze who is at fault in the case of car accidents with human fatalities. Therefore, an interesting topic of future research is to investigate explainable deep learning techniques in areas where insights into the decision making process are needed, e.g. medical diagnosis or autonomous driving. A possible approach to obtain more explainable models could be to use sensitivity analysis to quantify which model inputs are critical in the decision making process.

APPENDICES

A

OPTIMAL HYPERPARAMETERS AND FEATURES FOR FORECASTING MODELS IN ELECTRICITY PRICE BENCHMARK

In this appendix, we present the results of the hyperparameter and feature selection for the benchmark study of Chapter 4. Particularly, Chapter 4 presented a thorough benchmark for forecasting electricity prices that comprised 27 models. As the results of the hyperparameter and feature selection are vast, they are presented here.

A.1. OPTIMAL FEATURE SELECTION

In this section, we present the main results obtained during the feature selection process. In particular, we outline which of the input variables are helpful to predict the different day-ahead prices. The content of this analysis is qualitative, not quantitative; in particular, as we evaluate 27 models, each model predicts 24 hours, and as there are more than 750 individual input features available, the individual results would not only be very vast, but might not provide a very helpful insight.

The section consists of four subsections: first, we present the main feature selection results common to all forecasters. Then, we list the results for the forecasters that require an individual model per hour, e.g. ARIMA or SVR. Third, we present the results of the benchmark forecasters with multiple outputs, e.g. neural networks. Finally, we will provide an overall discussion based on the listed results.

A.1.1. COMMON RESULTS

Independently of the forecaster and the hour, using the day-ahead generation forecast in Belgium, i.e. g_B, decreases the overall accuracy. As discussed in Chapter 3, a possible

explanation for this effect might be a change in the Belgian generation conditions in mid 2015, which would lead, in our study, to different generation conditions between the training and validation datasets and the test dataset.

A.1.2. FORECASTERS WITH AN INDIVIDUAL MODEL PER HOUR

Within the class of forecasters with exogenous inputs and an individual model per hour, we can distinguish between the nonparametric models, i.e. IHMARX and SNARX, and the rest.

In the case of nonparametric models, the optimal features are the same as in the original study: lagged prices at 24, 48, and 168 hours; the minimum price of the previous day; and the day-ahead forecast of the grid load at the prediction hour in the local market, i.e. Belgium. For the rest of the forecasters, we can make a distinction between three groups:

- For the DR, TF, ARX, and TARX models:
  1. Lagged prices in Belgium at 24, 25, 47, 48, 49, 167, 168, 169 hours.
  2. The day-ahead grid load forecast in Belgium at prediction hour.
  3. For ARX and TARX, the minimum price of the previous day.
- For the five machine learning methods, i.e. the three SVR-based models, the RF, and the XGB:
  1. 24 lagged prices of the previous day in Belgium.
  2. 24 lagged prices of one week before in Belgium.
  3. 24 lagged prices of the previous day in France.
  4. The day-ahead grid load forecast in Belgium at prediction hour.
  5. The day-ahead grid load and generation forecasts in France at prediction hour.
- For the three fARX-based forecasters, the optimal features are: the past prices in Belgium and France, the day-ahead load forecast in Belgium and France, and the day-ahead generation forecast in France. For each of these 5 variables, the specific lagged values are the same as in the original paper [232]: a very large combination of past prices at different lags, and the day-ahead forecasts at different future time steps and lags (as it is a total of 107 inputs, we refer to the original paper [232] for full details).

A.1.3. RESULTS FOR FORECASTERS WITH MULTIPLE OUTPUTS

The 6 forecasters that predict the 24 prices in a single model have in common, as optimal features, the 48 inputs represented by the day-ahead generation and load forecast in France. With respect to the rest of the features, there is a division into 2 groups:

- The RBF model, which has as optimal features the day-ahead load forecast in Belgium and considers only lagged prices in Belgium: 48 lagged prices representing the previous day and one week before.

- The DL models, which disregard as optimal features the day-ahead load forecast in Belgium, and consider the same lagged prices in France and Belgium:
  - MLP and DNN: The 72 lagged prices of the previous two days and one week before.
  - CNN: The 168 lagged prices of the week before the day of prediction.
  - LSTM and GRU: The 336 lagged prices of the two weeks before.

A.1.4. DISCUSSION

If we look at the results, we can make the following observations:

1. The local prices in Belgium are the most important quantity. In particular, the lagged prices of the previous two days and one week before are the most important features.
2. For all statistical models, the load forecast for the prediction hour in the local market is also a very important feature.
3. For machine learning models, the features from the neighboring market, i.e. France, are also important. In particular, the lagged prices in France of the previous day and the load and generation forecasts for the prediction hour play an important role.
4. Except for the three fARX-based models, the effect of market integration, i.e. using features from neighboring markets, can only be observed in machine learning models.

A.2. OPTIMAL HYPERPARAMETERS FOR BASE MODELS

IN this section, we describe the hyperparameters that are optimized for each base model and the result of this optimization for our case study. The hyperparameters and optimization results for the machine learning methods are listed in Table A.1; likewise, the optimization results for the statistical methods are listed in A.2. For a more detailed explanation of the meaning of the different hyperparameters, we refer to the original papers. In addition, for the explanation of the hyperparameters of the SVR-based models and the RF model, the library [177] used for the implementation of these models is also a good reference.

Model	Symbol	Value	Definition
SVR	C	9.97	Penalty parameter of the error
	ϵ	0.0038	Epsilon of the epsilon-SVR model
SOM-SVR	C	1.57	Penalty parameter of the error
	ϵ	0.0029	Epsilon of the epsilon-SVR model
	n_c	3	Number of clusters
SVR-ARIMA	C	8.54	Penalty parameter of the error
	ϵ	0.0044	Epsilon of the epsilon-SVR model
	p	4	AR order of ARIMA part
	q	2	*Moving average (MA)* order of ARIMA part
	P	3	AR order of the daily seasonality of the ARIMA part
	Q	1	MA order of the daily seasonality of the ARIMA part
	d	1	Differencing order of ARIMA part
	D	0	Seasonal differencing order of ARIMA part
RF	n_t	470	Number of trees
	p_f	0.49	Percentage of features considered when looking for the best split
	n_{min}	1	Minimum number of samples per leaf node
XGB	n_t	105	Number of trees
	d_{max}	4	Maximum tree depth
	lr	0.0491	Learning rate
	γ	0.0071	Minimum loss reduction needed to make a new partition on a leaf node
	α	8.57	Coefficient for L1 regularization
	λ	0.4273	Coefficient for L2 regularization
	r_{sub}	0.7093	Subsample ratio of the training set used for training a tree
	r_{col}	0.3040	Subsample ratio of columns when training a tree
MLP	n	117	Number of neurons on the hidden layer
	nonlin	ReLU	Activation function on the hidden layer
	d	0	Dropout coefficient
	α	0.00032	Coefficient for L1 regularization
RBF	n	247	Number of neurons, a.k.a. kernels or basis functions.
	cluster	Birch	Clustering algorithm to find the centers for the kernels.

Table A.1: Summary of the optimized hyperparameter for the machine learning models (except the DL models).

fARX	N_{window}	40	Data window: number of past months used for estimating the model.
	N_{mod}	24	$N_{\text{mod}}=1$: one model to predict all 24 hours. $N_{\text{mod}}=24$: individual model per hour.
fARX-Lasso	N_{window}	40	Data window: number of past months used for estimating the model.
	α	0.0040	Coefficient for L1 regularization.
	N_{mod}	24	$N_{\text{mod}}=1$: one model to predict all 24 hours. $N_{\text{mod}}=24$: individual model per hour.
fARX-EN	N_{window}	39	Data window: number of past months used for estimating the model.
	α	0.0010	Coefficient for L1 regularization.
	r	0.95	Elastic net mixing parameter: $r=1$ is equal to Lasso.
	N_{mod}	24	$N_{\text{mod}}=1$: one model to predict all 24 hours. $N_{\text{mod}}=24$: individual model per hour.
IHMARX	N_{window}	32	Data window: number of past months used for estimating the model.
	N_{re}	20	Number of model re-estimations when optimizing the nonparametric model.
	N_{mod}	24	$N_{\text{mod}}=1$: one model to predict all 24 hours. $N_{\text{mod}}=24$: individual model per hour.
SNARX	N_{window}	43	Data window: number of past months used for estimating the model.
	N_{re}	17	Number of model re-estimations when optimizing the nonparametric model.
	N_{mod}	24	$N_{\text{mod}}=1$: one model to predict all 24 hours. $N_{\text{mod}}=24$: individual model per hour.
ARX	N_{window}	All	Data window: number of past months used for estimating the model.
	N_{mod}	24	$N_{\text{mod}}=1$: one model to predict all 24 hours. $N_{\text{mod}}=24$: individual model per hour.
TARX	N_{window}	All	Data window: number of past months used for estimating the model.
	N_{mod}	1	$N_{\text{mod}}=1$: one model to predict all 24 hours. $N_{\text{mod}}=24$: individual model per hour.
DR	N_{window}	36	Data window: number of past months used for estimating the model.
	$\mathbf{l}_{\text{max}}$	168	Largest lag for the demand.
	$\mathbf{p}_{\text{max}}$	192	Largest lag for the prices.
TF	N_{window}	36	Data window: number of past months used for estimating the model.
	$\mathbf{l}_{\text{max}}$	168	Largest lag for the demand.
	$\mathbf{p}_{\text{max}}$	192	Largest lag for the prices.
	$\mathbf{d}_{\text{max}}$	168	Largest lag for the disturbance term.
WARIMA	N_{window}	16	Data window: number of past months used for estimating the model.
WARIMA-RBF	N_{window}	23	Data window: number of past months used for estimating the model.
	N_{swarm}	350	Swarm size.
	N_{max}	150	Maximum number of iterations for particle swarm optimization.
	ω	0.65	Particle velocity scaling factor.
	ϕ_{p}	0.4	Scaling factor to search away from particle's best known position.
	ϕ_{g}	0.7	Scaling factor to search away from swarm's best known position.
ARIMA-GARCH	N_{window}	12	Data window: number of past months used for estimating the model.
DSHW	N_{window}	All	Data window: number of past months used for estimating the model.
TBATS	N_{window}	16	Data window: number of past months used for estimating the model.
DSARIMA	N_{window}	19	Data window: number of past months used for estimating the model.
AR	N_{window}	All	Data window: number of past months used for estimating the model.
	N_{mod}	24	$N_{\text{mod}}=1$: one model to predict all 24 hours. $N_{\text{mod}}=24$: individual model per hour.

Table A.2: Summary of the optimized hyperparameter for the statistical methods.

B

LONG-TERM SCENARIO GENERATION AND IMBALANCE PRICE FORECASTING METHODS

In this appendix, we present the method proposed in Chapter 8 for generating long-term scenarios. In addition, we define the method proposed in the same chapter for forecasting imbalance prices.

B.1. SCENARIO GENERATION

WHEN defining the MPC algorithms, it was assumed that the expected day-ahead prices $\{\hat{p}_k^{\text{dam}}\}_{k=1}^N$, imbalance prices $\{\hat{p}_k^{\text{imb}}\}_{k=1}^N$, heat demand values $\{\hat{Q}_k^{\text{d}}\}_{k=1}^N$, and disturbances $\{\hat{\mathbf{d}}_k\}_{k=1}^N$ were given. In this appendix, the methodology to generate these time series is explained.

B.1.1. MOTIVATION

In order for the MPC to provide good solutions, the expected time series have to be realistic. Therefore, any forecasting method for these time series has to model the time correlation of a single time series and the inter-correlation between the different times series. While there are methods in the literature to create those forecasts, these are limited to short-term horizons with small resolutions, e.g. hourly, or long-term horizons with broad resolutions, e.g. daily, [243]. The main problem of generating forecasts with small resolutions and long horizons is the accuracy: due to the large uncertainty, it is nearly impossible to forecast electricity prices or loads with an hourly resolution one year in advance. Instead of forecasting the expected value, one could generate a set of different scenarios representing possible future realizations. However, as with the literature of forecasts, the field of scenario generation has, to the best of our knowledge, no reliable method to generate long-term scenarios with small resolutions. In particular,

the literature of scenario generation for correlated time series is limited to short-term horizons [84, 156, 183]. In this case, the problem is computational tractability: whether the methods are based on trees [108] or on copulas[84, 156, 183], the computational cost is too large, e.g. in the case of trees the number of scenarios grows exponentially with the horizon [212].

B.1.2. METHOD

In this appendix, we propose a very simple, yet useful, method to generate scenarios of correlated time series for long-term horizons. Then, we use the average of the scenarios at every time point as the expected values used in the MPC. The only two requirements of the proposed method are: (i) to have at least as many historical data as the horizon length of the scenarios; (ii) to have historical data with a resolution equal to or lower than the resolution of the scenarios.

Given a set of n_{ts} historical time series of length n_{h}, i.e. $\{\mathbf{x}_{1,j}\}_{j=1}^{n_{\text{h}}}, \{\mathbf{x}_{2,j}\}_{j=1}^{n_{\text{h}}}, \ldots, \{\mathbf{x}_{n_{\text{ts}},j}\}_{j=1}^{n_{\text{h}}}$, the proposed method generates any number n_{s} of future scenarios of length $N \leq n_{\text{h}}$:

$$\left\{\{\hat{\mathbf{x}}^i_{1,k}\}_{k=n_{\text{h}}+1}^{n_{\text{h}}+N}, \{\hat{\mathbf{x}}^i_{2,k}\}_{k=n_{\text{h}}+1}^{n_{\text{h}}+N}, \ldots, \{\mathbf{x}^i_{n_{\text{ts}},k}\}_{k=n_{\text{h}}+1}^{n_{\text{h}}+N}\right\}_{i=1}^{n_{\text{s}}}, \tag{B.1}$$

In detail, the method consists of 7 steps:

1. Select a representative horizon $N' << N$ so that, given any two time series of length N', any point after N' is uncorrelated with the first point of both time series. In the case of day-ahead market prices, hourly resolution, and a year horizon, we empirically observed $N' = 8$ days to be a good choice as N' includes the weekly and daily seasonalities correlations w.r.t. the first price.

