Statistical Arbitrage Trading on Electricity Markets Using Deep Reinforcement Learning

Statistical Arbitrage Trading on Electricity Markets Using Deep Reinforcement Learning

S. Demir

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Demir, S.

Statistical Arbitrage Trading on Electricity Markets Using Deep Reinforcement Learning/ by S. Demir.
Eindhoven: Technische Universiteit Eindhoven, 2023. Proefschrift.

Cover design by S. Demir

A catalogue record is available from the Eindhoven University of Technology Library

ISBN 978-90-386-5758-5

Keywords: Algorithmic trading, A2C, A3C, Deep reinforcement learning, Electricity markets, Machine learning, Price forecasting, Statistical arbitrage

The work in this thesis has been sponsored by Scholt Energy.

Printed by ADC Nederland, ’s-Hertogenbosch Netherlands

Statistical Arbitrage Trading on Electricity Markets Using Deep Reinforcement Learning

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof. dr. S.K. Lenaerts, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op woensdag 7 juni 2023 om 11:00 uur

door

Sumeyra Demir

geboren te Izmir, Turkije

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

Voorzitter:	prof. dr. ir. A.B. Smolders
Promotor:	prof. dr. K. Kok
Copromotor:	dr. N. Paterakis
Promotiecommissieleden:	prof. dr. D. Ernst (the University of Liège)
	prof. dr. M. Pechenizkiy
	dr. M. Amelin (KTH Royal Institute of Technology)
	dr. P. Nguyen
Adviseur:	W. van Alst (Scholt Energy)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Preface

My PhD has been a tremendous adventure. Though the journey hasn't always been simple, the fleeting moments of hardship have moulded me into both a better researcher and person. Since I would not be here today without the support and guidance I received throughout my PhD, I want to use this opportunity to thank the people that have made it all possible.

First and foremost, I would like to thank my promotor Koen Kok and my supervisor Nikolaos Paterakis for their support and guidance, and also for the freedom they offered me. It was essential to my growth as an independent researcher.

I am equally grateful to Scholt Energy, and particularly Walter van Alst and Frank van Gastel, for helping make my dream a reality. They provided unwavering support and supplemented my theoretical understanding of the markets with practical insights from the industry. From Scholt, I would also like to thank the balancing and sourcing team, especially Bart Stappers, for their friendship.

Furthermore, I would like to thank Krystof Mincev for the research collaborations, making my PhD journey more fruitful. Programming alongside you has made me a significantly better AI practitioner.

I would also like to thank my committee members, Damien Ernst, Mykola Pechenizkiy, Mikael Amelin and Phuong Nguyen for their valuable time and effort. They have undoubtedly helped improve this thesis.

From TU/e, I would like to thank all my EES colleagues for easing the solitude that can take hold during a PhD. I am especially thankful to Guus Pemen for being there when I needed him the most. Also special thanks must go to Irena Dukovska and Mauricio Salazar Duque for their valuable friendship.

To my dearest friends Ezgi Vissing and Mert Turhan, I also want to extend my deepest thanks for putting up with me, listening to my complaints, and offering sage advice. I would also like to use this opportunity to thank my other friends, Seda Baser, Rabiye Ceran, Muge Onder, Sezen Vatensever, Inci Tezel Senturk, Ozge Gure, Zeynep Calik, Maren Suilman, Alicja Mokwinska, Tuncay Olcer, and Diederik M Roijers, for their valuable friendship. Despite the distance, their friendship has buoyed me these past years.

Last but not least, I would like to thank my family for their boundless support. They have been a continued source of inspiration. I am and will always be eternally grateful to them for standing beside me no matter the squalls.

Sumeyra Demir
Eindhoven, April 2023

Summary

Statistical Arbitrage Trading on Electricity Markets Using Deep Reinforcement Learning

Transitioning from non-renewables to renewables has made electricity prices more volatile. Because of increased volatility, forecasting energy prices has become more challenging. The European single intraday coupled market (CID) was established to combat this price uncertainty. With an established liquid CID, arbitrage trading opportunities have emerged across short-term markets. Note that potential arbitrage opportunities arise when prices vary between markets or when prices deviate from their long-term trends. Arbitrage trading could help in the energy transition process by reducing price differences between markets, and increasing market liquidity as well as efficiency while yielding profits for arbitrage traders. This PhD thesis aims to automate and optimise the decision-making of arbitrage trading agents for short-term electricity markets, namely the day-ahead market (DAM), CID and balancing market (BAL).

In optimising the decision-making of intelligent agents, the thesis makes eight contributions to three research areas: intraday market analytics, electricity price forecasting, and electricity trading. As our first contribution, visualisation methods are developed to gain novel market insights about the CID. These insights aid in developing trading algorithms. An analysis method, utilising several exploratory visual analytics tools, is devised. This method can monitor market trends, behaviours and price consensuses, as well as identify trading opportunities and market risks. To the best of our knowledge, we are the first to implement exploratory visual analytics tools for the CID, thus contributing to intraday market analytics research.

In addition to the above, this thesis makes five contributions to electricity price forecasting research. Firstly, the use of technical indicator features as inputs, such as moving averages and Bollinger bands, is proposed to capture the behavioural biases of traders. These features allow machine learning models to capture trading signals and better predict electricity prices. Secondly, we deploy data augmentation methods utilising autoencoders and generative adversarial networks. Data augmentation expands training set sizes and reduces the impacts of data drift. Thirdly, we employ technical indicator features alongside data augmentation in DAM price forecasting. The combined use of technical features and data augmentation further boosts forecast accuracies. Fourthly, a feature engineering method is proposed for CID forecasting. The proposed features capture the underlying price drivers hidden in the limit order book. Lastly, we evaluate machine learning models for BAL price forecasting. We are the first, to our knowledge, to compare the performances of LASSO, random forest, gradient boosting, and deep neural networks in BAL forecasting. Accurate forecasts are required to optimise risk-to-reward ratios of arbitrage trading strategies. Our contributions improve the benchmark forecasting accuracies of DAM, CID and BAL prices by, on average, 27%, 35% and 19% respectively.

Finally, this thesis makes two contributions to electricity trading research. Algorithms and methods are employed for optimising the risk-to-reward ratio of an end-to-end arbitrage trading strategy in short-term electricity trading. A novel rule-based trading strategy, which uses DAM and CID price forecasts, is developed for the DAM. This strategy accurately opened 74% of DAM positions, which later closed on the CID or BAL. Novel trading methods utilising deep reinforcement learning (DRL) are developed to optimise arbitrage trading decisions across the CID and BAL. We employ the synchronous advantage actor-critic algorithm (A2C) and asynchronous advantage actor-critic algorithm (A3C) with function approximators of deep neural networks. We are the first, to our knowledge, to employ A3C or A2C for arbitrage trading across the CID and BAL. Trained autonomous trading agents secured significant profit across a test set. Note that the test set contained Dutch DAM, CID and BAL contracts spanning 2020. While A3C traded roughly 7 GWh of electricity and yielded profits of approximately €20k, A2C traded close to 34 GWh of electricity and generated even higher positive profits of approximately €98k.

In conclusion, this thesis makes eight contributions to intraday market analytics, electricity price forecasting, and electricity trading research. We hope that our contributions would prove useful to researchers and traders who want to improve their forecasting accuracies and reward-to-risk ratios as well as develop autonomous trading methods for short-term electricity markets. Moreover, our contributions should aid in the reduction of inefficiencies emerging across electricity markets.

Samenvatting

Statistische Arbitragehandel op Elektriciteitsmarkten met Behulp van Diepgaand Leren

Door de overgang van niet-hernieuwbare naar hernieuwbare energie zijn de elektriciteitsprijzen volatieler geworden. Vanwege de toegenomen volatiliteit is het voorspellen van energieprijzen een grotere uitdaging geworden. Om deze prijsonzekerheid tegen te gaan, is de Europese single intraday coupled market (CID) opgericht. Met een gevestigde liquide CID zijn er arbitragehandelsmogelijkheden ontstaan op kortetermijnmarkten voor elektriciteit. Houd er rekening mee dat potentiële arbitragemogelijkheden zich voordoen wanneer prijzen variëren tussen markten of wanneer prijzen afwijken van hun langetermijntrends. Arbitragehandel zou kunnen helpen bij het energietransitieproces door prijsverschillen tussen markten te verkleinen en de marktliquiditeit en efficiëntie te vergroten, terwijl het winst oplevert voor arbitragehandelaren. Dit doctoraatsproefschrift heeft tot doel de besluitvorming van arbitragehandelsagenten voor kortetermijnelektriciteitsmarkten, namelijk de day-ahead markt (DAM), CID en balanceringsmarkt (BAL), te automatiseren en te optimaliseren.

Bij het optimaliseren van de besluitvorming van intelligente agenten levert het proefschrift acht bijdragen aan drie onderzoeksgebieden: intraday-marktanalyse, elektriciteitsprijsprognoses en elektriciteitshandel. Als onze eerste bijdrage worden visualisatiemethoden ontwikkeld om nieuwe marktinzichten over de CID te verkrijgen. Deze inzichten helpen bij het ontwikkelen van handelsalgoritmen. Er wordt een analysemethode ontwikkeld, waarbij gebruik wordt gemaakt van verschillende verkennende visuele analysetools. Deze methode kan mark-

ttrends, gedragingen en prijsconsensus volgen, evenals handelsmogelijkheden en marktrisico's identificeren. Voor zover wij weten, zijn wij de eersten die verkennende visuele analysetools voor de CID implementeren en zo bijdragen aan intraday-marktanalyseonderzoek.