2. Define a subset $\{j_1, j_2, \ldots\}$ of past indices whose associated values are correlated to the expected values at time step $k = 1$. For time series with seasonalities, these indices represent past values at lags equal to multiples of these seasons, e.g. for time series with daily and weekly seasonalities these indices could represent the values 1 day and 1 week in the past. In general, one could use a correlation study to determine the relevant indices.

3. Sample from the historical dataset a subset $\{\mathbf{x}_{1,j}\}_{j=j_i}^{j_i+N'}, \{\mathbf{x}_{2,j}\}_{j=j_i}^{j_i+N'}, \ldots, \{\mathbf{x}_{n_{\text{ts}},j}\}_{j=j_i}^{j_i+N'}$, where the time index j_i is randomly selected from the past indices $\{j_1, j_2, \ldots\}$.

4. Use the previous sample as the first N' points $\{\hat{\mathbf{x}}^1_{1,k}\}_{k=1}^{N'}, \{\hat{\mathbf{x}}^1_{2,k}\}_{k=1}^{N'}, \ldots, \{\mathbf{x}^1_{n_{\text{ts}},k}\}_{k=1}^{N'}$ of the first scenario.

5. Repeat steps 2–4 but defining the subset $\{j_1, j_2, \ldots\}$ as the time indices correlated to the next time point in the scenario, i.e. $k = N' + 1$.

6. Repeat step 5 until a whole scenario is obtained, i.e. N/N' times.

7. Repeat 6 until the n_{s} scenarios are obtained.

It is important to note that, depending on the application, the selection of j_i will vary. For example, in the case of electricity prices and $N' = 1$ week, considering that

prices have yearly, weekly and daily seasonalities, we observed that a good choice for the subset $\{j_1, j_2, \ldots\}$ are time indices representing 1, 2, 50, 51, 52, 53, 54, 102, 103, 104, 105, or 106 weeks in the past, i.e. past time indices that respect the yearly and weekly seasonality (the indices represent the last 2 weeks, last year $\pm$ 2 weeks, and 2 years ago $\pm$ 2 weeks). In a more general setup, one could use a correlation study or a method like k-nearest neighbors [100] to determine the relevant past indices.

B.2. IMBALANCE PRICE FORECAST

As explained in Section 8.4, a forecast of the imbalance price $\hat{p}_1^{\text{imb,acc}}$ at the first time step of the MPC algorithm is needed. In particular, as the MPC algorithm for the imbalance market decides the traded power directly based on $\hat{p}_1^{\text{imb,acc}}$, it is important for $\hat{p}_1^{\text{imb,acc}}$ to be as accurate as possible. While the expected imbalance price $\hat{p}_1^{\text{imb}}$ obtained from the scenario generation method could be used as a forecast, this is not the most accurate prediction as the scenario generation method simply resamples from past data and does necessarily consider the most recent information.

As the literature of electricity price forecasting does not contain, to the best of our knowledge, a method for imbalance price forecasting, in this appendix we propose a first method for it. In detail, the boosting trees model [42] is selected as the forecasting model due to its simplicity and recent success in forecasting day-ahead prices [135]. As input features, the model considers:

- The last n_1 imbalance prices, where n_1 is optimized.
- The last n_2 imbalance volumes, where n_2 is optimized.
- The day-ahead electricity price at the hour of interest.
- The hour of the day and the day of the week.

The hyperparmaters of the boosting tree model, e.g. number of trees, are simultaneously optimized with n_1 and n_2 using the tree-Parzen estimator [25]. The selection of this algorithm to do the feature and hyperparameter selection is motivated by its recent success in other energy applications [133, 135].

As a final remark, it is important to note that, while this is the first method for forecasting imbalance prices, there exist other methods to forecast real-time *local marginal prices (LMPs)* [78, 117, 118]. However, real-time LMPs have different characteristics and represent a different concept than imbalance prices. In particular, the volatility of imbalance prices is larger than real-time LMPs; thus, forecasting imbalance prices is arguably harder than forecasting real-time LMPs (in our experience, forecasting imbalance prices with a horizon larger than one hour is nearly impossible; however, methods for real-time LMPs usually have forecasting horizons up to 6 hours).

LIST OF SYMBOLS

CHAPTER 2

y_k	Variable y at time step k
$\hat{y}_k$	Forecast of variable y at time step k
$\hat{\mathbf{y}}$	Forecast of variable y over multiple time steps
$\hat{\mathbf{y}}_i$	i^{th} multi-step forecast of variable y
$\varepsilon_k^{\hat{\mathbf{y}}_i}$	Error of the forecast $\hat{\mathbf{y}}_i$ at time step k
$L(\varepsilon_k^{\hat{\mathbf{y}}_i})$	Loss function of forecasting error $\varepsilon_k^{\hat{\mathbf{y}}_i}$
$d_k^{\hat{\mathbf{y}}_1,\hat{\mathbf{y}}_2}$	Loss differential function at time step k between forecast $\hat{\mathbf{y}}_1$ and forecast $\hat{\mathbf{y}}_2$
s	Forecasting skill
$\mathbf{x}$	Inputs in a neural network
$\mathbf{z}_k$	State of the k^{th} hidden layer in a neural network
$\mathbf{w}_{\text{i},i}$	Vector of weights between the input layer and neuron i in the first hidden layer
$\mathbf{w}_{k,i}$	Vector of weights from the $(k-1)^{\text{th}}$ hidden layer to neuron i in the k^{th} hidden layer
$\mathbf{w}_{\text{o},i}$	Vector of weights between the last hidden layer and output y_i
$\mathbf{b}_k$	Vector of bias weights of the k^{th} hidden layer
$\mathbf{b}_{\text{o}}$	Vector of bias weights of the output layer
$\mathbf{w}$	Vector containing all the weights of a neural network
n_k	Number of neurons of the k^{th} hidden layer
$f_{k,i}$	Activation function of neuron i in the k^{th} hidden layer
$f_{\text{o},i}$	Activation function of neuron i in the output layer
$\mathcal{S}_{\text{T}}$	Training dataset
N	Dataset size
$F(\cdot)$	Mapping function of a neural network
$g(\cdot)$	Objective function to train a neural network
$\mathbf{c}_k$	Cell state of an LSTM neuron at time step k
$\boldsymbol{\theta}$	Vector of hyperparameters
n_{iter}	Number of iterations during hyperparameter optimization
η	Neural network performance during hyperparameter optimization
$\mathcal{H}$	Set containing pairs $(\eta_i, \boldsymbol{\theta}_i)$ of hyperparameters and neural network performances
$M(\cdot)$	Function that maps hyperparameters to neural network performance

Θ	Hyperparameter space
$\mathcal{Z}$	Hyperparameter set
$\mathcal{U}$	Hyperparameter subset
$\boldsymbol{\theta}_{\mathcal{U}}$	Partial hyperparameter instantiation over subset $\mathcal{U}$
$\hat{m}_{\mathcal{U}}(\cdot)$	Partial performance predictor of the hyperparameter subset $\mathcal{U}$
V	Variance of the neural network performance
$V_{\mathcal{U}}$	Contribution of hyperparameter subset $\mathcal{U}$ to the variance of the neural network performance
$F_{\mathcal{U}}$	Importance of hyperparameter subset $\mathcal{U}$

CHAPTER 3

p_{B}	Price in the EPEX-Belgium day-ahead market
p_{F}	Price in the EPEX-France day-ahead market
l_{B}	Day-ahead forecast of the grid load in Belgium
l_{F}	Day-ahead forecast of the grid load in France
g_{B}	Day-ahead forecast of the generation in Belgium
g_{F}	Day-ahead forecast of the generation in France
h_{B}	Set of dates representing public holidays in Belgium
h_{F}	Set of dates representing public holidays in France
$\mathbf{x}$	Inputs in a neural network
$\mathbf{p}$	Vector of day-ahead prices
$\mathbf{z}_k$	State of the k^{th} hidden layer in a neural network
n_k	Number of neurons of the k^{th} hidden layer in a neural network
$\mathbf{w}_{\mathrm{i},i}$	Vector of weights between the input layer and neuron i in the first hidden layer
$\mathbf{w}_{k,i}$	Vector of weights from the $(k-1)^{\text{th}}$ hidden layer to neuron i in the k^{th} hidden layer
$\mathbf{w}_{\mathrm{o},i}$	Vector of weights between the last hidden layer and output p_i
$\mathbf{b}_k$	Vector of bias weights of the k^{th} hidden layer
$\mathbf{b}_{\mathrm{o}}$	Vector of bias weights of the output layer
$\mathbf{w}$	Vector containing all the weights of a neural network
$f_{k,i}$	Activation function of neuron i in the k^{th} hidden layer
$f_{\mathrm{o},i}$	Activation function of neuron i in the output layer
$\mathcal{S}_{\mathrm{T}}$	Training dataset
N	Dataset size
$F(\cdot)$	Mapping function of a neural network
$\boldsymbol{\theta}$	Vector of hyperparameters
$\boldsymbol{\theta}_{\mathcal{B}}$	Hyperparameters that select the binary input features

$n_{\mathcal{B}}$	Number of binary input features
$\Theta_{\mathcal{B}}$	Hyperparameter space of the binary input features
$\mathcal{B}$	Hyperparameter subset representing the binary input features
$\boldsymbol{\theta}_{\mathcal{I}}$	Hyperparameters that select the integer input features
$n_{\mathcal{I}}$	Number of integer input features
$\Theta_{\mathcal{I}}$	Hyperparameter space of the integer input features
$\mathcal{I}$	Hyperparameter subset representing the integer input features
$\mathcal{Z}$	Hyperparameter set
Θ	Hyperparameter space
n_{iter}	Number of iterations during hyperparameter optimization
η	Neural network performance during hyperparameter optimization
$\mathcal{H}$	Set containing pairs $(\eta_k, \boldsymbol{\theta}_k)$ of hyperparameters and neural network performances
Θ_i	Hyperparameter space of feature i
θ_i	Instantiation $\theta_i \in \Theta_i$ of feature i
$\hat{m}_i(\cdot)$	Partial performance predictor of feature i
V	Variance of the neural network performance over the space of input features
V_i	Contribution of feature i to the variance of the neural network performance
$V_{i,j}$	Contribution of interaction between features i and j to the variance of the neural network performance
F_i	Importance of input feature i
$F_{i,j}$	Importance of pairwise interaction between features i and j
ϵ	Importance threshold to select optimal features
$\mathcal{U}_1^*$	Subset of features whose importance is larger than ϵ
$\mathcal{U}_2^*$	Subset of features that have at least a pairwise interaction with an importance larger than ϵ
$\mathcal{U}^*$	Subset of optimal input features
$\mathcal{U}_{\text{B}}^*$	Subset of optimal binary input features
$\mathcal{U}_{\text{I}}^*$	Subset of optimal integer input features
x_h^{C}	Input feature x in country C at day-ahead hour h, where x can either be the price p, grid load l, or generation g, and where C is either B (Belgium) or F (France)
θ_x	Hyperparameter to select input feature x, where x is either p_{F}, g_{B}, g_{F}, H_{F}, H_{F}, l_{F}, or l_{F}
$\theta_{p_{\text{B}}}^{\text{d}}$	Hyperparameter to select daily lags of Belgian prices
$\theta_{p_{\text{B}}}^{\text{w}}$	Hyperparameter to select weekly lags of Belgian prices
$\mathbf{x}_{p_{\text{B},h}}^{\text{d}}$	Vector of lagged Belgian prices selected by $\theta_{p_{\text{B}}}^{\text{d}}$
$\mathbf{x}_{p_{\text{B},h}}^{\text{w}}$	Vector of lagged Belgian prices selected by $\theta_{p_{\text{B}}}^{\text{w}}$
$d_k^{M_1,M_2}$	k^{th} loss differential between the forecasts of model M_1 and M_2

$d_{k,h}^{M_1,M_2}$	k^{th} loss differential between the forecasts of model M_1 and M_2 at hour h
M_{FR}	Single market forecaster that includes all the optimal input features
M_{NoFR}	Single market forecaster that includes all the optimal input features except from the input features related to the EPEX-France market
$M_{\text{FR},l_{\text{B}}}$	Single market forecaster that includes all the optimal input features plus the grid load in Belgium
$M_{\text{NoFR},l_{\text{B}}}$	Single market forecaster that disregards all the input features related to the EPEX-France but considers the grid load in Belgium
M_{Single}	Optimal single market forecaster
M_{Dual}	Optimal dual market forecaster

CHAPTER 4

$\mathbf{p}^{\text{L}}$	Prices in the local day-ahead market
$\mathbf{p}^{\text{M}_i}$	Prices in the neighboring day-ahead market M_i
$\mathbf{x}$	Inputs of the feedforward neural network
$\mathbf{z}_k$	State of the k^{th} hidden layer in the feedforward neural network
n_k	Number of neurons of the k^{th} hidden layer in the forward neural network
$\mathbf{x}^{\text{F}}$	Inputs representing future information in the hybrid LSTM/GRU models
$\mathbf{x}_i^{\text{P}}$	Input sequence i representing past information in the hybrid LSTM/GRU and the convolutional network models
n_{F}	Number of input features that represent future information in the hybrid LSTM/GRU models
q	Number of input sequences that represent past information in the hybrid LSTM/GRU and convolutional network models
n_{P}	Number of input features in each of the q input sequences that represent past information
n_{D}	Number of neurons in the feedforward layer of the hybrid LSTM/GRU models
n_{R}	Number of neurons in the recurrent layer of the hybrid LSTM/GRU models
z_i^{F}	State of the i^{th} neuron of the feedforward layer of the hybrid LSTM/GRU models
z_i^{P}	Hidden state of the i^{th} neuron of the recurrent layer of the hybrid LSTM/GRU models
c_i^{P}	Cell state of the i^{th} neuron of the recurrent layer of the hybrid LSTM model
$\mathbf{x}_i^{\text{F}}$	Input sequence i representing future information in the convolutional network model
r	Number of input sequences representing future information in the convolutional network model
$z_{j,k}^{\text{F,i}}$	Internal state of the k^{th} neuron in feature map j and layer i of the convolutional network that processes future information

$z_{j,k}^{\mathrm{P,i}}$	Internal state of the k^{th} neuron in feature map j and layer i of the convolutional network that processes past information
n_{DNN}	Hyperparameter that selects the number of neurons in the feedforward layer of the hybrid LSTM/GRU models
n_{LSTM}	Hyperparameter that selects the number of neurons in the recurrent layer of the hybrid LSTM models
n_{GRU}	Hyperparameter that selects the number of neurons in the recurrent layer of the hybrid GRU models
λ	Weight factor for the L_1-norm penalization
$\mathbf{w}$	Vector of weights in a neural network
$\mathcal{S}_{\mathrm{T}}$	Training dataset
N	Dataset size
$F(\cdot)$	Mapping function of a neural network
$d_k^{M_1,M_2}$	k^{th} loss differential between the forecasts of model M_1 and M_2
$d_{k,h}^{M_1,M_2}$	k^{th} loss differential between the forecasts of model M_1 and M_2 at hour h
DM_h	Diebold-Mariano test at hour h
$\hat{\mathrm{DM}}_h$	Complementary Diebold-Mariano test at hour h
$\mathrm{DM}_{\mathrm{sc}}$	Diebold-Mariano test at all hours considering serial correlation

CHAPTER 5

I	Solar irradiance
I_h	Measured irradiance at hour h
$\hat{I}_{h+i}$	Forecast of the solar irradiance at hour $h+i$ made at hour h
$\hat{\mathbf{I}}_h$	Multi-step forecast of the solar irradiance made at hour h
$\hat{\mathbf{I}}_h^{\mathrm{E}}$	Multi-step forecast of the solar irradiance made by the ECMWF model and used as input for the neural network model
$\hat{I}_{h+i}^{\mathrm{E}}$	Forecast of the solar irradiance at hour $h+i$ made by the ECMWF model
$\mathbf{I}_h^{\mathrm{c}}$	Vector of clear-sky irradiance values used as input for the neural network model
I_{h+i}^{c}	Clear-sky irradiance at hour $h+i$
$\mathbf{I}_h^{\mathrm{S}}$	Vector of satellite-based irradiance measurements used as input for the neural network model
I_{h+i}^{S}	Satellite-based irradiance measurement at hour $h+i$
$\mathbf{I}_h^{\mathrm{G}}$	Vector of ground irradiance measurements used as input for the local models
k_h^{c}	Clear-sky index at hour h
$\mathbf{x}$	Inputs in the neural network model
$\mathbf{z}_k$	State of the k^{th} hidden layer in the neural network model
n_k	Number of neurons of the k^{th} hidden layer in the neural network model

$\mathbf{w}_{\mathrm{i},i}$	Vector of weights between the input layer and neuron i in the first hidden layer
$\mathbf{w}_{k,i}$	Vector of weights from the $(k-1)^{\text{th}}$ hidden layer to neuron i in the k^{th} hidden layer
$\mathbf{w}_{\mathrm{o},i}$	Vector of weights between the last hidden layer and output $\hat{I}_{h+i}$
$\mathbf{b}_k$	Vector of bias weights of the k^{th} hidden layer
$\mathbf{b}_{\mathrm{o}}$	Vector of bias weights of the output layer
$\mathbf{w}$	Vector containing all the weights of a neural network
$f_{k,i}$	Activation function of neuron i in the k^{th} hidden layer
$f_{\mathrm{o},i}$	Activation function of neuron i in the output layer
$\mathcal{S}_{\mathrm{T}}$	Training dataset
N	Dataset size
$F(\cdot)$	Mapping function of the neural network model

CHAPTER 6

n_{L}	Number of discrete layers used in the dynamical model of a stratified tank
i	Index associated with layer i of the dynamical model of a stratified tank
T_i	Temperature of layer i in the model of a stratified tank
$\dot{Q}_i$	Net heat added/extracted from layer i
$\dot{m}_i$	Net flow added/extracted from layer i
T_i^{in}	Temperature of the flow in layer i
A_i	Cross-sectional area of layer i
P_i	Perimeter of layer i
Δz_i	Thickness of layer i
κ_i	Thermal conductivity of the isolation wall in layer i
z	Direction of stratification in the stratified tank
α	Diffusivity of the fluid in the stratified tank
ρ	Density of the fluid in the stratified tank
c_{p}	Specific heat of the fluid in the stratified tank
T_∞	Ambient temperature
V_i	Volume of layer i
Δt	Simulation time interval
N	Simulation horizon
$\mathbf{x}$	State of a general dynamic system
$\mathbf{u}$	Control input of a general dynamic system
k	Discrete time index, e.g. $\mathbf{x}_k$ represents the state at time step k
$\mathbf{u}^\star$	Optimal control input
$\mathbf{s}$	State of an RL agent

$\pi^\star(\cdot)$	Optimal policy of an RL agent
$\mathcal{U}$	Discrete set of possible actions
n_u	Number of actions in the action set $\mathcal{U}$
r	Reward obtained by an RL agent after taking an action
$p(\cdot)$	Probability distribution of the dynamics of an RL agent
$q(\cdot)$	Probability distribution of the reward of an RL agent
$\mathcal{M}$	Memory dataset used by an RL agent to store exploration data
n_steps	Number of steps between each data dump in the memory $\mathcal{M}$
R	Cumulative reward received by an RL agent
γ	Discount factor of the cumulative reward
T_e	Episode length in an RL algorithm
ϵ	Probability of an RL agent taking a random action
$Q(\cdot)$	Q-function in the Q-learning algorithm

CHAPTER 7

n_L	Number of discrete layers used in the dynamical model of an STESS
i	Index associated with layer i of the dynamical model of an STESS
T_i	Temperature of layer i in the STESS model
$\dot{Q}_i$	Net heat added/extracted from layer i
$\dot{m}_i$	Net flow added/extracted from layer i
T_i^in	Temperature of the flow in layer i
A_i	Cross-sectional area of layer i
P_i	Perimeter of layer i
Δz_i	Thickness of layer i
κ_i	Thermal conductivity of the isolation wall in layer i
z	Direction of stratification in the STESS
α	Diffusivity of the fluid in the STESS
ρ	Density of the fluid in the STESS
c_p	Specific heat of the fluid in the STESS
T_∞	Ambient temperature
$\mathbf{T}$	Temperature distribution in the STESS
k	Discrete time index
$y_{k,i}$	Variable y in layer i at time step k, where y can be any of the layer-specific variables, e.g. temperature T or input heat $\dot{Q}$
Δt	Time interval
β_i	Parameter of the STESS model that equals $\frac{P_i \kappa}{\rho c_\mathrm{p} A_i}$
λ_i	Parameter of the STESS model that equals $\frac{1}{\rho c_\mathrm{p} A_i}$

ϕ	Parameter of the STESS model that equals $\frac{1}{\rho A_i}$
V_i	Volume of layer i
$\theta_{i+1,i}$	Ratio between the volume of layer i and the sum of volumes of layers i and $i+1$
μ	Scaling factor in the *log-sum-exp* approximation of the *max* function
$H(\cdot)$	Heaviside step function
$S(\cdot)$	Logistic function
σ	Scaling factor in the logistic function
$\dot{Q}'_i$	Approximation of the net heat in layer i considering buoyancy
$\dot{m}'_i$	Approximation of the net flow in layer i considering buoyancy
$\boldsymbol{\varphi}$	Vector of model parameters in the parameter estimation problem
$G_i(\cdot)$	Model of the discrete dynamics of layer i
$\bar{\mathbf{T}}$	Vector of temperature measurements
$\mathcal{S}$	Set of discrete time indices for which measurements exist
N	Number of discrete time points in the optimization problems
$\hat{Q}^{\mathrm{d}}$	Heat demand
p	Market price
$\bar{\mathbf{T}}_1$	Estimation of the initial temperature distribution
$\mathbf{G}(\cdot)$	Vector function that models the discrete dynamics
$\dot{\mathbf{Q}}^{\mathrm{in}}$	Vector representing the input heat in the STESS
$\dot{\mathbf{Q}}^{\mathrm{out}}$	Vector representing the output heat in the STESS
j	Index associated with the heat buffer j of the STESS
$y_{k,j}$	Variable y in buffer j at time step k, where y can be any of the buffer-specific variables, e.g. temperature T or input heat $\dot{Q}^{\mathrm{in}}$
γ	Penalty coefficient in the periodical penalty cost of the optimal control problem
$\mathcal{L}_j$	Set of indices that represent the layers that are part of heat buffer j
$n_{\mathrm{L}\,j}$	Number of layers in heat buffer j
$T_{\max}$	Maximum temperature in the STESS
$\dot{m}_{\max}$	Maximum water flow through the heat exchangers of the STESS
T_{d}	Minimum input temperature of the discharging heat exchanger
T_{c}	Maximum input temperature of the charging heat exchanger
k_{he}	Heat exchanger coefficient

CHAPTER 8

$\mathbf{x}$	State of a general dynamic system
$\mathbf{u}$	Control input of a general dynamic system
$\mathbf{d}$	Disturbance of a general dynamic system

t	Continuous time index
k	Discrete time index
h	Discrete hourly time index
i	Index associated with the i^{th} layer in an STESS, e.g. x_i represents the state of layer i and $x_{k,i}$ the state of the same layer at time point k
$\dot{\mathbf{Q}}^{\text{in}}$	Input power in an STESS
$\dot{\mathbf{Q}}^{\text{out}}$	Output power in an STESS
n_{in}	Number of inputs in an STESS
n_{out}	Number of outputs in an STESS
n_{units}	Number of individual storage units in an STESS
$\mathbf{T}$	Temperature distribution in an STESS
z	Direction of stratification in an STESS
$\dot{Q}^{\text{d}}$	Heat demand supplied by an STESS
$\dot{Q}^{\text{m}}$	Allocated power from a general market
$\dot{Q}^{\text{dam}}$	Allocated power from the day-ahead market
$\dot{Q}^{\text{imb}}$	Allocated power from the imbalance market
$\dot{Q}_h^{\text{b}}(\cdot)$	Biding function in the day-ahead market at hour h
p	General price
p^{dam}	Price in the day-ahead market
p^{imb}	Price in the imbalance market
$\dot{Q}_{\bar{p}}^{\text{dam}}$	Optimal market power from the day-ahead market if the price is $\bar{p}$
n_{p}	Number of discrete prices in the bidding functions of the MPC approaches
p^i	i^{th} discrete price in the bidding functions of the MPC approaches
$\hat{p}_h^{\text{dam}}$	Forecast of the day-ahead price at hour h
T	Optimization horizon in the MPC approaches
N	Number of discrete time intervals in the MPC approaches for the day-ahead market
t_k	k^{th} discrete time index in the MPC approaches for the day-ahead market
Δt_k	Time interval at time step k in the MPC approaches for the day-ahead market
$\hat{\dot{Q}}_k^{\text{d}}$	Expected heat demand at time step k
$\hat{p}_k^{\text{dam}}$	Expected day-ahead price at time step k
$\bar{\mathbf{x}}_1$	Known initial state in the MPC approaches
$\mathbf{x}_{\text{min}}$	Lower bound for the state in the MPC approaches
$\mathbf{x}_{\text{max}}$	Upper bound for the state in the MPC approaches
$\mathbf{G}(\cdot)$	Discrete dynamics of an STESS
$\mathbf{g}_{\text{in}}(\cdot)$	Upper limit of the input power in an STESS
$\mathbf{g}_{\text{out}}(\cdot)$	Upper limit of the output power in an STESS
$\dot{Q}_{\text{max}}^{\text{m}}$	Maximum net power that can be purchased by the MPC approaches

N_1	Number of discrete time intervals in the MPC approach for the imbalance market
t'_k	k^{th} discrete time index in the MPC approach for the imbalance market
$\Delta t'_k$	Time interval at time step k in the MPC approach for the imbalance market
t'_{n_d}	Last time index during the first day in the MPC approach for the imbalance market, e.g. if $t'_1 = 21:00\,\text{h}$ and $\Delta t'_k = 1\,\text{h}$ then $t'_{n_\text{d}} = t'_3 = 23:00\,\text{h}$
n_d	Number of discrete time intervals remaining during the first day in the MPC approach for the imbalance market
$\hat{p}^{\text{imb}}_k$	Expected imbalance price at time step k
$\hat{p}^{\text{imb,acc}}_1$	Forecast of the imbalance price at the next time step
$\mathbf{x}^\star_{N_1+1}$	Optimal state at time index t_{N_1+1}
$\mathbf{s}$	State of the RL agent
$\pi^\star(\cdot)$	Optimal policy of the RL agent
$\mathbf{u}^\star$	Optimal action defined by the optimal policy $\pi^\star(\cdot)$
τ	Seasonal time index
n_dis	Number of discretized inputs
$\mathcal{U}$	Discrete set of possible actions
r	Reward obtained by the RL agent after taking an action
T_e	Episode length of the RL algorithm
v^{imb}	Volume of the imbalance
n_hrl	Number of lagged imbalance prices and volumes in the state of the RL agent for the imbalance market
$\dot{Q}^{\text{max}}$	Maximum input power in an STESS as defined by $\mathbf{g}_\text{in}(\cdot)$
$\dot{Q}^{\text{in,dam}}_i$	Power for device i purchased in the day-ahead market
κ	Heat exchanger coefficient
n_ts	Number of time series
n_h	Length of each time series
n_s	Number of generated scenarios

BIBLIOGRAPHY

[1] J. O. Abe, A. P. I. Popoola, E. Ajenifuja and O. Popoola. 'Hydrogen energy, economy and storage: Review and recommendation'. *International Journal of Hydrogen Energy* 44, 15072–15086, 2019. DOI: 10.1016/j.ijhydene.2019.04.068.