Naast het bovenstaande levert dit proefschrift vijf bijdragen aan onderzoek naar elektriciteitsprijsprognoses. Ten eerste het gebruik van technische indicatorkenmerken als invoer, zoals voortschrijdende gemiddelden en Bollingerbanden, wordt voorgesteld om de gebruikelijke vooroordelen van handelaren. Met deze functies kunnen machine learning-modellen handelssignalen vastleggen en elektriciteitsprijzen beter voorspellen. Ten tweede passen we methoden voor gegevensvergroting toe met behulp van auto-encoders en generatieve vijandige netwerken. Data-augmentatie breidt de omvang van trainingssets uit en vermindert de impact van data-drift. Ten derde gebruiken we technische indicatorfuncties naast gegevensvergroting bij DAM-prijsvoorspellingen. Het gecombineerde gebruik van technische kenmerken en data-uitbreiding verhoogt de voorspellingsnauwkeurigheid verder. Ten vierde wordt een feature engineering-methode voorgesteld voor CID-prognoses. De voorgestelde functies leggen de onderliggende prijsbepalende factoren vast die verborgen zijn in het limietorderboek. Ten slotte evalueren we machine learning-modellen voor BAL-prijsprognoses. Voor zover wij weten, zijn wij de eersten die de prestaties van LASSO, willekeurig bos, gradiëntversterking en diepe neurale netwerken in BAL-voorspelling vergelijken. Nauwkeurige prognoses zijn vereist om de risico-opbrengstverhoudingen van arbitragehandelsstrategieën te optimaliseren. Onze bijdragen verbeteren de benchmarkvoorspellingsnauwkeurigheden van DAM-, CID- en BAL-prijzen met gemiddeld 27%, 35% en 19%.

Ten slotte levert dit proefschrift twee bijdragen aan onderzoek naar elektriciteithandel. Er worden algoritmen en methoden gebruikt voor het optimaliseren van de risico-opbrengstverhouding van een end-to-end arbitragehandelsstrategie bij de kortetermijnhandel in elektriciteit. Voor de DAM is een nieuwe op regels gebaseerde handelsstrategie ontwikkeld, die gebruikmaakt van DAM- en CID-prijsprognoses. Deze strategie opende nauwkeurig 74% van DAM-posities, die later op de CID of BAL werden gesloten. Er zijn nieuwe handelsmethoden ontwikkeld die gebruikmaken van Deep Reinforcement Learning (DRL) om arbitragehandelsbeslissingen over de CID en BAL te optimaliseren. We gebruiken het synchrone voordeel actor-critic algoritme (A2C) en het asynchrone voordeel actor-critic algoritme (A3C) met functiebenaderingen van diepe neurale netwerken. Voor zover wij weten, zijn wij de eersten die A3C of A2C gebruiken voor arbitragehandel over de CID en BAL. Getrainde autonome handelsagenten behaalden aanzienlijke winst over een testset. Merk op dat de testset Nederlandse DAM-, CID- en BAL-contracten bevatte voor 2020. Terwijl A3C ongeveer 7 GWh aan elektriciteit verhandelde en een winst opleverde van ongeveer €20k, verhandelde A2C bijna 34 GWh aan elektriciteit en genereerde een nog hoger positief winst van ongeveer €98k.

Concluderend levert dit proefschrift acht bijdragen aan intraday-marktanalyses, elektriciteitsprijsprognoses en onderzoek naar elektriciteitshandel. We hopen dat onze bijdragen nuttig zullen zijn voor onderzoekers en handelaren die hun voorspellingsnauwkeurigheden en beloning-naar-risicoverhoudingen willen verbeteren en autonome handelsmethoden voor kortetermijnmarkten willen ontwikkelen. Bovendien zouden onze bijdragen moeten helpen bij het terugdringen van inefficiënties die op de elektriciteitsmarkten ontstaan.

Contents

Preface v

Summary vii

Samenvatting xi

1 Introduction 1

1.1 Motivation of the Research . . . 1

1.2 Research Goals and Questions . . . 2

1.3 Main Contributions . . . 4

1.4 Structure of the Thesis . . . 9

1.5 Mathematical Notation . . . 12

I Electricity Markets 13

2 Background: Electricity Markets 15

2.1 Introduction . . . 15

2.2 Day-Ahead Markets . . . 16

2.3 Continuous Intraday Markets . . . 17

2.4 Balancing Markets . . . 21

2.5 Concluding Remarks . . . 22

3 Exploring the Continuous Intraday Market 23
3.1 Introduction 23
3.2 Visual Analytics Approach to Continuous Intraday Market Exploration 25
3.3 Conclusion 32

II Forecasting Electricity Market Prices 33

4 Background: Forecasting Electricity Market Prices 35
4.1 Introduction 35
4.2 Literature Survey 36
4.3 Autoregressive-Exogenous Models (ARXs) 39
4.4 Machine Learning Models 39
4.5 Concluding Remarks 42

5 Forecasting Day-Ahead Market Prices: Technical Indicators 43
5.1 Introduction 44
5.2 Technical Indicators 45
5.3 Feature Engineering Approach to Day-Ahead Market Price Forecasting 50
5.4 Results and Discussion 54
5.5 Conclusion 55

6 Forecasting Day-Ahead Market Prices: Data Augmentation 57
6.1 Introduction 58
6.2 Kullback-Leibler Divergence 62
6.3 Autoencoders 63
6.4 Variational Autoencoders 63
6.5 Wasserstein Generative Adversarial Networks 64
6.6 Data Augmentation to Day-Ahead Market Price Forecasting . . . 67
6.7 Numerical Results and Discussion 79
6.8 Conclusion 82

7 Forecasting Day-Ahead Market Prices 83
7.1 Introduction 83
7.2 Ensemble Approach to Day-Ahead Market Price Forecasting . . . 84
7.3 Results and Discussion 88
7.4 Conclusion 89

8 Forecasting Continuous Intraday Market Prices 91
8.1 Introduction 92
8.2 Continuous Intraday Market Prices 93
8.3 Feature Engineering Approach to Continuous Intraday Market Price Forecasting 94
8.4 Results and Discussion 100
8.5 Conclusion 103

9 Forecasting Balancing Market Prices 105
9.1 Introduction 105
9.2 Balancing Market Price Forecasting Using Machine Learning Models 106
9.3 Results and Discussion 109
9.4 Conclusion 111

III Statistical Arbitrage Trading on Electricity Markets 113

10 Background: Statistical Arbitrage Trading on Electricity Markets 115
10.1 Introduction 115
10.2 Literature Survey 116
10.3 Statistical Arbitrage Trading (SAT) 117
10.4 Reinforcement Learning 120
10.5 Concluding Remarks 122

11 Trading Across the Intraday and Balancing Markets 123
11.1 Introduction 124
11.2 Asynchronous Advantage Actor-Critic Method to Intraday Market Trading 125
11.3 Case Study 134
11.4 Conclusion 142

12 Trading Across the Day-Ahead, Intraday and Balancing Markets 143
12.1 Introduction 143
12.2 Rule-Based Approach to Day-Ahead Market Trading 145
12.3 Advantage Actor-Critic Approach to Intraday Market Trading . . 146
12.4 Case Study 150
12.5 Results and Discussion 151
12.6 Conclusion 154

13 Conclusion and Recommendations **155**
13.1 Conclusion 156
13.2 Impact of the Research 159
13.3 Suggestions for Future Research 161

Bibliography **167**

Chapter 1

Introduction

1.1 Motivation of the Research

An urgent need to transition away from fossil fuels, tackle climate change, and achieve energy independence exists. Though necessary, the transition away from dirty yet stable energy sources raises technological and economic challenges. In this thesis, we offer novel AI-based solutions to tackle some of these challenges. Our solutions help ensure the stability of electrical grids and the efficiency of short-term electricity markets: the day-ahead market (DAM), the continuous intraday market (CID) and the balancing market (BAL).

The supply from renewable energy sources is dependent on weather conditions which are difficult to forecast accurately. Given this limitation, electricity traders, experiencing regulators' efforts to incentivise renewable energy production, have been forced to adjust their trading strategies. They have pivoted from long-term to short-term trading, in order to be able to use more recent and reliable forecasts, i.e. hour-ahead forecasts. This shift in liquidity, however, has been insufficient to offset the additional risk created by renewable energy productions and combat increasing price deviations emerging across markets.

Arbitrage trading strategies offer the opportunity to reduce price differences between markets and increase market liquidity and efficiency while yielding profits for traders [1–7]. Arbitrage trading strategies are short-term investment decisions that employ analysis to determine reasonable price levels at which to enter and exit trades without making long-term physical commitments. They arise when price differences between markets widen. Arbitrage traders try to profit from such price differences by, for instance, buying energy on the DAM and selling back the same amount of energy, at a higher price, on the CID/BAL. In this context, arbitrage trading is not risk-free and only results in a profit when traders correctly predict the direction of prices. Arbitrage trading is automated when mathematical or intelligent learning methods, e.g. deep reinforcement learning, are employed. Automated arbitrage trading is also called statistical arbitrage trading.

Arbitrage trading electricity is more difficult than arbitrage trading other commodities. The volatility of electricity prices often surpasses the volatility of other commodities [8]. The uncertainty from renewable sources mentioned above impacts the accuracy of electricity price forecasts, increasing the risk for traders. Unlike other commodity traders, electricity traders are forced to close their open positions at the end of the trading session at volatile BAL prices; because demand must always match supply across the grid. Short trading windows, during which trades are executed, further complicate the optimisation of trading opportunities. Novel methods are thus needed to predict prices accurately and make trading decisions optimally.