[2] O. Abedinia, N. Amjady and H. Zareipour. 'A new feature selection technique for load and price forecast of electrical power systems'. *IEEE Transactions on Power Systems* 32(1), 62–74, 2017. DOI: 10.1109/TPWRS.2016.2556620.

[3] A. Ahmad, T. N. Anderson and T. T. Lie. 'Hourly global solar irradiation forecasting for New Zealand'. *Solar Energy* 122, 1398–1408, 2015. DOI: 10.1016/j.solener.2015.10.055.

[4] A. Akbari-Dibavar, K. Zare and S. Nojavan. 'A hybrid stochastic-robust optimization approach for energy storage arbitrage in day-ahead and real-time markets'. *Sustainable Cities and Society* 49, 101600, 2019. DOI: 10.1016/j.scs.2019.101600.

[5] N. Amjady and M. Hemmati. 'Energy price forecasting - problems and proposals for such predictions'. *IEEE Power and Energy Magazine* 4(2), 20–29, 2006. DOI: 10.1109/MPAE.2006.1597990.

[6] N. Amjady. 'Day-ahead price forecasting of electricity markets by a new fuzzy neural network'. *IEEE Transactions on Power Systems* 21(2), 887–896, 2006. DOI: 10.1109/tpwrs.2006.873409.

[7] N. Amjady, A. Daraeepour and F. Keynia. 'Day-ahead electricity price forecasting by modified relief algorithm and hybrid neural network'. *IET Generation, Transmission & Distribution* 4(3), 432–444, 2010. DOI: 10.1049/iet-gtd.2009.0297.

[8] N. Amjady and F. Keynia. 'Day ahead price forecasting of electricity markets by a mixed data model and hybrid forecast method'. *International Journal of Electrical Power & Energy Systems* 30(9), 533–546, 2008. DOI: 10.1016/j.ijepes.2008.06.001.

[9] N. Amjady and F. Keynia. 'Day-ahead price forecasting of electricity markets by mutual information technique and cascaded neuro-evolutionary algorithm'. *IEEE Transactions on Power Systems* 24(1), 306–318, 2009. DOI: 10.1109/tpwrs.2008.2006997.

[10] S. Anbazhagan and N. Kumarappan. 'Day-ahead deregulated electricity market price forecasting using recurrent neural network'. *IEEE Systems Journal* 7(4), 866–872, 2013. DOI: 10.1109/JSYST.2012.2225733.

[11] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings and M. Diehl. 'CasADi – A software framework for nonlinear optimization and optimal control'. *Mathematical Programming Computation* 11, 1–36, 2018. DOI: 10.1007/s12532-018-0139-4.

[12] R. W. Andrews, J. S. Stein, C. Hansen and D. Riley. 'Introduction to the open source PV LIB for python photovoltaic system modelling package'. In: *Proceedings of the 40th IEEE Photovoltaic Specialist Conference*, 170–174, 2014. DOI: 10.1109/pvsc.2014.6925501.

[13] J. Arteaga and H. Zareipour. 'A price-maker/price-taker model for the operation of battery storage systems in electricity markets'. *IEEE Transactions on Smart Grid* 10(6), 6912–6920, 2019. DOI: 10.1109/tsg.2019.2913818.

[14] B. Baeten, T. Confrey, S. Pecceu, F. Rogiers and L. Helsen. 'A validated model for mixing and buoyancy in stratified hot water storage tanks for use in building energy simulations'. *Applied Energy* 172, 217–229, 2016. DOI: 10.1016/j.apenergy.2016.03.118.

[15] D. Bahdanau, K. Cho and Y. Bengio. *Neural machine translation by jointly learning to align and translate*. 2014. arXiv: 1409.0473 [cs.CL].

[16] R. Baldick. 'Wind and energy markets: A case study of Texas'. *IEEE Systems Journal* 6(1), 27–34, 2012. DOI: 10.1109/JSYST.2011.2162798.

[17] U. Bardi. 'The grand challenge of the energy transition'. *Frontiers in Energy Research* 1, 2, 2013. DOI: 10.3389/fenrg.2013.00002.

[18] C. Batlle. 'Electricity generation and wholesale markets'. In: *Regulation of the Power Sector*, I. J. Pérez-Arriaga (Ed.), 341–395. Springer, 2013. DOI: 10.1007/978-1-4471-5034-3_7.

[19] B. Battke and T. S. Schmidt. 'Cost-efficient demand-pull policies for multi-purpose technologies – The case of stationary electricity storage'. *Applied Energy* 155, 334–348, 2015. DOI: 10.1016/j.apenergy.2015.06.010.

[20] E. Begoli, T. Bhattacharya and D. Kusnezov. 'The need for uncertainty quantification in machine-assisted medical decision making'. *Nature Machine Intelligence* 1(1), 20–23, 2019. DOI: 10.1038/s42256-018-0004-1.

[21] B. E. Bejnordi et al. 'Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer'. *JAMA* 318(22), 2199–2210, 2017. DOI: 10.1001/jama.2017.14585.

[22] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[23] Y. Bengio, P. Simard and P. Frasconi. 'Learning long-term dependencies with gradient descent is difficult'. *IEEE Transactions on Neural Networks* 5(2), 157–166, 1994. DOI: 10.1109/72.279181.

[24] J. Bergstra, D. Yamins and D. D. Cox. 'Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures'. In: *Proceedings of the 30th International Conference on Machine Learning*, 115–123, 2013. URL: https://dl.acm.org/doi/10.5555/3042817.3042832.

[25] J. Bergstra, R. Bardenet, Y. Bengio and B. Kégl. 'Algorithms for hyper-parameter optimization'. In: *Proceedings of the 25th Conference on Neural Information Processing Systems*, 2546–2554, 2011. URL: https://dl.acm.org/doi/10.5555/2986459.2986743.

[26] F. Berkenkamp, M. Turchetta, A. P. Schoellig and A. Krause. 'Safe model-based reinforcement learning with stability guarantees'. In: *Proceedings of the 31st Conference on Neural Information Processing Systems*, 908–919, 2017. URL: https://dl.acm.org/doi/10.5555/3294771.3294858.

[27] H. G. Bock and K.-J. Plitt. 'A multiple shooting algorithm for direct solution of optimal control problems'. *IFAC Proceedings Volumes* 17(2), 1603–1608, 1984. DOI: 10.1016/s1474-6670(17)61205-9.

[28] A. Borovykh, S. Bohte and C. W. Oosterlee. *Conditional time series forecasting with convolutional neural networks*. 2017. arXiv: 1703.04691 [stat.ML].

[29] L. Bottou. 'Large-scale machine learning with stochastic gradient descent'. In: *Proceedings of the 19th International Conference on Computational Statistics*, 177–186, 2010. DOI: 10.1007/978-3-7908-2604-3_16.

[30] G. E. P. Box and D. R. Cox. 'An analysis of transformations'. *Journal of the Royal Statistical Society: Series B (Methodological)* 26(2), 211–252, 1964. DOI: 10.1111/j.2517-6161.1964.tb00553.x.

[31] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. DOI: 10.1017/cbo9780511804441.

[32] C. Brancucci Martinez-Anido, G. Brinkman and B.-M. Hodge. 'The impact of wind power on electricity prices'. *Renewable Energy* 94, 474–487, 2016. DOI: 10.1016/j.renene.2016.03.053.

[33] L. Breiman. 'Random forests'. *Machine Learning* 45(1), 5–32, 2001. DOI: 10.1023/A:1010933404324.

[34] D. W. Bunn and A. Gianfreda. 'Integration and shock transmissions across European electricity forward markets'. *Energy Economics* 32(2), 278–291, 2010. DOI: 10.1016/j.eneco.2009.09.005.

[35] D. Bunn, A. Gianfreda and S. Kermer. 'A trading-based evaluation of density forecasts in a real-time electricity market'. *Energies* 11(10), 2658, 2018. DOI: 10.3390/en11102658.

[36] L. Busoniu, R. Babuška and B. De Schutter. 'A comprehensive survey of multiagent reinforcement learning'. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38(2), 156–172, 2008. DOI: 10.1109/tsmcc.2007.913919.

[37] R. Cao, J. Hart and A. Saavedra. 'Nonparametric maximum likelihood estimators for AR and MA time series'. *Journal of Statistical Computation and Simulation* 73(5), 347–360, 2003. DOI: 10.1080/0094965021000040640.

[38] J. A. Carta, P. Cabrera, J. M. Matías and F. Castellano. 'Comparison of feature selection methods using ANNs in MCP-wind speed methods. A case study'. *Applied Energy* 158, 490–507, 2015. DOI: 10.1016/j.apenergy.2015.08.102.

[39] J. P. S. Catalão, S. J. P. S. Mariano, V. M. F. Mendes and L. A. F. M. Ferreira. 'Short-term electricity prices forecasting in a competitive market: A neural network approach'. *Electric Power Systems Research* 77(10), 1297–1304, 2007. DOI: 10.1016/j.epsr.2006.09.022.

[40] C. Chang, Z. Wu, H. Navarro, C. Li, G. Leng, X. Li, M. Yang, Z. Wang and Y. Ding. 'Comparative study of the transient natural convection in an underground water pit thermal storage'. *Applied Energy* 208, 1162–1173, 2017. DOI: 10.1016/j.apenergy.2017.09.036.

[41] J. Che and J. Wang. 'Short-term electricity prices forecasting based on support vector regression and auto-regressive integrated moving average modeling'. *Energy Conversion and Management* 51(10), 1911–1917, 2010. DOI: 10.1016/j.enconman.2010.02.023.

[42] T. Chen and C. Guestrin. 'Xgboost: A scalable tree boosting system'. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794, 2016. DOI: 10.1145/2939672.2939785.

[43] B. Cheng and W. Powell. 'Co-optimizing battery storage for the frequency regulation and energy arbitrage using multi-scale dynamic programming'. *IEEE Transactions on Smart Grid* 9(3), 1997–2005, 2016. DOI: 10.1109/tsg.2016.2605141.

[44] K. Cho, B. van Merrienboer, D. Bahdanau and Y. Bengio. *On the properties of neural machine translation: Encoder-decoder approaches*. 2014. arXiv: 1409.1259 [cs.CL].

[45] J. W. Choi and D. Aurbach. 'Promise and reality of post-lithium-ion batteries with high energy densities'. *Nature Reviews Materials* 1(4), 16013, 2016. DOI: 10.1038/natrevmats.2016.13.

[46] F. Chollet. *Keras*. Version 2.1.5. 2015. URL: https://keras.io/ (visited on 02/04/2020).

[47] J. Chung, C. Gulcehre, K. Cho and Y. Bengio. *Empirical evaluation of gated recurrent neural networks on sequence modeling*. 2014. arXiv: 1412.3555 [cs.NE].

[48] C. Claus and C. Boutilier. 'The dynamics of reinforcement learning in cooperative multiagent systems'. In: *Proceedings of the 15th AAAI Conference on Artificial Intelligence*, 746–752, 1998. URL: https://dl.acm.org/doi/10.5555/295240.295800.

[49] A. Cochran, T. Hauschild, W. B. Elder, L. A. Neumayer, K. J. Brasel and M. L. Crandall. 'Perceived gender-based barriers to careers in academic surgery'. *The American Journal of Surgery* 206(2), 263–268, 2013. DOI: 10.1016/j.amjsurg.2012.07.044.

[50] I. Coelho, V. Coelho, E. Luz, L. Ochi, F. Guimarães and E. Rios. 'A GPU deep learning metaheuristic based model for time series forecasting'. *Applied Energy* 201, 412–418, 2017. DOI: 10.1016/j.apenergy.2017.01.003.

[51] 'Commission regulation (EU) 2017/2195 of 23 November 2017 establishing a guideline on electricity balancing'. *Official Journal of the European Union* 312, 6–53, 2017. URL: https://data.europa.eu/eli/reg/2017/2195/oj.

[52] A. Conejo, M. Plazas, R. Espínola and A. Molina. 'Day-ahead electricity price forecasting using the wavelet transform and ARIMA models'. *IEEE Transactions on Power Systems* 20(2), 1035–1042, 2005. DOI: 10.1109/TPWRS.2005.846054.

[53] A. J. Conejo, M. Carrión and J. M. Morales. *Decision Making Under Uncertainty in Electricity Markets*. Springer, 2010. DOI: 10.1007/978-1-4419-7421-1.

[54] A. J. Conejo, J. Contreras, R. Espínola and M. A. Plazas. 'Forecasting electricity prices for a day-ahead pool-based electric energy market'. *International Journal of Forecasting* 21(3), 435–462, 2005. DOI: 10.1016/j.ijforecast.2004.12.005.

[55] J. Crespo Cuaresma, J. Hlouskova, S. Kossmeier and M. Obersteiner. 'Forecasting electricity spot-prices using linear univariate time-series models'. *Applied Energy* 77(1), 87–106, 2004. DOI: 10.1016/S0306-2619(03)00096-5.

[56] A. Cruz, A. Muñoz, J. Zamora and R. Espínola. 'The effect of wind generation and weekday on Spanish electricity spot price forecasting'. *Electric Power Systems Research* 81(10), 1924–1935, 2011. DOI: 10.1016/j.epsr.2011.06.002.

[57] G. Darivianakis, A. Eichler, R. S. Smith and J. Lygeros. 'A data-driven stochastic optimization approach to the seasonal storage energy management'. *IEEE Control Systems Letters* 1(2), 394–399, 2017. DOI: 10.1109/lcsys.2017.2714426.

[58] R. De Césaro Oliveski, A. Krenzinger and H. A. Vielmo. 'Comparison between models for the simulation of hot water storage tanks'. *Solar Energy* 75(2), 121–134, 2003. DOI: 10.1016/j.solener.2003.07.009.

[59] I. De Jaeger, J. Lago and D. Saelens. 'A probabilistic building characterization method for district energy simulations'. *Energy and Buildings*, Under review.

[60] I. De Jaeger, J. Lago and D. Saelens. 'A probabilistic approach to allocate building parameters within district energy simulations'. In: *Proceedings of the Urban Energy Simulation Conference*, 2019.

[61] A. M. De Livera, R. J. Hyndman and R. D. Snyder. 'Forecasting time series with complex seasonal patterns using exponential smoothing'. *Journal of the American Statistical Association* 106(496), 1513–1527, 2011. DOI: 10.1198/jasa.2011.tm09771.

[62] F. De Ridder and M. Coomans. 'Grey-box model and identification procedure for domestic thermal storage vessels'. *Applied Thermal Engineering* 67(1), 147–158, 2014. DOI: 10.1016/j.applthermaleng.2014.03.003.

[63] F. De Ridder, M. Diehl, G. Mulder, J. Desmedt and J. Van Bael. 'An optimal control algorithm for borehole thermal energy storage systems'. *Energy and Buildings* 43(10), 2918–2925, 2011. DOI: 10.1016/j.enbuild.2011.07.015.

[64] H. Deneke, A. Feijt and R. Roebeling. 'Estimating surface solar irradiance from METEOSAT SEVIRI-derived cloud properties'. *Remote Sensing of Environment* 112(6), 3131–3141, 2008. DOI: 10.1016/j.rse.2008.03.012.

[65] M. Diagne, M. David, P. Lauret, J. Boland and N. Schmutz. 'Review of solar irradiance forecasting methods and a proposition for small-scale insular grids'. *Renewable and Sustainable Energy Reviews* 27, 65–76, 2013. DOI: 10.1016/j.rser.2013.06.042.

[66] F. X. Diebold and R. S. Mariano. 'Comparing predictive accuracy'. *Journal of Business & Economic Statistics* 13(3), 253–263, 1995. DOI: 10.1080/07350015.1995.10524599.

[67] A. K. Diongue, D. Guégan and B. Vignal. 'Forecasting electricity spot market prices with a k-factor GIGARCH process'. *Applied Energy* 86(4), 505–510, 2009. DOI: 10.1016/j.apenergy.2008.07.005.

[68] 'Directive (EU) 2019/944 of the European Parliament and of the Council of 5 june 2019 on common rules for the internal market for electricity and amending directive 2012/27/EU'. *Official Journal of the European Union* 158, 125–199, 2019. URL: http://data.europa.eu/eli/dir/2019/944/oj.

[69] *Ecovat Seasonal Thermal Energy Storage*. URL: https://www.ecovat.eu (visited on 03/04/2020).

[70] *Energy, transport and environment statistics*. Eurostat, European Commission, 2019. DOI: 10.2785/660147.

[71] *EPEX SPOT market rules - Operational rules*. EPEX SPOT, 2020. URL: https://www.epexspot.com/en/downloads (visited on 31/03/2020).

[72] G. Erbach. *Understanding electricity markets in the EU*. European Parliamentary Research Service, 2016.

[73] D. Ernst, P. Geurts and L. Wehenkel. 'Tree-based batch mode reinforcement learning'. *Journal of Machine Learning Research* 6, 503–556, 2005. URL: https://dl.acm.org/doi/10.5555/1046920.1088690.

[74] *European Centre for Medium-Range Weather Forecasts (ECMWF)*. URL: https://www.ecmwf.int/ (visited on 01/04/2020).

[75] C. Fan, F. Xiao and Y. Zhao. 'A short-term building cooling load prediction method using deep learning algorithms'. *Applied Energy* 195, 222–233, 2017. DOI: 10.1016/j.apenergy.2017.03.064.

[76] S. Fan, C. Mao and L. Chen. 'Next-day electricity-price forecasting using a hybrid network'. *IET Generation, Transmission & Distribution* 1(1), 176–182, 2007. DOI: 10.1049/iet-gtd:20060006.

[77] X. Fang, B.-M. Hodge, L. Bai, H. Cui and F. Li. 'Mean-variance optimization-based energy storage scheduling considering day-ahead and real-time LMP uncertainties'. *IEEE Transactions on Power Systems* 33(6), 7292–7295, 2018. DOI: 10.1109/tpwrs.2018.2852951.

[78] F. Feijoo, W. Silva and T. K. Das. 'A computationally efficient electricity price forecasting model for real time energy markets'. *Energy Conversion and Management* 113, 27–35, 2016. DOI: 10.1016/j.enconman.2016.01.043.

[79] C. Feng, M. Cui, B.-M. Hodge and J. Zhang. 'A data-driven multi-model methodology with deep feature selection for short-term wind forecasting'. *Applied Energy* 190, 1245–1257, 2017. DOI: 10.1016/j.apenergy.2017.01.043.

[80] S.-E. Fleten and T. K. Kristoffersen. 'Stochastic programming for optimizing bidding strategies of a nordic hydropower producer'. *European Journal of Operational Research* 181(2), 916–928, 2007. DOI: 10.1016/j.ejor.2006.08.023.

[81] Frontier Economics. *Overview of European electricity markets*. Directorate-General for Energy, European Commission, 2019. DOI: 10.2833/65488.

[82] R. C. Garcia, J. Contreras, M. Van Akkeren and J. B. C. Garcia. 'A GARCH forecasting model to predict day-ahead electricity prices'. *IEEE Transactions on Power Systems* 20(2), 867–874, 2005. DOI: 10.1109/tpwrs.2005.846044.

[83] A. Ghasemi, H. Shayeghi, M. Moradzadeh and M. Nooshyar. 'A novel hybrid algorithm for electricity price and load forecasting in smart grids with demand-side management'. *Applied Energy* 177, 40–59, 2016. DOI: 10.1016/j.apenergy.2016.05.083.

[84] F. Golestaneh, H. B. Gooi and P. Pinson. 'Generation and evaluation of space–time trajectories of photovoltaic power'. *Applied Energy* 176, 80–91, 2016. DOI: 10.1016/j.apenergy.2016.05.025.

[85] A. Gonzalez-Garrido, A. Saez-de-Ibarra, H. Gaztanaga, A. Milo and P. Eguia. 'Annual optimized bidding and operation strategy in energy and secondary reserve markets for solar plants with storage systems'. *IEEE Transactions on Power Systems* 34(6), 5115–5124, 2019. DOI: 10.1109/tpwrs.2018.2869626.

[86] I. Goodfellow, Y. Bengio and A. Courville. *Deep Learning*. MIT Press, 2016.

[87] A. Graves. *Generating sequences with recurrent neural networks*. 2013. arXiv: 1308.0850 [cs.NE].

[88] R. Green and N. Vasilakos. 'Market behaviour with large amounts of intermittent generation'. *Energy Policy* 38(7), 3211–3220, 2010. DOI: 10.1016/j.enpol.2009.07.038.

[89] W. Greuell, J. Meirink and P. Wang. 'Retrieval and validation of global, direct, and diffuse irradiance derived from SEVIRI satellite observations'. *Journal of Geophysical Research: Atmospheres* 118(5), 2340–2361, 2013. DOI: 10.1002/jgrd.50194.

[90] *Grid data*. RTE. URL: https://data.rte-france.com/ (visited on 01/04/2020).

[91] *Grid data*. Elia. URL: https://www.elia.be/en/grid-data/dashboard (visited on 01/04/2020).

[92] *Grid data*. TenneT. URL: https://www.tennet.org/english/operational_management/index.aspx (visited on 01/04/2020).

[93] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, 2008. DOI: 10.1137/1.9780898717761.

[94] S. Grillo, A. Pievatolo and E. Tironi. 'Optimal storage scheduling using Markov decision processes'. *IEEE Transactions on Sustainable Energy* 7(2), 755–764, 2016. DOI: 10.1109/tste.2015.2497718.

[95] I. Guyon and A. Elisseeff. 'An introduction to variable and feature selection'. *Journal of Machine Learning Research* 3, 1157–1182, 2003. URL: https://dl.acm.org/doi/10.5555/944919.944968.

[96] A. Al-Habaibeh, B. Shakmak and S. Fanshawe. 'Assessment of a novel technology for a stratified hot water energy storage – the water snake'. *Applied Energy* 222, 189–198, 2018. DOI: 10.1016/j.apenergy.2018.04.014.

[97] A. Hammer, D. Heinemann, E. Lorenz and B. Lückehe. 'Short-term forecasting of solar radiation: A statistical approach using satellite data'. *Solar Energy* 67(1), 139–150, 1999. DOI: 10.1016/S0038-092X(00)00038-4.

[98] X. Han, D. Liu, J. Liu and L. Kong. 'Sensitivity analysis of acquisition granularity of photovoltaic output power to capacity configuration of energy storage systems'. *Applied Energy* 203, 794–807, 2017. DOI: 10.1016/j.apenergy.2017.06.062.

[99] Y. M. Han, R. Z. Wang and Y. J. Dai. 'Thermal stratification within the water tank'. *Renewable and Sustainable Energy Reviews* 13(5), 1014–1026, 2009. DOI: 10.1016/j.rser.2008.03.001.

[100] T. Hastie, R. Tibshirani and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001. DOI: 10.1007/978-0-387-21606-5.

[101] W. K. Hastings. 'Monte Carlo sampling methods using Markov chains and their applications'. *Biometrika* 57(1), 97–109, 1970. DOI: 10.2307/2334940.

[102] G. E. Hinton, S. Osindero and Y. W. Teh. 'A fast learning algorithm for deep belief nets'. *Neural Computation* 18(7), 1527–1554, 2006. DOI: 10.1162/neco.2006.18.7.1527.

[103] G. Hinton et al. 'Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups'. *IEEE Signal Processing Magazine* 29(6), 82–97, 2012. DOI: 10.1109/MSP.2012.2205597.

[104] S. Hochreiter and J. Schmidhuber. 'Long short-term memory'. *Neural Computation* 9(8), 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[105] K. O. Homan and S. L. Soo. 'Model of the transient stratified flow into a chilled-water storage tank'. *International Journal of Heat and Mass Transfer* 40(18), 4367–4377, 1997. DOI: 10.1016/S0017-9310(97)00063-X.

[106] Y. Hong and C. Wu. 'Day-ahead electricity price forecasting using a hybrid principal component analysis network'. *Energies* 5(11), 4711–4725. 2012. DOI: 10.3390/en5114711.

[107] K. Hornik, M. Stinchcombe and H. White. 'Multilayer feedforward networks are universal approximators'. *Neural Networks* 2(5), 359–366, 1989. DOI: 10.1016/0893-6080(89)90020-8.

[108] K. Høyland and S. W. Wallace. 'Generating scenario trees for multistage decision problems'. *Management Science* 47(2), 295–307, 2001. DOI: 10.1287/mnsc.47.2.295.9834.

[109] J. Huang, M. Korolkiewicz, M. Agrawal and J. Boland. 'Forecasting solar radiation on an hourly time scale using a coupled autoregressive and dynamical system (CARDS) model'. *Solar Energy* 87, 136–149, 2013. DOI: 10.1016/j.solener.2012.10.012.

[110] F. Hutter, H. H. Hoos and K. Leyton-Brown. 'Sequential model-based optimization for general algorithm configuration'. In: *Learning and Intelligent Optimization*, C. A. C. Coello (Ed.), 507–523. Springer, 2011. DOI: 10.1007/978-3-642-25566-3_40.

[111] F. Hutter, H. Hoos and K. Leyton-Brown. 'An efficient approach for assessing hyperparameter importance'. In: *Proceedings of the 31st International Conference on Machine Learning*, 754–762, 2014. URL: https://dl.acm.org/doi/10.5555/3044805.3044891.

[112] R. J. Hyndman. *forecast: Forecasting functions for time series and linear models*. Version 8.0. 2017. URL: https://pkg.robjhyndman.com/forecast/ (visited on 01/04/2020).

[113] *Imbalance pricing system: How are the (directions of) payment determined?* TenneT, 2019. URL: https://www.tennet.eu/fileadmin/user_upload/SO_NL/ALG_imbalance_pricing_system.doc.pdf (visited on 01/04/2020).

[114] P. Ineichen and R. Perez. 'A new airmass independent formulation for the Linke turbidity coefficient'. *Solar Energy* 73(3), 151–157, 2002. DOI: 10.1016/S0038-092X(02)00045-2.

[115] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver and K. Kavukcuoglu. *Reinforcement learning with unsupervised auxiliary tasks*. 2016. arXiv: 1611.05397 [cs.LG].

[116] T. Jamasb and M. Pollitt. 'Electricity market reform in the European union: Review of progress toward liberalization & integration'. *The Energy Journal* 26, 11–41, 2005. DOI: 10.5547/issn0195-6574-ej-vol26-nosi-2.

[117] Y. Ji, J. Kim, R. J. Thomas and L. Tong. 'Forecasting real-time locational marginal price: A state space approach'. In: *Proceedings of the 2013 Asilomar Conference on Signals, Systems and Computers*, 379–383, 2013. DOI: 10.1109/acssc.2013.6810300.

[118] Y. Ji, R. J. Thomas and L. Tong. 'Probabilistic forecast of real-time LMP via multiparametric programming'. In: *Proceedings of the 48th Hawaii International Conference on System Sciences*, 2549–2556, 2015. DOI: 10.1109/hicss.2015.306.