1.2 Research Goals and Questions

This PhD thesis aims to optimise the decision-making of arbitrage trading agents for short-term electricity markets. In optimising the decision-making of intelligent agents, the thesis focuses on answering the following research questions:

1. How can trading behaviours across the CID be visualised?

2. Can new features capable of capturing the effects of technical price drivers be identified for the DAM?

3. How can time series be augmented to boost DAM prediction accuracies?

4. Can ensembles be used to boost forecast accuracies of DAM prices?

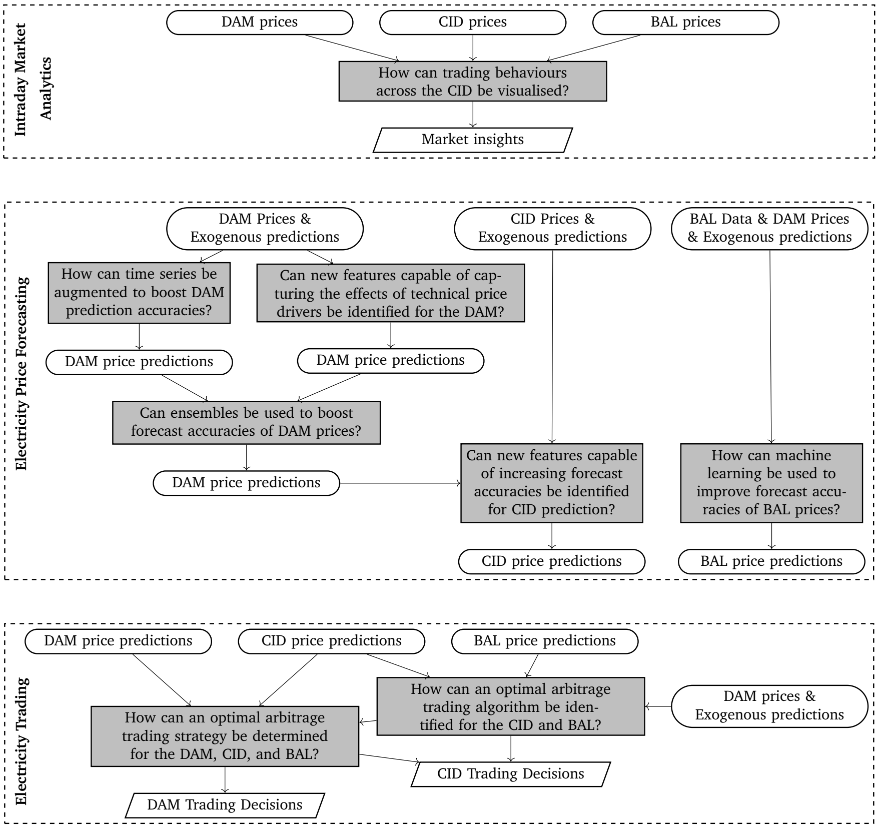

Intraday Market Analytics
DAM prices
CID prices
BAL prices
How can trading behaviours across the CID be visualised?
Market insights
Electricity Price Forecasting
DAM Prices & Exogenous predictions
CID Prices & Exogenous predictions
BAL Data & DAM Prices & Exogenous predictions
How can time series be augmented to boost DAM prediction accuracies?
Can new features capable of capturing the effects of technical price drivers be identified for the DAM?
DAM price predictions
DAM price predictions
Can ensembles be used to boost forecast accuracies of DAM prices?
DAM price predictions
Can new features capable of increasing forecast accuracies be identified for CID prediction?
How can machine learning be used to improve forecast accuracies of BAL prices?
CID price predictions
BAL price predictions
Electricity Trading
DAM price predictions
CID price predictions
BAL price predictions
How can an optimal arbitrage trading algorithm be identified for the CID and BAL?
DAM prices & Exogenous predictions
How can an optimal arbitrage trading strategy be determined for the DAM, CID, and BAL?
CID Trading Decisions
DAM Trading Decisions

Figure 1.1: The diagram highlighting the research questions of the thesis. Research questions are shown with grey rectangles. Data, both actual and predictions, are shown with rounded rectangles. Trading insights and decisions are shown with parallelograms. Exogenous predictions refer to predictions of exogenous variables, such as load, generation, temperature, wind speed, solar irradiance and precipitation.

5. Can new features capable of increasing forecast accuracies be identified for CID prediction?

6. How can machine learning be used to improve forecast accuracies of BAL prices?

7. How can an optimal arbitrage trading algorithm be identified for the CID and BAL?

8. How can an optimal arbitrage trading strategy be determined for the DAM, CID, and BAL?

Figure 1.1 shows the order in which the research questions are tackled.

1.3 Main Contributions

This PhD thesis makes eight contributions to three research areas, namely intraday market analytics, electricity price forecasting, and electricity trading. Figure 1.2 highlights the thesis's contributions. Below these contributions are further detailed.

Intraday Market Analytics

Original market insights about the CID are attained using novel exploratory visualisation techniques. Market insights aid in developing both our forecasting methods and trading algorithms. Our contribution is described below. We expect our contribution to be useful for practitioners and researchers seeking additional market insights with easily implementable techniques.

1. **Exploratory visual analytics tools for the CID:** Since 2018 a new integrated European intraday trading platform, the CID, has been operational, offering market participants cross-border liquidity. Trading activity on the CID has increased substantially since its inception, mainly due to increasing renewable energy penetration. CID trading volumes should increase further, in line with anticipated renewable energy production growth. Given its rising status, more research is needed to better understand the workings of the CID. With this in mind, we develop exploratory visual analytic tools capable of analysing CID trade flows. Some of the visualisation techniques are designed, using data from the CID limit order book and trade book, to monitor market trends, behaviours, price consensus and liquidity. Meanwhile, other visualisation techniques are created, using CID volumes and short-term electricity market prices, to identify trading opportunities and market risks. Visualisations are created using CID data available for the Dutch market area.

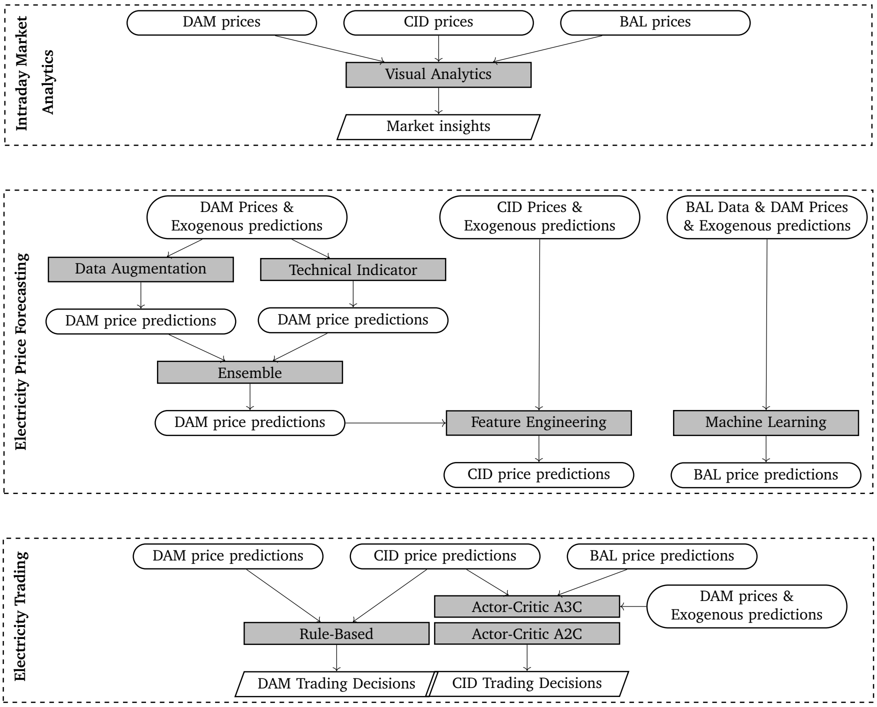


Figure 1.2: The diagram highlighting the contributions of the thesis. Contributions, methods, are shown with grey rectangles. Data, both actual and predictions, are shown with rounded rectangles. Trading insights and decisions are shown with parallelograms. Exogenous predictions refer to predictions of exogenous variables, such as load, generation, temperature, wind speed, solar irradiance and precipitation.

Electricity Price Forecasting

Statistical arbitrage trading (SAT) agents need accurately predicted prices to reduce uncertainty, increase profits, and ensure better decision-making. To do so, this thesis develops novel forecasting methods, which significantly increase the accuracy of short-term electricity price forecasts. These novel methods impact the field of electricity price forecasting through five unique contributions listed below. We expect our contributions to benefit electricity traders and researchers who seek to improve their forecasting accuracies of electricity prices.

2. **Technical indicators for forecasting DAM prices:** Volatility and forecast errors of DAM prices have grown in recent years. Changing market conditions, epitomised by geopolitical factors and increasing renewable energy production, have spurred this growth. If forecast accuracies of DAM prices are to improve, new features capable of capturing the effects of technical or fundamental price drivers should be identified. Technical indicators (TIs), such as Bollinger Bands and Momentum indicators, are widely used across financial markets to identify traders' behavioural biases. To date, TIs have never been applied to the forecasting of DAM prices. TI features capable of capturing the behavioural biases of DAM traders are proposed to improve the forecasting accuracies of DAM prices. The simple inclusion of TI features in DAM forecasting is shown to significantly boost the accuracies of machine learning models for the Belgian DAM; reducing the root mean squared errors of linear, ensemble, and deep model forecasts by up to 4.50%, 5.42%, and 4.09% respectively. Moreover, tailored TIs are identified for each of these models, highlighting the added explanatory power offered by technical features.

3. **Data augmentation methods for forecasting DAM prices:** A model's expected generalisation error is inversely proportional to its training set size. This relationship can pose a problem when modelling multivariate time series because structural breaks and low sampling rates can severely restrict training set sizes, increasing a model's expected generalisation error by spurring regression model overfitting. Artificially expanding the training set size, using data augmentation methods, can, however, counteract the restrictions imposed by small sample sizes, increasing a model's robustness to overfitting and boosting out-of-sample prediction accuracies. To artificially expand training set sizes and boost DAM price prediction accuracies, autoencoders (AEs), variational autoencoders (VAEs) and Wasserstein generative adversarial networks with a gradient penalty (WGAN-GPs) are proposed for time series augmentation. To evaluate our proposed augmentors, as a case study we forecast Belgian and Dutch DAM prices using both autoregressive models and artificial neural networks. Overall, our results demonstrate that AEs, VAEs, and WGAN-GPs can significantly boost regression accuracies; on average decreasing mean absolute errors by 2.23%, 2.73% and 2.97% respectively.