[119] S. Just and C. Weber. 'Strategic behavior in the german balancing energy mechanism: Incentives, evidence, costs and solutions'. *Journal of Regulatory Economics* 48(2), 218–243, 2015. DOI: 10.1007/s11149-015-9270-6.

[120] S. Karhinen and H. Huuki. 'Private and social benefits of a pumped hydro energy storage with increasing amount of wind power'. *Energy Economics* 81, 942–959, 2019. DOI: 10.1016/j.eneco.2019.05.024.

[121] M. Kaut and S. W. Wallace. 'Evaluation of scenario-generation methods for stochastic programming'. *Pacific Journal of Optimization* 3, 257–271, 2003. DOI: 10.18452/8296.

[122] M. Kazemi, H. Zareipour, N. Amjady, W. D. Rosehart and M. Ehsan. 'Operation scheduling of battery storage systems in joint energy and ancillary services markets'. *IEEE Transactions on Sustainable Energy* 8(4), 1726–1735, 2017. DOI: 10.1109/tste.2017.2706563.

[123] D. Keles, J. Scelle, F. Paraschiv and W. Fichtner. 'Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks'. *Applied Energy* 162, 218–230, 2016. DOI: 10.1016/j.apenergy.2015.09.087.

[124] J. Kennedy. 'Particle swarm optimization'. In: *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb (Eds.), 760–766. Springer, 2011. DOI: 10.1007/978-0-387-30164-8_630.

[125] F. Keynia. 'A new feature selection algorithm and composite neural network for electricity price forecasting'. *Engineering Applications of Artificial Intelligence* 25(8), 1687–1697, 2012. DOI: 10.1016/j.engappai.2011.12.001.

[126] H. Khani, R. K. Varma, M. R. D. Zadeh and A. H. Hajimiragha. 'A real-time multistep optimization-based model for scheduling of storage-based large-scale electricity consumers in a wholesale market'. *IEEE Transactions on Sustainable Energy* 8(2), 836–845, 2017. DOI: 10.1109/tste.2016.2622003.

[127] D. P. Kingma and J. Ba. *Adam: A method for stochastic optimization*. 2014. arXiv: 1412.6980 [cs.LG].

[128] E. M. Kleinbach, W. A. Beckman and S. A. Klein. 'Performance study of one-dimensional models for stratified thermal storage tanks'. *Solar Energy* 50(2), 155–166, 1993. DOI: 10.1016/0038-092X(93)90087-5.

[129] C. Koch and L. Hirth. 'Short-term electricity trading for system balancing: An empirical analysis of the role of intraday trading in balancing germany's electricity system'. *Renewable and Sustainable Energy Reviews* 113, 109275, 2019. DOI: 10.1016/j.rser.2019.109275.

[130] X. Kong, X. Xu, Z. Yan, S. Chen, H. Yang and D. Han. 'Deep learning hybrid method for islanding detection in distributed generation'. *Applied Energy* 210, 776–785, 2018. DOI: 10.1016/j.apenergy.2017.08.014.

[131] A. Krizhevsky, I. Sutskever and G. E. Hinton. 'ImageNet classification with deep convolutional neural networks'. *Communications of the ACM* 60(6), 84–90, 2017. DOI: 10.1145/3065386.

[132] J. Lago, K. De Brabandere, F. De Ridder and B. De Schutter. 'A generalized model for short-term forecasting of solar irradiance'. In: *Proceedings of the 2018 IEEE Conference on Decision and Control*, 3165–3170, 2018. DOI: 10.1109/cdc.2018.8618693.

[133] J. Lago, K. De Brabandere, F. De Ridder and B. De Schutter. 'Short-term forecasting of solar irradiance without local telemetry: A generalized model using satellite data'. *Solar Energy* 173, 566–577, 2018. DOI: 10.1016/j.solener.2018.07.050.

[134] J. Lago, F. De Ridder and B. De Schutter. *Diebold-Mariano test results for "Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms"*. 2017. DOI: 10.4121/UUID:B33D16E9-2ADF-49C8-93BB-3B63A9D0B9E8.

[135] J. Lago, F. De Ridder and B. De Schutter. 'Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms'. *Applied Energy* 221, 386–405, 2018. DOI: 10.1016/j.apenergy.2018.02.069.

[136] J. Lago, F. De Ridder, W. Mazairac and B. De Schutter. 'A 1-dimensional continuous and smooth model for thermally stratified storage tanks including mixing and buoyancy'. *Applied Energy* 248, 640–655, 2019. DOI: 10.1016/j.apenergy.2019.04.139.

[137] J. Lago, F. De Ridder, P. Vrancx and B. De Schutter. 'Forecasting day-ahead electricity prices in Europe: The importance of considering market integration'. *Applied Energy* 211, 890–903, 2018. DOI: 10.1016/j.apenergy.2017.11.098.

[138] J. Lago, M. Erhard and M. Diehl. 'Warping model predictive control for application in control of a real airborne wind energy system'. *Control Engineering Practice* 78, 65–78, 2018. DOI: 10.1016/j.conengprac.2018.06.008.

[139] J. Lago, G. Marcjasz, B. De Schutter and R. Weron. 'Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark'. *Renewable and Sustainable Energy Reviews*, Under Review.

[140] J. Lago, K. Poplavskaya, G. Suryanarayana and B. De Schutter. 'A market framework for grid balancing support through imbalances trading'. *Renewable and Sustainable Energy Reviews*, Under review.

[141] J. Lago, E. Sogancioglu, G. Suryanarayana, F. De Ridder and B. De Schutter. 'Building day-ahead bidding functions for seasonal storage systems: A reinforcement learning approach'. In: *Proceedings of the IFAC Workshop on Control of Smart Grid and Renewable Energy Systems*, 488–493, 2019. DOI: 10.1016/j.ifacol.2019.08.258.

[142] J. Lago, G. Suryanarayana, E. Sogancioglu and B. De Schutter. 'Optimal control strategies for seasonal thermal energy storage systems with market interaction'. *IEEE Transactions on Control Systems Technology*, Early Access, 2020. DOI: 10.1109/TCST.2020.3016077.

[143] P. Lakhani and B. Sundaram. 'Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks'. *Radiology* 284(2), 574–582, 2017. DOI: 10.1148/radiol.2017162326.

[144] V. Lara-Fanego, J. A. Ruiz-Arias, D. Pozo-Vázquez, F. J. Santos-Alamillos and J. Tovar-Pescador. 'Evaluation of the WRF model solar irradiance forecasts in Andalusia (southern Spain)'. *Solar Energy* 86(8), 2200–2217, 2012. DOI: 10.1016/j.solener.2011.02.014.

[145] P. Lauret, C. Voyant, T. Soubdhan, M. David and P. Poggi. 'A benchmarking of machine learning techniques for solar radiation forecasting in an insular context'. *Solar Energy* 112, 446–457, 2015. DOI: 10.1016/j.solener.2014.12.014.

[146] E. W. Law, A. A. Prasad, M. Kay and R. A. Taylor. 'Direct normal irradiance forecasting and its application to concentrated solar thermal output forecasting – A review'. *Solar Energy* 108, 287–307, 2014. DOI: 10.1016/j.solener.2014.07.008.

[147] Y. LeCun, L. Bottou, G. B. Orr and K.-R. Müller. 'Efficient BackProp'. In: *Neural Networks: Tricks of the Trade*, G. B. Orr and K. R. Müller (Eds.), 9–50. Springer, 1998. DOI: 10.1007/3-540-49430-8_2.

[148] A. Lee. *Particle swarm optimization with constraint support*. Version 0.6. 2017. URL: https://pythonhosted.org/pyswarm/ (visited on 04/04/2020).

[149] H. Lee et al. 'An explainable deep-learning algorithm for the detection of acute intracranial haemorrhage from small datasets'. *Nature Biomedical Engineering* 3(3), 173–182, 2018. DOI: 10.1038/s41551-018-0324-9.

[150] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling and J. Wang. 'Deepsaliency: Multi-Task Deep Neural Network model for salient object detection'. *IEEE Transactions on Image Processing* 25(8), 3919–3930, 2016. DOI: 10.1109/TIP.2016.2579306.

[151] W. M. Lin, H. J. Gow and M. T. Tsai. 'An enhanced radial basis function network for short-term electricity price forecasting'. *Applied Energy* 87(10), 3226–3234, 2010. DOI: 10.1016/j.apenergy.2010.04.006.

[152] E. Lindström and F. Regland. 'Modeling extreme dependence between European electricity markets'. *Energy Economics* 34(4), 899–904, 2012. DOI: 10.1016/j.eneco.2012.04.006.

[153] D. C. Liu and J. Nocedal. 'On the limited memory BFGS method for large scale optimization'. *Mathematical Programming* 45, 503–528, 1989. DOI: 10.1007/bf01589116.

[154] Z. Liu, Y. Chen, R. Zhuo and H. Jia. 'Energy storage capacity optimization for autonomy microgrid considering CHP and EV scheduling'. *Applied Energy* 210, 1113–1125, 2018. DOI: 10.1016/j.apenergy.2017.07.002.

[155] E. Lorenz and D. Heinemann. 'Prediction of solar irradiance and photovoltaic power'. In: *Comprehensive Renewable Energy*, A. Sayigh (Ed.), 239–292. Elsevier, 2012. DOI: 10.1016/B978-0-08-087872-0.00114-1.

[156] H. Louie. 'Evaluation of bivariate archimedean and elliptical copulas to model wind power dependency structures'. *Wind Energy* 17(2), 225–240, 2014. DOI: 10.1002/we.1571.

[157] A. Lucas and S. Chondrogiannis. 'Smart grid energy storage controller for frequency regulation and peak shaving, using a vanadium redox flow battery'. *International Journal of Electrical Power & Energy Systems* 80, 26–36, 2016. DOI: 10.1016/j.ijepes.2016.01.025.

[158] S. Makridakis. 'Accuracy measures: Theoretical and practical concerns'. *International Journal of Forecasting* 9(4), 527–529, 1993. DOI: 10.1016/0169-2070(93)90079-3.

[159] G. Marcjasz, J. Lago and R. Weron. 'Neural networks in day-ahead electricity price forecasting: Single vs. multiple outputs'. *Applied Energy*, Under review.

[160] R. Marquez and C. F. M. Coimbra. 'Proposed metric for evaluation of solar forecasting models'. *Journal of Solar Energy Engineering* 135(1), 011016, 2012. DOI: 10.1115/1.4007496.

[161] L. Matignon, G. J. Laurent and N. Le Fort-Piat. 'Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems'. *The Knowledge Engineering Review* 27(1), 1–31, 2012. DOI: 10.1017/s0269888912000057.

[162] T. C. McCandless, S. E. Haupt and G. S. Young. 'A model tree approach to forecasting solar irradiance variability'. *Solar Energy* 120, 514–524, 2015. DOI: 10.1016/j.solener.2015.07.020.

[163] A. Mellit and A. M. Pavan. 'A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy'. *Solar Energy* 84(5), 807–821, 2010. DOI: 10.1016/j.solener.2010.02.006.

[164] L. M. de Menezes and M. A. Houllier. 'Reassessing the integration of European electricity markets: A fractional cointegration analysis'. *Energy Economics* 53, 132–150, 2016. DOI: 10.1016/j.eneco.2014.10.021.

[165] I. Milstein and A. Tishler. 'Can price volatility enhance market power? The case of renewable technologies in competitive electricity markets'. *Resource and Energy Economics* 41, 70–90, 2015. DOI: 10.1016/j.reseneeco.2015.04.001.

[166] F. Nadeem, S. M. S. Hussain, P. K. Tiwari, A. K. Goswami and T. S. Ustun. 'Comparative review of energy storage systems, their roles, and impacts on future power systems'. *IEEE Access* 7, 4555–4585, 2019. DOI: 10.1109/ACCESS.2018.2888497.

[167] V. Nair and G. E. Hinton. 'Rectified linear units improve restricted Boltzmann machines'. In: *Proceedings of the 27th International Conference on Machine Learning*, 807–814, 2010. URL: https://dl.acm.org/doi/10.5555/3104322.3104425.

[168] E. Nasrolahpour, J. Kazempour, H. Zareipour and W. D. Rosehart. 'A bilevel model for participation of a storage system in energy and reserve markets'. *IEEE Transactions on Sustainable Energy* 9(2), 582–598, 2018. DOI: 10.1109/tste.2017.2749434.

[169] R. B. Nelsen. *An Introduction to Copulas*. Springer, 2006. DOI: 10.1007/0-387-28678-0.

[170] D. Niu, D. Liu and D. D. Wu. 'A soft computing system for day-ahead electricity price forecasting'. *Applied Soft Computing* 10(3), 868–875, 2010. DOI: 10.1016/j.asoc.2009.10.004.

[171] F. J. Nogales, J. Contreras, A. J. Conejo and R. Espínola. 'Forecasting next-day electricity prices by time series models'. *IEEE Transactions on Power Systems* 17(2), 342–348, 2002. DOI: 10.1109/MPER.2002.4312063.

[172] S. Nojavan, A. Akbari-Dibavar and K. Zare. 'Optimal energy management of compressed air energy storage in day-ahead and real-time energy markets'. *IET Generation, Transmission & Distribution* 13(16), 3673–3679, 2019. DOI: 10.1049/iet-gtd.2018.7022.

[173] J. Nowotarski, E. Raviv, S. Trück and R. Weron. 'An empirical comparison of alternative schemes for combining electricity spot price forecasts'. *Energy Economics* 46, 395–412, 2014. DOI: 10.1016/j.eneco.2014.07.014.

[174] A. Oudalov, D. Chartouni and C. Ohler. 'Optimizing a battery energy storage system for primary frequency control'. *IEEE Transactions on Power Systems* 22(3), 1259–1266, 2007. DOI: 10.1109/tpwrs.2007.901459.