4. **Ensemble methods for forecasting DAM prices:** Given observed improvements obtained utilising TIs and augmentation methods, ensemble

methods utilising various combinations of each method are proposed. Ensemble methods should further improve forecasting accuracies of DAM prices. Across the Belgian and Dutch DAMs, our results show that combining AE, VAE, and WGAN-GP forecasts can, on average, increase naive benchmark accuracies by 25.40%. Moreover, our results demonstrate that combining TI forecasts with augmentation method forecasts can similarly boost naive benchmark accuracies by up to 24.04%. An ensemble method averaging AE, VAE, WGAN-GP, and TI model forecasts outperformed all other evaluated forecasting methods, decreasing naive benchmark errors by 27.63%.

5. **A feature engineering method for forecasting CID prices:** Increasing renewable energy production has significantly increased the volatility of CID prices and decreased the forecasting accuracy of regression models. If forecast accuracies of CID prices are to improve, feature engineering methods, which capture significant CID price drivers, should be identified. A novel feature engineering method for capturing CID price drivers is proposed. The CID forecasting accuracies of machine learning models, accepting these features as inputs, are evaluated. Note that while most CID studies only focus on forecasting the volume-weighted average price of trades (VWAP), this thesis focuses on predicting: the VWAP, the lowest traded price and the highest traded price, the average best ask price, the average best bid price, and the average mid-price to account for continuous trading characteristics. Overall, LASSO feature selection suggests that our proposed features, such as the VWAP of exported trades, the number of bid orders, the number of limit order book revisions and the minimum of the best ask price for a contract, are significant predictors of CID prices. Using the selected features as inputs in a range of models is shown to boost their Dutch CID forecasting accuracies. For instance, deep neural networks (DNN), using our novel input features, are found to outperform naive benchmark forecasts by 35%.

6. **Machine learning methods for forecasting BAL prices:** BAL prices are the most unpredictable among short-term electricity market prices because the BAL deals with unforeseen incidents which are random and hard to predict. Nevertheless, accurate forecasts of BAL prices are valuable inputs of arbitrage trading strategies. We develop machine learning methods capable of better forecasting BAL prices. The forecasting mod-

els, such as LASSO, random forest, gradient boosting, and deep neural networks, are evaluated.

Given the demonstrated gains made possible by using technical indicators (TIs) in our earlier contributions, TI features are used as inputs in evaluated models. As a case study, Dutch BAL prices, namely take and feed prices, are predicted. The best-performing model, deep neural networks, is found to boost the forecasting accuracy of a naive benchmark model, on average, by 19%.

Electricity Trading

This thesis deploys autonomous trading agents to explore statistical arbitrage trading (SAT) strategies. Novel trading algorithms capable of significantly increasing traders' profits are identified. These novel methods advance the field of electricity trading via two distinct contributions listed below. We expect our contributions to benefit electricity traders and researchers who seek to develop state-of-art intelligent trading strategies.

7. **Asynchronous advantage actor-critic (A3C) agent for statistical arbitrage trading across the CID and BAL:** SAT can reduce price differences across the CID and BAL while yielding profits for trading agents. These trading agents aim to optimise the risk-reward ratio of their autonomous trading strategies. To find an optimal trading policy, an algorithm employing a deep reinforcement learning method, namely asynchronous advantage actor-critic (A3C), is proposed. State engineering and selection processes are conducted to increase the performance of trading agents. A novel reward function and goal-based exploration, i.e. behaviour cloning, are introduced to motivate trading agents to explore the state space. A3C is evaluated across the Dutch CID and BAL markets. On 1760 test contracts, A3C trades a total of 7017 MWh of electricity and generates profits of €19927.

8. **Rule-based (RB) and advantage actor-critic (A2C) agents for statistical arbitrage trading across the DAM, CID and BAL:** SAT can reduce price differences across the DAM, CID and BAL while yielding profits for trading agents. SAT agents aim to optimise the risk-reward ratio of an end-to-end trading strategy. To find an optimal DAM trading policy, a rule-based (RB) trading method using DAM and CID price forecasts is

proposed. Using these forecasts, our RB agent opens a majority of DAM positions (74%) accurately. To find optimal CID and BAL trading policies, an algorithm employing a deep reinforcement learning method, namely advantage actor-critic (A2C), is proposed. Evaluated across Dutch short-term markets, A2C yields profits surpassing those obtained using A3C and other benchmarks. Our best agent trades 33805 MWh of electricity across 1760 hourly test contracts; yielding significantly positive profits of €97853.

1.4 Structure of the Thesis

This PhD thesis is divided into three parts. Each part begins with a background chapter that lays the groundwork for the main contributing chapters. Figure 1.3 displays the organisation of the thesis and links between the main chapters. Below, the contents of contributing chapters are summarised for each part of the thesis.

Part I Electricity Markets

The first part of the thesis, consists of Chapters 2-3, describes short-term electricity markets and explores the CID with visual tools. This part aids in developing both forecasting methods in Part II and trading algorithms in Part III.

Chapter 3 This chapter introduces visual analytics tools for analysing the CID, and identifies easily implementable tools for succinctly visualising market risks. The content of this chapter has been published in:

[9] S. Demir, K. Kok, N. G. Paterakis, "Exploratory Visual Analytics for the European Single Intra-day Coupled Electricity Market," in *2020 International Conference on Smart Energy Systems and Technologies (SEST)*, 2020, pp. 1-6. DOI: 10.1109/SEST48500.2020.9203043

Part II Forecasting Electricity Market Prices

Statistical arbitrage trading (SAT) agents, implemented in Part III of the thesis, need accurately predicted prices to reduce uncertainty, increase profits, and ensure better decision-making. To do so, Part II of the thesis, consists of Chapters 4-9, develops novel forecasting methods, which significantly increase the accuracy of price forecasts across short-term electricity markets.

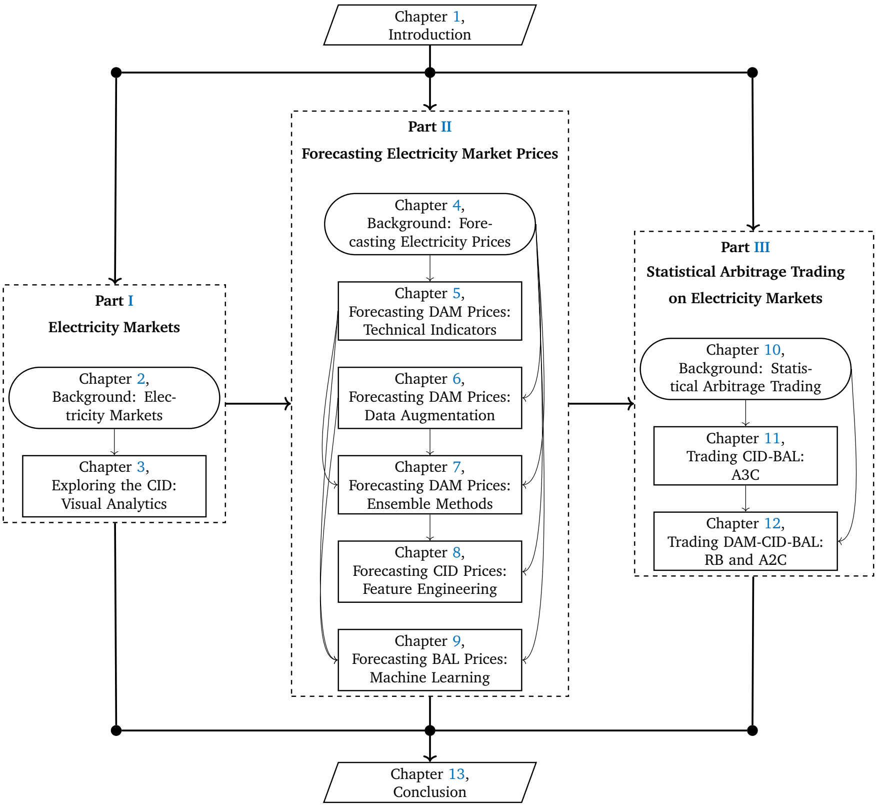


Figure 1.3: The organisation of the thesis. While background chapters are shown with rounded rectangles, contributing chapters are shown with rectangles. The introduction and conclusion chapters are shown with parallelograms.

Chapter 5 This chapter investigates the explanatory power of TIs in DAM price forecasting. This chapter's content has been published in:

[10] S. Demir, K. Mincev, K. Kok, N. G. Paterakis, "Introducing Technical Indicators to Electricity Price Forecasting: A Feature Engineering Study for Linear, Ensemble, and Deep Machine Learning Models," *Applied Sciences*, vol. 10, no. 1, 2020. DOI: 10.3390/app10010255

Chapter 6 This chapter examines the use of data augmentation as a method for boosting DAM forecasting accuracies. The content of this chapter has been published in:

[11] S. Demir, K. Mincev, K. Kok, N. G. Paterakis, "Data Augmentation for Time Series Regression: Applying Transformations, Autoencoders and Adversarial Networks to Electricity Price Forecasting," *Applied Energy*, 304, 117695, 2021. DOI: 10.1016/j.apenergy.2021.117695

Chapter 7 This chapter examines ensemble models that utilise different combinations of forecasts from Chapters 5 and 6. This chapter's content has been published in Sections 3.1 and 5.1 in:

[12] S. Demir, K. Kok, N. G. Paterakis, "Statistical Arbitrage Trading Across Electricity Markets Using the Advantage Actor-Critic Methods," *Sustainable Energy, Grids and Networks*, 34, 101023, 2023. DOI: 10.1016/j.segan-2023.101023

Chapter 8 This chapter studies the explanatory power of novel features created from the limit order book and trade book in CID price forecasting. A part of this chapter's content has been published in Sections 3.2 and 5.2 in:

[12] S. Demir, K. Kok, N. G. Paterakis, "Statistical Arbitrage Trading Across Electricity Markets Using the Advantage Actor-Critic Methods," *Sustainable Energy, Grids and Networks*, 34, 101023, 2023. DOI: 10.1016/j.segan-2023.101023

Chapter 9 This chapter compares a range of machine learning models for BAL price forecasting.

Part III Statistical Arbitrage Trading Across Electricity Markets

The third part of the thesis, consists of Chapters 10-12, deploys autonomous trading agents in the exploration of SAT strategies.

Chapter 11 This chapter investigates the profitability of SAT across the CID and BAL using the A3C algorithm. This chapter's content has been published in:

[13] S. Demir, B. Stappers, K. Kok, N. G. Paterakis, "Statistical Arbitrage Trading on the Intraday Market Using the Asynchronous Advantage Actor-Critic Method," *Applied Energy*, 314, 118912, 2022. DOI: 10.1016/j.apenergy.2022.118912

Chapter 12 This chapter examines the profitability of a SAT strategy across the DAM, CID and BAL using the RB and A2C algorithms. This chapter's content has been published in Sections 4, 5.3 and 5.4 in:

[12] S. Demir, K. Kok, N. G. Paterakis, "Statistical Arbitrage Trading Across Electricity Markets Using the Advantage Actor-Critic Methods," *Sustainable Energy, Grids and Networks*, 34, 101023, 2023. DOI: 10.1016/j.segan-2023.101023

1.5 Mathematical Notation

In this thesis, scalar variables are denoted using lowercase Latin letters, such as x, or symbols, such as θ. Note that exceptions to this notation occur. The number of samples, for instance, is shown with the capital letter N following common convention seen in the existing literature. Vectors are denoted by bold lowercase Latin letters, such as $\mathbf{x}$, or bold symbols, such as $\boldsymbol{\theta}$.

Matrices and sets are denoted by capital letters. Following the existing literature, the set of integers is denoted by $\mathbb{Z}$, and the set of real numbers by $\mathbb{R}$. The set of a real-valued vector of dimension n is shown with $\mathbb{R}^n$.

The expected value is denoted by $\mathbb{E}$. Forecasts are denoted by using a hat on the predicted variable, e.g. $\hat{p}^{dam}$ for the forecast of DAM price p^{dam}.

Functions are denoted by lower and capital letters, such as $f(.)$ and $F(.)$. Functions defined by abbreviations, such as SMA(.) for the simple moving average function, are also used. Note that exceptions still occur for functions. The loss function, for instance, is shown with the calligraphic letter $\mathcal{L}(.)$ following the existing literature.

Part I

Electricity Markets

Chapter 2

Background: Electricity Markets

This chapter provides background information about electricity markets: the first part of the thesis. Specifically, the chapter describes European short-term electricity markets: namely the day-ahead market (DAM), continuous intraday market (CID), and balancing market (BAL).

2.1 Introduction

Energy producers and suppliers must together meet the energy needs of everyday consumers and businesses. In doing so, they forecast hourly or quarter-hourly energy demand and plan a path for satisfying this demand. They do this, while jointly trying to maximise profits and minimise price and volume risks.

Numerous markets for trading electricity and meeting the demand for energy exist: from long-term futures markets with maturities ranging from weeks to years, to short-term markets where contracts with maturities ranging from one hour to one day are traded. Figure 2.1 highlights the order of the electricity market floors; the first trading floor is the futures market, the second trading

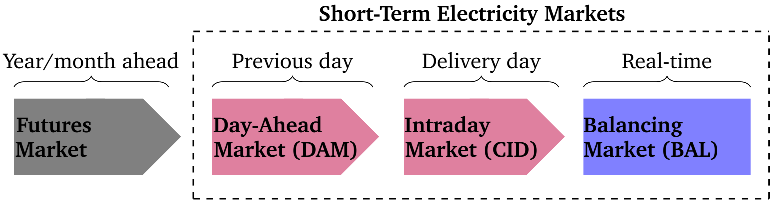


Figure 2.1: Electricity markets.

floor is the DAM, and the final trading floor is the CID. Outstanding positions are cleared on the BAL.

The scope of the thesis comprises short-term electricity markets, namely the DAM, CID, and BAL. To date, research has primarily focused on the DAM. The transition from non-renewable to renewable energy sources is, however, changing this; influencing short-term electricity markets by shifting the share of volume from the DAM to CID. The CID's traded volume and liquidity have risen, as has the quantity of studies researching the CID. Among these studies, [14] explored CID trading behaviours, [15] provided a literature review of CID forecasting, and [16, 17] debated the trading mechanism between continuous and auction-based markets. Nonetheless, more research is needed to understand this growing market.

The next chapter introduces and explains the novel visualisation methods developed for the CID. Before that, however, this chapter focuses on introducing short-term electricity markets, paying extra attention to introducing the CID.

2.2 Day-Ahead Markets

The DAM is the largest short-term electricity market by volume. The DAM is an auction-based market; facilitating the purchase and sale of 24 hourly energy contracts for next-day delivery. Uniquely, DAM clearing prices are computed using a matching mechanism that aggregates all asks and bids submitted before noon. For the DAM's 24 hourly contracts, the matching engine sets prices where demand and supply intersect. Trade settlement occurs once prices are set, with hourly energy delivery commencing at 00:00 the following day.

2.3 Continuous Intraday Markets

Before market participants' outstanding positions become binding on the BAL, traders are granted one final opportunity, through the CID, to close out their positions. The CID facilitates market-based transactions of energy both nationally and internationally. While national markets allow trading only within one market area, coupled markets allow trading between several market areas. There were several intraday coupled markets in Europe such as Nordic, Iberian and EPEX before 2018. As stated in [18], on 12/06/2018, the single intraday coupled market (CID)[1] centralised all previous coupled markets by initially launching across Austria, Belgium, Denmark, Estonia, Finland, France, Germany, Latvia, Lithuania, Norway, the Netherlands, Portugal, Spain and Sweden. The CID was expanded to cover more countries such as Bulgaria, Croatia, the Czech Republic, Hungary, Poland, Romania and Slovenia in 2019. The CID was expanded further by including Italy in 2021, as well as Greece and Slovakia in 2022.

2.3.1 CID Design

As described in [18], the CID system consists of trading, capacity management, and shipping modules. The trading module combines all orders into one shared order book; delivering continuous trading services. The trading module also allocates the first implicit capacity for the capacity management module.[2] Consequently, the capacity management module updates cross-border capacity in real-time. Finally, the shipping module informs market participants, whose orders are matched, about the post-coupling process.

All three modules mentioned above work collaboratively. Per Figure 2.2, initially, participants submit their orders to their market areas. If there is available

[1] The CID might be referred to as the SIDC or XBID by market operators.

[2] Note that the implicit market loss, the allocated capacity difference between two counterparties, is accounted for and calculated as follows. When an order with quantity (q in MWh) and price (p in €/MWh) is submitted to the CID, the trade amount is calculated by multiplying quantity and price: $q \times p$. The trade amount should be the same for both counterparties to match their orders. Implicit losses, such as 5% between the Netherlands and Norway and 4% between the Netherlands and Norway2, are, however, taken into consideration [18]. For example, a bid order is submitted from the Dutch market area with $q^b = 100$ MWh and $p^b = 50$ €/MWh. The trade amount of this order is $100 \times 50 = 5000$ €. If this bid order is to be matched with an ask order from the Norway2 market area, the quantity and price of the ask order should be $q^a \approx 104.17$ MWh and $p^a \approx 48$ €/MWh respectively to preserve an equal trade amount $100 \times 50 \approx 104.17 \times 48$ ($q^b \times p^b = q^a \times p^a$).

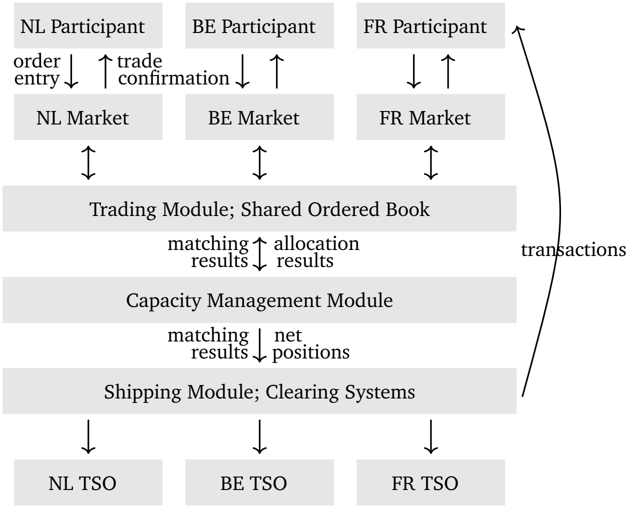


Figure 2.2: CID flow for Dutch (NL), Belgian (BE) and French (FR) participants.

capacity and a match is found, those orders are stricken from the register. Elaborating, the shared order book immediately updates itself by removing these orders from its registry, and the capacity management module updates itself by allocating implicit capacity. The matched orders are sent to the shipping module from the shared order book, where the post-coupling process is initiated. The European Commodity Clearing, which is a clearing house for all transactions, informs exchange members' transmission system operators (TSO) about transactions every fifteen minutes.

2.3.2 Trading

In Europe, intraday markets are more complex than the DAM because they have more trading procedures and product types. Elaborating, trading procedures can be auction-based or continuous-based. Orders are stored in the limit order book for both trading procedures. While orders for auction trading are cleared at fixed discrete times, orders for continuous trading are executed instantaneously when they are matched.