[175] I. P. Panapakidis and A. S. Dagoumas. 'Day-ahead electricity price forecasting via the application of artificial neural network based models'. *Applied Energy* 172, 132–151, 2016. DOI: 10.1016/j.apenergy.2016.03.089.

[176] P. L. Paradis, D. R. Rousse, L. Lamarche, H. Nesreddine and M. H. Talbot. 'One-dimensional model of a stratified thermal storage tank with supercritical coiled heat exchanger'. *Applied Thermal Engineering* 134, 379–395, 2018. DOI: 10.1016/j.applthermaleng.2018.02.015.

[177] F. Pedregosa et al. *Scikit-learn: Machine Learning in Python*. Version 0.19.1. 2011. URL: https://scikit-learn.org/stable/ (visited on 01/04/2020).

[178] G. Pepermans, J. Driesen, D. Haeseldonckx, R. Belmans and W. D'haeseleer. 'Distributed generation: Definition, benefits and issues'. *Energy Policy* 33(6), 787–798, 2005. DOI: 10.1016/j.enpol.2003.10.004.

[179] R. Perez, S. Kivalov, J. Schlemmer, K. Hemker, D. Renné and T. E. Hoff. 'Validation of short and medium term operational solar radiation forecasts in the US'. *Solar Energy* 84(12), 2161–2172, 2010. DOI: 10.1016/j.solener.2010.08.014.

[180] J. Perner, M. Unteutsch and A. Lövenich. *The future cost of electricity-based synthetic fuels*. Agora Verkehrswende, Agora Energiewende, and Frontier Economics, 2018.

[181] P. Petsagkourakis, I. Sandoval, E. Bradford, D. Zhang and E. del Rio-Chanona. 'Reinforcement learning for batch bioprocess optimization'. *Computers & Chemical Engineering* 133, 106649, 2020. DOI: 10.1016/j.compchemeng.2019.106649.

[182] R. Pino, J. Parreno, A. Gomez and P. Priore. 'Forecasting next-day price of electricity in the Spanish energy market using artificial neural networks'. *Engineering Applications of Artificial Intelligence* 21(1), 53–62, 2008. DOI: 10.1016/j.engappai.2007.02.001.

[183] 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* 12, 51–62, 2009. DOI: 10.1002/we.284.

[184] T. Pippia, J. Lago, R. De Coninck and B. De Schutter. 'Scenario-based nonlinear model predictive control for building heating systems'. *Applied Energy*, Under review.

[185] T. Pippia, J. Lago, R. De Coninck, J. Sijs and B. De Schutter. 'Scenario-based model predictive control approach for heating systems in an office building'. In: *Proceedings of the 15th IEEE International Conference on Automation Science and Engineering*, 1243–1248, 2019. DOI: 10.1109/coase.2019.8842846.

[186] A. Pizzolato, F. Donato, V. Verda and M. Santarelli. 'CFD-based reduced model for the simulation of thermocline thermal energy storage systems'. *Applied Thermal Engineering* 76, 391–399, 2015. DOI: 10.1016/j.applthermaleng.2014.11.029.

[187] M. Pollitt. 'The European single market in electricity: An economic assessment'. *Review of Industrial Organization* 55(1), 63–87, 2019. DOI: 10.1007/s11151-019-09682-w.

[188] K. Poplavskaya and L. De Vries. 'Distributed energy resources and the organized balancing market: A symbiosis yet? Case of three European balancing markets'. *Energy Policy* 126, 264–276, 2019. DOI: 10.1016/j.enpol.2018.11.009.

[189] K. Poplavskaya, J. Lago, S. Strömer and L. de Vries. 'Making the most of short-term flexibility in the balancing market: Opportunities and challenges in the new balancing market design'. *Energy Policy*, Under Review.

[190] K. Poplavskaya, J. Lago and L. de Vries. 'Effect of market design on strategic bidding behavior: Model-based analysis of european electricity balancing markets'. *Applied Energy* 270, 115130, 2020. DOI: 10.1016/j.apenergy.2020.115130.

[191] H. M. I. Pousinho, V. M. F. Mendes and J. P. S. Catalão. 'Short-term electricity prices forecasting in a competitive market by a hybrid PSO–ANFIS approach'. *International Journal of Electrical Power & Energy Systems* 39(1), 29–35, 2012. DOI: 10.1016/j.ijepes.2012.01.001.

[192] *Prequalified providers per balancing product type in the German balancing market*. TenneT, 50hertz, Amprion, Transnet BW, 2019. URL: https://www.regelleistung.net/ext/ (visited on 01/04/2020).

[193] A. Rahman, A. D. Smith and N. Fumo. 'Performance modeling and parametric study of a stratified water thermal storage tank'. *Applied Thermal Engineering* 100, 668–679, 2016. DOI: 10.1016/j.applthermaleng.2016.01.163.

[194] P. Ralon, M. Taylor, A. Ilas, H. Diaz-Bone and K. Kairies. *Electricity storage and renewables: Costs and markets to 2030*. International Renewable Energy Agency, 2017.

[195] J. B. Rawlings, D. Q. Mayne and M. M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017.

[196] G. Reikard. 'Predicting solar radiation at high resolutions: A comparison of time series forecasts'. *Solar Energy* 83(3), 342–349, 2009. DOI: 10.1016/j.solener.2008.08.007.

[197] R. Renaldi, A. Kiprakis and D. Friedrich. 'An optimisation framework for thermal energy storage integration in a residential heat pump heating system'. *Applied Energy* 186, 520–529, 2017. DOI: 10.1016/j.apenergy.2016.02.067.

[198] C. P. Rodriguez and G. J. Anders. 'Energy price forecasting in the Ontario competitive power system market'. *IEEE Transactions on Power Systems* 19(1), 366–374, 2004. DOI: 10.1109/TPWRS.2003.821470.

[199] V. Rostampour, M. Jaxa-Rozen, M. Bloemendal and T. Keviczky. 'Building climate energy management in smart thermal grids via aquifer thermal energy storage systems'. *Energy Procedia* 97, 59–66, 2016. DOI: 10.1016/j.egypro.2016.10.019.

[200] V. Rostampour and T. Keviczky. 'Probabilistic energy management for building climate comfort in smart thermal grids with seasonal storage systems'. *IEEE Transactions on Smart Grid* 10(4), 3687–3697, 2019. DOI: 10.1109/TSG.2018.2834150.

[201] *Royal Netherlands Meteorological Institute (KNMI)*. URL: https://knmi.nl/ (visited on 01/04/2020).

[202] E. Saloux and J. A. Candanedo. 'Control-oriented model of a solar community with seasonal thermal energy storage: Development, calibration and validation'. *Journal of Building Performance Simulation* 12(5), 1–23, 2018. DOI: 10.1080/19401493.2018.1523950.

[203] E. Saloux and J. A. Candanedo. 'Modelling stratified thermal energy storage tanks using an advanced flowrate distribution of the received flow'. *Applied Energy* 241, 34–45, 2019. DOI: 10.1016/j.apenergy.2019.02.075.

[204] N. Sapountzoglou, J. Lago and B. Raison. 'Fault diagnosis in low voltage smart distribution grids using gradient boosting trees'. *Electric Power Systems Research* 182, 106254, 2020. DOI: 10.1016/j.epsr.2020.106254.

[205] N. Sapountzoglou, J. Lago, B. D. Schutter and B. Raison. 'A generalizable and sensor-independent deep learning method for fault detection and location in low-voltage distribution grids'. *Applied Energy* 276, 115299, 2020. DOI: 10.1016/j.apenergy.2020.115299.

[206] I. Sarbu and C. Sebarchievici. 'A comprehensive review of thermal energy storage'. *Sustainability* 10(1), 191, 2018. DOI: 10.3390/su10010191.

[207] O. Schmidt, S. Melchior, A. Hawkes and I. Staffell. 'Projecting the future levelized cost of electricity storage technologies'. *Joule* 3(1), 81–100, 2019. DOI: 10.1016/j.joule.2018.12.008.

[208] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine and G. Brain. 'Time-contrastive networks: Self-supervised learning from video'. In: *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*, 1134–1141, 2018. DOI: 10.1109/icra.2018.8462891.

[209] A. Sfetsos and A. H. Coonick. 'Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques'. *Solar Energy* 68(2), 169–178, 2000. DOI: 10.1016/S0038-092X(99)00064-X.

[210] M. Shafie-Khah, M. P. Moghaddam and M. Sheikh-El-Eslami. 'Price forecasting of day-ahead electricity markets using a hybrid forecast method'. *Energy Conversion and Management* 52(5), 2165–2169, 2011. DOI: 10.1016/j.enconman.2010.10.047.

[211] M. Shahidehpour, H. Yamin and Z. Li. *Market Operations in Electric Power Systems*. John Wiley & Sons, 2002. DOI: 10.1002/047122412X.

[212] A. Shapiro and A. Nemirovski. 'On complexity of stochastic programming problems'. In: *Continuous Optimization*, V. Jeyakumar and A. Rubinov (Eds.), 111–146. Springer, 2005. DOI: 10.1007/0-387-26771-9_4.

[213] V. Sharma and D. Srinivasan. 'A hybrid intelligent model based on recurrent neural networks and excitable dynamics for price prediction in deregulated electricity market'. *Engineering Applications of Artificial Intelligence* 26(5), 1562–1574, 2013. DOI: 10.1016/j.engappai.2012.12.012.

[214] A. Sklar. 'Random variables, joint distribution functions, and copulas'. *Kybernetika* 9(6), 449–460, 1973. URL: https://dml.cz/dmlcz/125838.

[215] E. Sogancioglu, K. Murphy, E. Calli, E. T. Scholten, S. Schalekamp and B. V. Ginneken. 'Cardiomegaly detection on chest radiographs: Segmentation versus classification'. *IEEE Access* 8, 94631–94642, 2020. DOI: 10.1109/access.2020.2995567.

[216] B. K. Sovacool. 'How long will it take? Conceptualizing the temporal dynamics of energy transitions'. *Energy Research & Social Science* 13, 202–215, 2016. DOI: 10.1016/j.erss.2015.12.020.

[217] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov. 'Dropout: A simple way to prevent neural networks from overfitting'. *Journal of Machine Learning Research* 15(56), 1929–1958, 2014. URL: https://dl.acm.org/doi/10.5555/2627435.2670313.

[218] S. Stinner, K. Huchtemann and D. Müller. 'Quantifying the operational flexibility of building energy systems with thermal energy storages'. *Applied Energy* 181, 140–154, 2016. DOI: 10.1016/j.apenergy.2016.08.055.

[219] S. Stoft. *Power System Economics: Designing Markets for Electricity*. Wiley-IEEE Press, 2002. DOI: 10.1109/9780470545584.

[220] G. Suryanarayana, J. Arroyo, L. Helsen and J. Lago. 'A data driven method for optimal sensor placement in multi-zone buildings'. *Energy and Buildings*, Under review.

[221] G. Suryanarayana, J. Lago, D. Geysen, P. Aleksiejuk and C. Johansson. 'Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods'. *Energy* 157, 141–149, 2018. DOI: 10.1016/j.energy.2018.05.111.

[222] I. Sutskever, O. Vinyals and Q. V. Le. 'Sequence to sequence learning with neural networks'. In: *Proceedings of the 28th Conference on Neural Information Processing Systems*, 3104–3112, 2014. URL: https://dl.acm.org/doi/10.5555/2969033.2969173.

[223] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

[224] B. Szkuta, L. Sanabria and T. Dillon. 'Electricity price short-term forecasting using artificial neural networks'. *IEEE Transactions on Power Systems* 14(3), 851–857, 1999. DOI: 10.1109/59.780895.

[225] Z. Tan, J. Zhang, J. Wang and J. Xu. 'Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models'. *Applied Energy* 87(11), 3606–3610, 2010. DOI: 10.1016/j.apenergy.2010.05.012.

[226] J. W. Taylor. 'Short-term electricity demand forecasting using double seasonal exponential smoothing'. *Journal of Operational Research Society* 54, 799–805, 2003. DOI: 10.1057/palgrave.jors.2601589.

[227] Theano Development Team. *Theano: A Python framework for fast computation of mathematical expressions*. 2016. arXiv: 1605.02688 [cs.SC].

[228] R. Tibshirani. 'Regression shrinkage and selection via the lasso'. *Journal of the Royal Statistical Society: Series B (Methodological)*, 267–288, 1996. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[229] L. Torrey and J. Shavlik. 'Transfer learning'. In: *Handbook of Research on Machine Learning Applications and Trends*, E. Soria-Olivas, J. D. Martín-Guerrero, M. Martinez-Sober, J. R. Magdalena-Benedito and A. J. Serrano-López (Eds.), 242–264. IGI Global, 2010. DOI: 10.4018/978-1-60566-766-9.ch011.

[230] *Transparency platform*. ENTSO-E. URL: https://transparency.entsoe.eu/ (visited on 01/04/2020).

[231] S. Ugarte et al. *Energy storage: Which market designs and regulatory incentives are needed?* European Parliament's Committee on Industry, Research and Energy, 2015. DOI: 10.2861/234447.

[232] B. Uniejewski, J. Nowotarski and R. Weron. 'Automated variable selection and shrinkage for day-ahead electricity price forecasting'. *Energies* 9(8), 621, 2016. DOI: 10.3390/en9080621.

[233] V. Vahidinasab, S. Jadid and A. Kazemi. 'Day-ahead price forecasting in restructured power systems using artificial neural networks'. *Electric Power Systems Research* 78(8), 1332–1342, 2008. DOI: 10.1016/j.epsr.2007.12.001.

[234] R. A. van der Veen, A. Abbasy and R. A. Hakvoort. 'Agent-based analysis of the impact of the imbalance pricing mechanism on market behavior in electricity balancing markets'. *Energy Economics* 34(4), 874–881, 2012. DOI: 10.1016/j.eneco.2012.04.001.