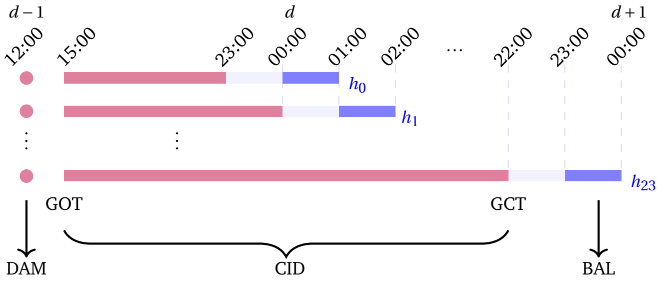


Figure 2.3: Trading timelines for all hourly contracts. Blue bars are delivery periods. Pink circles represent the auction-based DAM. Pink bars are continuous-based CID trading sessions, starting at 15:00 on the day $d-1$, a day earlier than the delivery day.

Product types can be 60-minute (hourly), 30-minute or 15-minute. Note that the hourly product has 24 tradable contracts (expiries) per day, the 30-minute product has 48 contracts, and the 15-minute product has 96 contracts. In practice, most volumes are traded with the hourly product in the intraday market [18]. Figure 2.3 further details trading timelines of the hourly electricity contracts, $h \in \{h_0, h_1, \ldots h_{23}\}$, for the intraday market with continuous trading.

The CID is a continuous market, offering hourly contracts, as shown in Figure 2.3. The start of trading and the end of trading are referred to as the gate opening time (GOT) and the gate closing time (GCT) respectively. The GCT is 60 minutes before delivery. The GOT may differ across countries. For example, the GOT is at 14:00 on the day $d-1$, a day earlier than the delivery day, for Belgium, and at 15:00 on the day $d-1$ for the Netherlands, Germany, Denmark, France, Switzerland and Austria [18]. The period between GOT and GCT is defined as a trading period or a trading session. During this trading session, orders, with attributes described below, can be continuously submitted to and cleared by the CID. Detailing the attributes:

- *order type* specifies orders either with a bid (to buy electricity) or an ask (to sell electricity).

- *quantity* is the total available volume (in MWh).

- *price limit (price)* is the price restriction (in €/MWh) for limit orders. Using this price limit, orders are matched at this price or a better price.
- *delivery area* is a participant's delivery area, such as The Netherlands.
- *expiry* defines a tradable contract, $h \in \{h_0, h_1, \dots h_{23}\}$. For instance, h_3 refers to an hourly contract for energy delivery between 3 and 4 am. As stated in [18], the expiries can be traded regularly throughout the year except for two days when daylight saving time changes. From winter to summer time change, h_2 cannot be traded. From summer to winter time change, the same expiry is divided into two hour expiries.
- *validity restriction* bounds orders by time, such as "good for session" and "good till date", or by volume such as "iceberg"[3].
- *execution restriction* bounds orders by a specific execution, such as "fill or kill" (FOK), "linked fill or kill" (LFOK), "immediate or cancel" (IOC), and "all or none" (AON).[4]

The Limit Order Book

The limit order book (LOB) at time t consists of all active ask and bid orders submitted to the CID. See an example LOB in Figure 2.4. In general, orders are removed from the LOB if they are matched, deleted or deactivated. Otherwise, orders are removed from the LOB with time-related *validity restrictions*, such as "good for session" and "good till date".

Describing basic LOB terminologies, as also shown in Figure 2.4, the best ask price p_t^a is the lowest of all available ask prices at time $t \in [1, T]$, where T is the last time step of a trading session, can be also referred to as the GCT. Meanwhile, the best bid price p_t^b is the highest of all available bid prices at

[3] Iceberg orders hide total quantity information from other participants and are used especially for large orders. Iceberg orders have two additional attributes: total quantity and initial quantity. Note that initial quantity is the tip of the Iceberg mountain, and the remaining quantity is the hidden-quantity. An iceberg order is divided into equally smaller orders (slices) and gradually introduced to the limit order book. Thus, other participants can only see the introduced slice. If the initial quantity is matched, then the next slice is introduced to the order book.

[4] The IOC, FOK and LFOK, i.e. linked FOK, orders need to be executed immediately or cancelled without entering the limit order book. While the IOC orders can be matched partially or entirely, the FOK and LFOK orders can only be matched entirely. The AON orders also need to be executed entirely. However, they stay in the limit order book until their execution or cancellation.

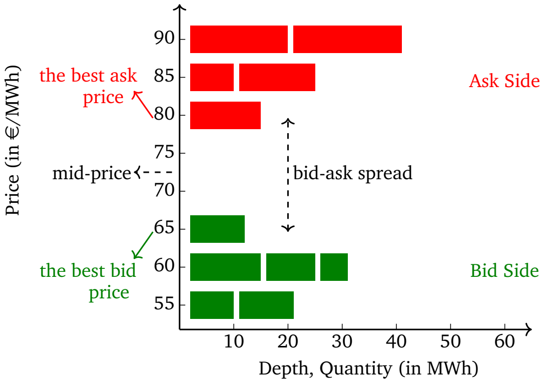


Figure 2.4: An example LOB at time t. Bid orders are shown with green blocks. Meanwhile, ask orders are presented with red blocks.

time t. Using the best ask and bid prices, the mid-price at time t is calculated as $p_t^{mid} = (p_t^a + p_t^b)/2$, and the bid-ask spread at time t is calculated as $p_t^{spread} = p_t^a - p_t^b$. The bid-ask spread can be used as a metric to evaluate market liquidity. Note that liquidity is defined as the possibility of finding a trading counterparty. The higher the bid-ask spread the lower the market liquidity.

Order Matching

During a trading session, submitted orders are automatically matched based on the first-come-first-served principle, which prioritises the best price and order entry time. When a new order is submitted, orders are matched if the submitted bid price is higher than the best ask price, or if the submitted ask price is lower than the best bid price. The transaction occurs based on the lower quantity and any remaining quantity is kept in the limit order book. If multiple ask or bid orders have the same price, the order matching engine prioritises time, by executing earlier entered orders.

2.4 Balancing Markets

The real-time balancing market (BAL), also referred to as the imbalance or regulation market, is built to keep the electricity system balanced. The BAL, managed by a transmission system operator (TSO), settles imbalanced positions for each delivery period with balance responsible parties (BRPs).[5] Note that a delivery period is defined as 15 minutes in the Netherlands, resulting in a total of 96 delivery periods across a 24h window [19].

For any delivery period, after trading on the DAM and/or the CID, a BRP might still hold an outstanding open position. If a BRP has a remaining long position, i.e. positive imbalance or imbalance surplus, usually caused by generating more electricity than is scheduled or consuming less than is scheduled, then this BRP receives the long imbalance price, i.e. feed price, for its long position. If a BRP has a remaining short position, i.e. negative imbalance or imbalance shortage, usually caused by generating less electricity than is scheduled or consuming more than is scheduled, then this BRP pays the short imbalance price, i.e. take price, for its short position.

Imbalance prices are calculated through a single-pricing scheme in the Netherlands [19]. Elaborating, if upward or downward reserves are activated, the activation cost of reserves is used for both imbalance shortages and imbalance surpluses, and BRPs pay or receive the same price. If, however, both upward and downward reserves are activated, the activation cost of upward reserves is used for imbalance shortages and the activation cost of downward reserves is used for imbalance surpluses, resulting in different imbalance prices. Moreover, when the system imbalance is high, a balance incentivising component is used to weigh imbalance prices. A BRP is punished if its position is in the same direction as the system imbalance. Otherwise, a BRP is awarded.

2.5 Concluding Remarks

In this chapter, the background knowledge required for visualising and analysing electricity markets was outlined. Motivations for visualising electricity markets, and descriptions of each short-term market layer were provided. The next chapter of the thesis uses this background information to build novel visualisation methods for the CID. Note that the background information about short-term electricity markets is also used in the following parts of the thesis to develop novel price forecasting methods and trading strategies.

[5] The BAL could be referred to as a market where a TSO trades with flexible players to adjust the balance of the power system, e.g. tertiary control (mFRR). However, in this thesis, the BAL is referred to as a market where a TSO settles BRPs' open positions, i.e. the imbalance settlement.

Chapter 3

Exploring the Continuous Intraday Market

The European continuous intraday market (CID) trading activity has increased substantially in recent years mainly as a consequence of the increasing penetration of renewable energy production and its subsequent impact on imbalance market prices. Nonetheless, more research is needed to understand this growing market. This chapter presents exploratory visual analytics tools for tradable contracts of the CID. The main visualisations are created from the limit order book (LOB) and trade book (TB), and intended to increase our domain knowledge of the CID by monitoring market trends, behaviours, depth, price consensus and liquidity. Furthermore, previous contracts of CID volumes and balancing prices are visualised to identify trading opportunities and risks. We expect that the presented visual analytics will be useful for both practitioners and researchers seeking quick and easily implementable tools for acquiring additional market insights and developing manual or automated trading strategies.

3.1 Introduction

CID trading activity has significantly increased in recent years, in line with increasing renewable energy production. To understand why, the expected impact of increased renewable production on balancing energy need must be explained. With this in mind, note that: firstly, a high demand on balancing power results from large differences between expected and actually delivered energy consumption/production within a grid area; secondly, renewable energy production is relatively unpredictable. Putting the above together, it should be apparent how increased penetration of renewable energies could be expected, ceteris paribus, to increase balancing energy needs [20]. The CID offers a counterweight to the above, preventing balancing energy needs from rising [21].

From the market participants' perspective, trading activity in the CID has increased mainly because participants aim to: hedge their position against balancing market (BAL) prices, benefit from the flexibility of the CID trading period, and reach the available liquidity of neighbouring countries. Dissecting these aims in turn, after the closure of the day-ahead market (DAM), participants often find themselves with an energy gap, requiring the revaluation of their bids and asks. This gap can arise because of DAM forecast errors or unpredictable intermittent renewable energy supply spikes. Given updated and more accurate energy forecasts, practitioners can use the CID to balance their energy gap and further hedge their risk exposure to highly uncertain BAL prices.