[235] R. A. van der Veen and R. A. Hakvoort. 'The electricity balancing market: Exploring the design challenge'. *Utilities Policy* 43, 186–194, 2016. DOI: 10.1016/j.jup.2016.10.008.

[236] C. Voyant, G. Notton, S. Kalogirou, M.-L. Nivet, C. Paoli, F. Motte and A. Fouilloy. 'Machine learning methods for solar radiation forecasting: A review'. *Renewable Energy* 105, 569–582, 2017. DOI: 10.1016/j.renene.2016.12.095.

[237] A. Wächter and L. T. Biegler. 'On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming'. *Mathematical Programming* 106(1), 25–57, 2006. DOI: 10.1007/s10107-004-0559-y.

[238] M. Walter, M. V. Kovalenko and K. V. Kravchyk. 'Challenges and benefits of post-lithium-ion batteries'. *New Journal of Chemistry* 44(5), 2020. DOI: 10.1039/c9nj05682c.

[239] D. Wang, H. Luo, O. Grunder, Y. Lin and H. Guo. 'Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and BP neural network optimized by firefly algorithm'. *Applied Energy* 190, 390–407, 2017. DOI: 10.1016/j.apenergy.2016.12.134.

[240] H. Wang, G. Li, G. Wang, J. Peng, H. Jiang and Y. Liu. 'Deep learning based ensemble approach for probabilistic wind power forecasting'. *Applied Energy* 188, 56–70, 2017. DOI: 10.1016/j.apenergy.2016.11.111.

[241] H. Wang, G. Wang, G. Li, J. Peng and Y. Liu. 'Deep belief network based deterministic and probabilistic wind speed forecasting approach'. *Applied Energy* 182, 80–93, 2016. DOI: 10.1016/j.apenergy.2016.08.108.

[242] W. Wei, C. Gu, D. Huo, S. Le Blond and X. Yan. 'Optimal borehole energy storage charging strategy in a low carbon space heat system'. *IEEE Access* 6, 76176–76186, 2018. DOI: 10.1109/access.2018.2883798.

[243] R. Weron. 'Electricity price forecasting: A review of the state-of-the-art with a look into the future'. *International Journal of Forecasting* 30(4), 1030–1081, 2014. DOI: 10.1016/j.ijforecast.2014.08.008.

[244] R. Weron and A. Misiorek. 'Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models'. *International Journal of Forecasting* 24(4), 744–763, 2008. DOI: 10.1016/j.ijforecast.2008.08.004.

[245] L. Xiao, W. Shao, M. Yu, J. Ma and C. Jin. 'Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system forecasting'. *Applied Energy* 198, 203–222, 2017. DOI: 10.1016/j.apenergy.2017.04.039.

[246] S. Xie, H. He and J. Peng. 'An energy management strategy based on stochastic model predictive control for plug-in hybrid electric buses'. *Applied Energy* 196, 279–288, 2017. DOI: 10.1016/j.apenergy.2016.12.112.

[247] J. Xu, R. Wang and Y. Li. 'A review of available technologies for seasonal thermal energy storage'. *Solar Energy* 103, 610–638, 2014. DOI: 10.1016/j.solener.2013.06.006.

[248] Q. Xu and S. Dubljevic. 'Model predictive control of solar thermal system with borehole seasonal storage'. *Computers & Chemical Engineering* 101, 59–72, 2017. DOI: 10.1016/j.compchemeng.2017.02.023.

[249] W. Yaïci, M. Ghorab, E. Entchev and S. Hayden. 'Three-dimensional unsteady CFD simulations of a thermal storage tank performance for optimum design'. *Applied Thermal Engineering* 60(1), 152–163, 2013. DOI: 10.1016/j.applthermaleng.2013.07.001.

[250] Y. Yan, C. Zhang, K. Li and Z. Wang. 'An integrated design for hybrid combined cooling, heating and power system with compressed air energy storage'. *Applied Energy* 210, 1151–1166, 2018. DOI: 10.1016/j.apenergy.2017.07.005.

[251] D. Yang, Z. Ye, L. H. I. Lim and Z. Dong. 'Very short term irradiance forecasting using the LASSO'. *Solar Energy* 114, 314–326, 2015. DOI: 10.1016/j.solener.2015.01.016.

[252] Z. Yang, L. Ce and L. Lian. 'Electricity price forecasting by a hybrid model, combining wavelet transform, ARMA and kernel-based extreme learning machine methods'. *Applied Energy* 190, 291–305, 2017. DOI: 10.1016/j.apenergy.2016.12.130.

[253] Y. Yao, L. Rosasco and A. Caponnetto. 'On early stopping in gradient descent learning'. *Constructive Approximation* 26(2), 289–315, 2007. DOI: 10.1007/s00365-006-0663-2.

[254] J. Yosinski, J. Clune, Y. Bengio and H. Lipson. 'How transferable are features in deep neural networks?' In: *Proceedings of the 28th Conference on Neural Information Processing Systems*, 3320–3328, 2014. URL: https://dl.acm.org/doi/10.5555/2969033.2969197.

[255] N. Yu and B. Foggo. 'Stochastic valuation of energy storage in wholesale power markets'. *Energy Economics* 64, 177–185, 2017. DOI: 10.1016/j.eneco.2017.03.010.

[256] G. Zachmann. 'Electricity wholesale market prices in Europe: Convergence?' *Energy Economics* 30(4), 1659–1671, 2008. DOI: 10.1016/j.eneco.2007.07.002.

[257] Q. Zhou, J. Li, B. Shuai, H. Williams, Y. He, Z. Li, H. Xu and F. Yan. 'Multi-step reinforcement learning for model-free predictive energy management of an electrified off-highway vehicle'. *Applied Energy* 255, 113755, 2019. DOI: 10.1016/j.apenergy.2019.113755.

[258] F. Ziel, R. Steinert and S. Husmann. 'Forecasting day ahead electricity spot prices: The impact of the EXAA to other European electricity markets'. *Energy Economics* 51, 430–444, 2015. DOI: 10.1016/j.eneco.2015.08.005.

[259] H. Zou and T. Hastie. 'Regularization and variable selection via the elastic net'. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(2), 301–320, 2005. DOI: 10.1111/j.1467-9868.2005.00503.x.

[260] P. Zou, Q. Chen, Q. Xia, G. He and C. Kang. 'Evaluating the contribution of energy storages to support large-scale renewable generation in joint energy and ancillary service markets'. *IEEE Transactions on Sustainable Energy* 7(2), 808–818, 2016. DOI: 10.1109/tste.2015.2497283.

CURRICULUM VITÆ

Jesus Lago was born in May 1989 in Vigo, Spain. In 2013, he obtained his five-years Engineer's degree in Control, Electronics, and Robotics from the University of Vigo. Next, he got admitted at the University of Freiburg, Germany, to pursue a M.Sc. in Microsystems Engineering. He graduated in 2016 with a major in Optimization and Optimal Control as one of the top students of his class.

In 2016, he started his PhD at the Delft Center for Systems and Control (DCSC), Delft University of Technology, supervised by Prof. Bart De Schutter. His research focused on forecasting and control techniques that, by interaction with the electricity markets, allow more integration of renewable sources and lead to market agents increasing their profits. During his PhD, he was also employed by the Flemish Institute for Technological Research (VITO) and affiliated with Energyville. There, he performed most of his research and was co-supervised by Dr. Fjo De Ridder and Dr. Gowri Suryanarayana. His PhD research was funded by a Marie Skłodowska-Curie training network named INCITE.

The forecasting techniques developed during his PhD program received an award in an international forecasting competition. In addition, he had the opportunity to collaborate with multiple research departments and companies across Europe: the group of Prof. Bertrand Raison at the University of Grenoble, the group of Prof. Dirk Saelens at KU Leuven, the group of Prof. Moritz Diehl at the University of Freiburg, the group of Prof. Rafał Weron at Wrocław University of Science and Technology, the group of Prof. Geert Deconinck at KU Leuven, the group of Prof. Laurens de Vries at Delft University of Technology, the group of Prof. Lieve Helsen at KU Leuven, the company 3E in Belgium, and the company Ecovat in The Netherlands.

LIST OF PUBLICATIONS

JOURNAL ARTICLES - LEAD AUTHOR

11. J. Lago, G. Marcjasz, B. De Schutter and R. Weron. 'Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark'. *Renewable and Sustainable Energy Reviews*, 2020. Under review.

10. J. Lago, K. Poplavskaya, G. Suryanarayana and B. De Schutter. 'A market framework for grid balancing support through imbalances trading'. *Renewable and Sustainable Energy Reviews*, 2020. Under review.

9. T. Pippia[1], J. Lago[1], R. De Coninck and B. De Schutter. 'Scenario-based nonlinear model predictive control for building heating systems'. *Applied Energy*, 2020. Under review.

8. J. Lago, G. Suryanarayana, E. Sogancioglu and B. De Schutter. 'Optimal control strategies for seasonal thermal energy storage systems with market interaction'. *IEEE Transactions on Control Systems Technology*, Early Access, 2020. DOI: 10.1109/TCST.2020.3016077

7. N. Sapountzoglou[2], J. Lago[2], B. De Schutter and B. Raison. 'A generalizable and sensor-independent deep learning method for fault detection and location in low-voltage distribution grids'. *Applied Energy* 276, 115299, 2020. DOI: 10.1016/j.apenergy.2020.115299

6. N. Sapountzoglou[2], J. Lago[2] and B. Raison. 'Fault diagnosis in low voltage smart distribution grids using gradient boosting trees'. *Electric Power Systems Research* 182, 106254, 2020. DOI: 10.1016/j.epsr.2020.106254

5. J. Lago, F. De Ridder, W. Mazairac and B. De Schutter. 'A 1-dimensional continuous and smooth model for thermally stratified storage tanks including mixing and buoyancy'. *Applied Energy* 248, 640–655, 2019. DOI: 10.1016/j.apenergy.2019.04.139

4. J. Lago, K. De Brabandere, F. De Ridder and B. De Schutter. 'Short-term forecasting of solar irradiance without local telemetry: A generalized model using satellite data'. *Solar Energy* 173, 566–577, 2018. DOI: 10.1016/j.solener.2018.07.050

3. J. Lago, F. De Ridder and B. De Schutter. 'Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms'. *Applied Energy* 221, 386–405, 2018. DOI: 10.1016/j.apenergy.2018.02.069

2. J. Lago, F. De Ridder, P. Vrancx and B. De Schutter. 'Forecasting day-ahead electricity prices in Europe: The importance of considering market integration'. *Applied Energy* 211, 890–903, 2018. DOI: 10.1016/j.apenergy.2017.11.098

[1] J. Lago and T. Pippia contributed equally to this work
[2] J. Lago and N. Sapountzoglou contributed equally to this work

1. J. Lago, M. Erhard and M. Diehl. 'Warping model predictive control for application in control of a real airborne wind energy system'. *Control Engineering Practice* 78, 65–78, 2018. DOI: 10.1016/j.conengprac.2018.06.00

JOURNAL ARTICLES - REGULAR AUTHOR

7. K. Poplavskaya, J. Lago, S. Strömer and L. De Vries. 'Making the most of short-term flexibility in the balancing market: Opportunities and challenges in the new balancing market design'. *Energy Policy*, 2020. Under review.

6. I. De Jaeger, J. Lago and D. Saelens. 'A probabilistic building characterization method for district energy simulations'. *Energy and Buildings*, 2020. Under review.

5. G. Suryanarayana, J. Arroyo, L. Helsen and J. Lago. 'A Data Driven Method for Optimal Sensor Placement in Multi-Zone Buildings'. *Energy and Buildings*, 2020. Under review.

4. A. Soares, J. Camargo, J. Al-Koussa, J. Diriken, J. Van Bael, J. Lago. 'Efficient state of charge estimation for thermally stratified storage tanks with buoyancy and mixing effects'. *Journal of Energy Storage*, 2020. Under review.

3. G. Marcjasz, J. Lago and R. Weron. 'Neural networks in day-ahead electricity price forecasting: Single vs. multiple outputs'. *Applied Energy*, 2020. Under review.

2. K. Poplavskaya, J. Lago and L. De Vries. 'Effect of market design on strategic bidding behavior: Model-based analysis of European electricity balancing markets'. *Applied Energy* 270, 115130, 2020. DOI: 10.1016/j.apenergy.2020.115130

1. G. Suryanarayana, J. Lago, D. Geysen, P. Aleksiejuk and C. Johansson. 'Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods'. *Energy* 157, 141–149, 2018. DOI: 10.1016/j.energy.2018.05.111

CONFERENCE PAPERS

4. J. Lago, E. Sogancioglu, G. Suryanarayana, F. De Ridder and B. De Schutter. 'Building day-ahead bidding functions for seasonal storage systems: A reinforcement learning approach'. In: *Proceedings of the IFAC Workshop on Control of Smart Grid and Renewable Energy Systems*, 488–493, 2019. DOI: 10.1016/j.ifacol.2019.08.258

3. T. Pippia, J. Lago, R. De Coninck, J. Sijs and B. De Schutter. 'Scenario-based model predictive control approach for heating systems in an office building'. In: *Proceedings of the 15th IEEE International Conference on Automation Science and Engineering*, 1243–1248, 2019. DOI: 10.1109/coase.2019.8842846

2. J. Lago, K. De Brabandere, F. De Ridder and B. De Schutter. 'A generalized model for short-term forecasting of solar irradiance'. In: *Proceedings of the 2018 IEEE Conference on Decision and Control*, 3165–3170, 2018. DOI: 10.1109/cdc.2018.8618693

1. I. De Jaeger, J. Lago and D. Saelens. 'A probabilistic approach to allocate building parameters within district energy simulations'. In: *Proceedings of the Urban Energy Simulation Conference*, 2018. URL: https://lirias.kuleuven.be/2786819