Examining the flexibility of the CID trading period, CIDs offer continuous trading opportunities starting from the closure of the DAM to shortly before delivery. This flexibility helps participants to continuously optimise their portfolio and update their planning decisions over an extended period of time. Unlike with the DAM, participants can trade multiple times. The producer may decide to buy energy whenever it becomes cheaper than his/her production cost.

Finally, elaborating on the opportunity of reaching the available liquidity of neighbouring countries, note that the liquidity is defined as the possibility of finding a trading counterparty. Since 2018 the CID has been operating to offer market participants the available liquidity of other countries. For instance, Danish wind producers may not able to sell their energy inside Denmark due to a significant difference between the bid and ask volumes for hours with high wind power production. The CID offers these producers an opportunity to export energy to other countries which have more favorable market conditions.

CID trading volumes are expected to increase even further in the upcoming years since the penetration of renewable energy sources is expected to rise [14]. This growth will further increase the liquidity and importance of the CID, and attract not only more practitioners but also more academic researchers. We expect that practitioners and researchers will be more aware of the trading opportunities on the CID and create more sophisticated optimisation processes taking into account the increased volatility of electricity prices and the increased penetration of renewable sources on electricity markets.

Above, participants' increasing interest in CID trading is explained; however, how participants make their trading decisions (e.g. which trading contract to trade, when to submit orders, or how to price orders) has not been discussed yet. To reach trading decisions, practitioners continuously monitor fundamental indicators including market behaviours, price drivers and market insights such as the market depth, bid-ask spread and price consensus. Exploratory visual analytics offer practitioners tools for effectively visualising these fundamental indicators to be used in their development of trading strategies. Beyond manual trading, practitioners can also use these fundamental visualisations to acquire market insights prior to setting their automated trades.

With the above in mind, in this thesis we outline exploratory visual tools for tradable CID contracts. We develop tools for visualising transactions, executed trades, the volume-at-price, and distributions of prices and volumes. We also identify easily implementable tools for succinctly visualising previous hour risks using CID trading volumes and balancing prices. These visualisations are created with Python software using the CID LOB and TB, and balancing prices.

3.2 Visual Analytics Approach to Continuous Intraday Market Exploration

3.2.1 Data

Our primary data sources are the TB and LOB of the CID, available for the Dutch market area. The LOB includes all submitted orders entered with an entry time, price and quantity. The TB consists of all executed trades with their buy/sell areas, matched times, prices and quantities. CID trade and LOBs are not publicly available. The data was acquired by Scholt Energy, an energy supplier, for 2018.[1] Additionally, DAM prices for The Netherlands and Belgium,

[1] After 2018, note that several incidents, such as the Covid-19 pandemic and the Russia-Ukraine conflict, have affected the CID. Thus, our observations in this chapter may or may not be observed during these deviating times.

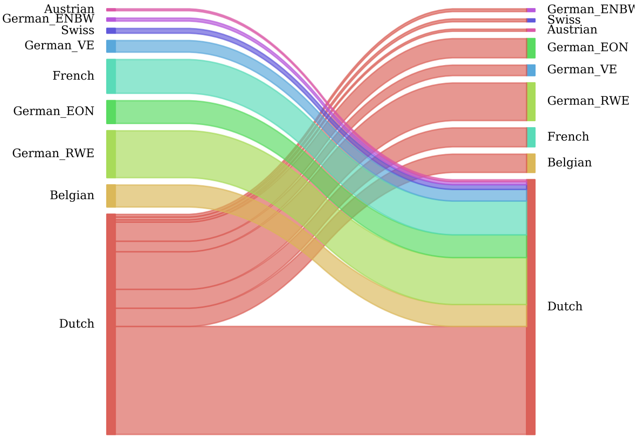


Figure 3.1: Transaction flow graph showing the number of CID trades from and to the Dutch market area in 2018. Buy areas are on the left side and sell areas are on the right.

and BAL prices for The Netherlands are collected. These prices are publicly available at the ENTSO-E Transparency Platform [22].

3.2.2 Visualisations for Transactions

Domain knowledge is a very crucial strength for any market participant. Most of CID participants follow import and export trading behaviours of their neighbouring countries. Figure 3.1 presents a Sankey diagram [23] showing transaction flows for the Dutch market area. It is clear from this figure that for cross-border transactions, Dutch participants exchanged electricity mostly with Germany. This might indicate that features impacting German markets may also significantly impact CID prices. By visualising transactions, participants can gain insights about potential price drivers affecting CID prices.

While only transactions for a year are presented, it should be noted that plotting transactions for each season across several years could expose further insights about countries' trading behaviors and the price drivers. This is because, trading behaviors can exhibit seasonalities only identifiable across a longer term period.

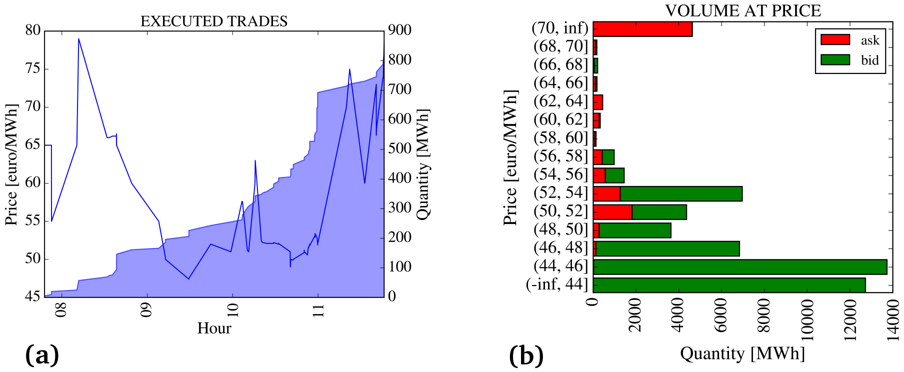


Figure 3.2: Executed trades and volume-at-price graphs for the Expiry-1, 16/03/2018. Figure (a) displays trades on the y-axes - prices (the left side) and accumulated volumes (the right side) - and timestamps on the x-axes. Figure (b) shows all submitted orders using price bins on the y-axes and volumes on the x-axes.

3.2.3 Visualisations for the Contract History

Contract history visualisations can be used to gain insights about the liquidity of a market or to develop trading strategies. In this section, the visualisations are presented for the Expiry-13, i.e. h_{13} an hourly contract for energy delivery between 1 and 2 pm, on 16/03/2018. They are however applicable to any expiry for any day to gain insights about the CID.

Firstly, the TB history is investigated for contract history visualisations. Executed trades, extracted from the TB, are plotted in Figure 3.2a. Prices and the cumulative volume are plotted together to facilitate a combined analysis. By monitoring executed prices and volumes, practitioners can explore the market behaviour, or deduce the possible remaining trading volume together with its price range. Figure 3.2a shows that overall the energy was exchanged in small quantities. The highest volume of trades happened around 11 am; impacting prices which increased from 50 to 75 €/MWh. While the majority of transactions occurred between 9 and 11 am and were traded in a narrow price band [47, 60] €/WWh, a small number of transactions occurred either before 9 am or after 11 am and were traded in a higher price band [55, 79] €/MWh. This difference may be explained by a change in traders' overall risk tolerance e.g.

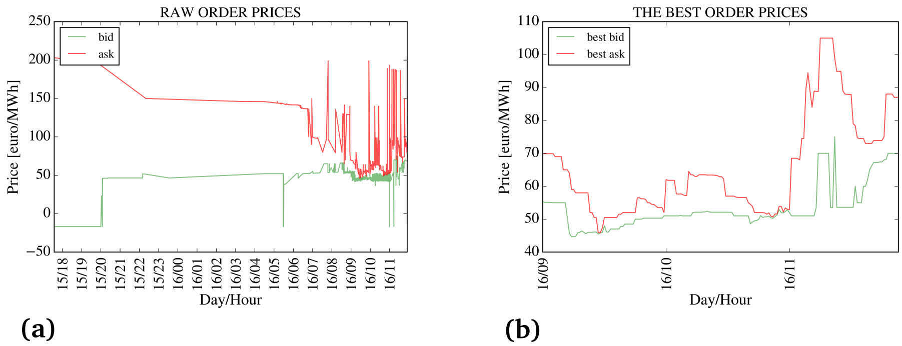


Figure 3.3: Raw and best order price graphs for the Expiry-13, 16/03/2018. Figure (a) illustrates prices of all submitted orders throughout the trading session. Note that orders are plotted based on their timestamps; however, minutes and seconds are not displayed in the x-axis for simplicity. Figure (b) shows only the best ask and bid prices for the final hours of the trading session.

risk-averse for trading before 9 am and risk-taker for trading after 11 am. Using executed trades plots, practitioners can thus anticipate price changes by monitoring transactions and gauging the markets' risk appetite. An additional observation from Figure 3.2a is that transactions only occurred in several last hours of the trading session. This is also one of the CID trading behaviours deduced using ELBAS data by [14].

Moving from visualising the TB to visualising the LOB, practitioners can monitor the depths of bids and asks, and potential price consensuses using Figure 3.2b, a volume-at-price graph, also known as a market depth graph. The volume-at-price graph visualises volumes of all available orders for each price band. A price band, e.g. (54, 56] €/MWh, is considered as a price consensus of the market if it includes both bids and asks. In practice, volume-at-price graphs typically have heavy tails. While the upper heavy tail - the most expensive price band - consists of a considerable proportion of asks, the lower heavy tail - the least expensive price band - consists of a considerable proportion of bids.

Figure 3.2b presents an CID volume-at-price graph with all submitted orders extracted from the LOB. The figure illustrates that Expiry-13, as expected, has heavy tails. The upper tail - the price band of (70, ∞) €/MWh - consists of asks and the lower tail - the price bands of (-∞, 44] and (44, 46] €/MWh - consists of bids. Between these two tails, various price consensus, i.e. price bands which include both order types, are observed. The most dominant price bands of them all, however, are the price bands of (50, 52] and (52, 54] €/MWh with their higher quantities.

Volume-at-price graphs are very useful tools to investigate the market depth. They, however, fail to visualise a very important component: entry times of orders. One of the reasons practitioners may wish to track orders with their entry times is that it allows them to visualise any changes in the market trend. Figure 3.2b fails to capture changing market trends. Raw and best order price graphs, Figure 3.3, and quantile price and volume graphs, Figure 3.4, are generated to address this shortcoming. Together they can be used to track the CID's trend and liquidity by integrating orders' entry times into the visualisation. Figure 3.3a shows prices of all submitted orders based on their entry times. It can be observed that the price spread between ask and bid orders is very distinguishable and the market liquidity is very low in early hours. This leads us to concentrate on actively traded hours which are the last hours of the trading session. Figure 3.3b is based on the best bid and ask prices, respectively the highest and lowest prices of their types. These have major impact on matching, their plot shows the most important information about the LOB and probably is the most commonly used diagram among practitioners. Note that the best prices are calculated every minute after orders have been executed. Figure 3.3b shows higher ask prices for the final hour. A possible explanation for this might be that sellers increase prices after seeing demand from buyers, the high amount of bid volumes in Figure 3.2b. Increased ask prices force bid prices to increase, as shown in Figure 3.3b.

To gain insight into future market price movements, we propose to visualise quantile bids and asks. Once best bids and asks are matched, significant price swings can ensue, if there is a lack of liquidity, immediately after matched orders are removed. This leads us to plot Figures 3.4a and 3.4b, the distributions of prices and volumes with their important quantiles. Again note that the quantiles are calculated every minute after orders have been executed. Figures 3.4a and 3.4b show that the final hour is less liquid than the previous two hours; a wider bid ask spread and less available quantity signify this. Figure 3.2a supports this conclusion; fewer trades are executed during the final hour. Considering that sellers with a large portfolio have less risk to be balanced in the BALs, they may be able to take more risk in the final hour of trading because of a potentially high/skewed reward to risk ratio. In other words, they might prefer waiting to

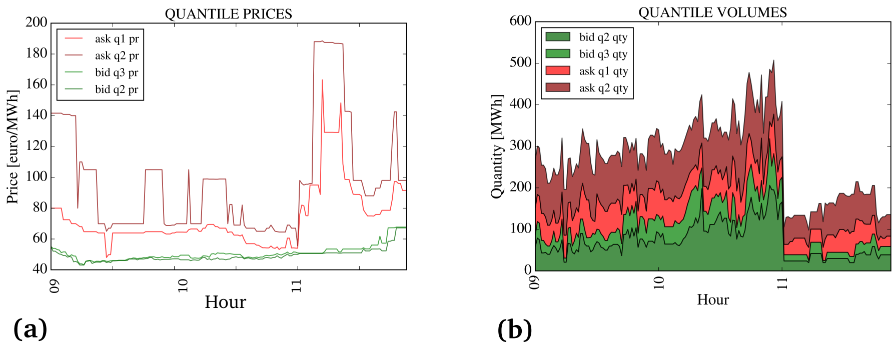


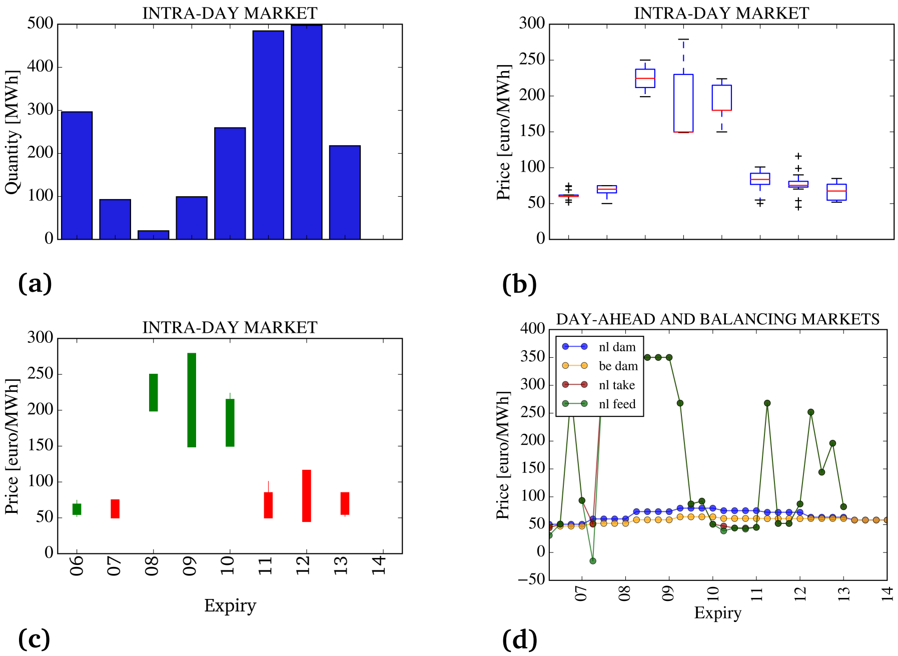
Figure 3.4: Quantile price and volume graphs for the Expiry-13, 16/03/2018. The lower quartile (q1), median (q2), and upper quartile (q3) are displayed for each graph. Figure (a) presents ask and bid price quantiles of all available orders for the latest hours. Figure (b) illustrates the total ask and bid volumes of price quantiles for the same period. The x-axis is shared across subplots.

sell their energy at higher prices. Using the distributions of prices and volumes, participants could, then, monitor the market trend, behaviour and liquidity.

3.2.4 Visualisations for the Previous Hour Risks

Market participants submit orders while weighing both risks and rewards. They usually begin this process by examining the volume and price history of previous expiries, and the price history of the BAL. For instance, a power generator might find an opportunity if previous hour CID expiries show high executed prices - significantly higher than their corresponding DAM prices - and low total volumes. It is likely that the upcoming expiry would show the same price and volume trends. This producer could thus decide to produce some additional energy for the upcoming expiry. This decision is, however, made while considering the balancing price risk. The balancing price of the upcoming expiry might show the same trend as several previous hours. If previous balancing prices are very high, the producer incorporates this risk into his or her calculations.

Figure 3.5 illustrates an overview of electricity markets for the previous expiries of the Expiry-13, 16/03/2018. 8 previous hours are displayed. Note that, prior to trading, neither CID prices and volumes, nor balancing prices for the Expiry-13, would be available. Consequently, CID's volume for the Expiry-13 is not



(a)

(b)

(c)

(d)

Figure 3.5: Previous hour visualisations for the Expiry-13, 16/03/2018. 8 previous hours, from the Expiry-06 to Expiry-13, are displayed. Figure (a) shows the total executed volumes. Figure (b) presents a box plot of executed prices. Figure (c) illustrates the same information, but with Japanese candlesticks. Figure (d) exhibits prices of DAMs, Dutch (nl) and Belgian (be), and the balancing prices, take and feed. The x-axis is shared across subplots.

displayed in Figure 3.5a, CID's prices for the Expiry-13 in Figures 3.5b and 3.5c, and balancing prices for the Expiry-13 in Figure 3.5d. Only DAM prices are available prior to the Expiry-13 being traded and they are thus displayed in Figure 3.5d. Figure 3.5b visualises the distribution of executed prices using multiple box plots. Each box plot highlights the median (red line) and interquartile range using its main body, and the range and outliers using its whiskers. Meanwhile, Figure 3.5c visualises the progression of trading using Japanese candlesticks [24]. Each candlestick highlights the open and close price using its main body, and minimum and maximum price using its whiskers. A red candlestick represents a price drop, while a green candlestick represents a price increase.

Together Figures 3.5b and 3.5c can be used to gain both statistical and technical insights about price movements. For example, from Figures 3.5b and 3.5c it can be discerned that Expiry-09 opened at a price of 150, also its median price, and closed at a price of 290, also its maximum price.

Examining the previous hour risks of the Expiry-13, the contract is expected to follow a similar price trend to its previous contracts, Expiries-11-12-13. Using Figures 3.5b, 3.5c and 3.5d a low CID price risk but a high balancing price risk is expected. Specifically, the Expiry-13 prices are assumed to follow a moderate interquantile range of [50, 80] €/MWh and a red candlestick. Figure 3.2a confirms our expectations for the price range but not for the candlestick.

Figure 3.5 also shows that the Expiry-08 and Expiry-09 have low CID volumes, high CID prices - far greater than hour 8 and 9 DAM prices - and high balancing prices. Our aforementioned power generator might find an opportunity and decide to produce some additional energy for the Expiry-10. The presence of this opportunity may explain why the volume doubled for that expiry. The market behaviour of previous hours is an important input of trading decision mechanisms for most participants. By visually inspecting previous hours, participants can explore opportunities and risks.

3.3 Conclusion

We developed visualisations for tradable contracts of the European single intraday coupled electricity market (CID). Specifically, we visualised transaction flows and contract histories; presenting executed trades, the volume-at-price and the distributions of prices and volumes. While our contract history plots are illustrated for the end of the trading session, they can be used for any timestamp throughout the session. Participants can thus continuously monitor the market trend, behaviour, depth, price consensus and liquidity. Additionally, we visualised CID trading volumes and balancing prices of previous hour contracts to highlight a contract's trading risk. Using this plot, we highlighted how potential trading opportunities as well as risks can be identified prior to trading.

Part II

Forecasting Electricity Market Prices

Chapter 4

Background: Forecasting Electricity Market Prices

This chapter provides background information about electricity price forecasting: the second part of the thesis. Specifically, the chapter performs a review of existing electricity price forecasting literature, and briefly describes models utilised in this thesis.

4.1 Introduction

The accuracy of price forecasts directly impacts the profitability and risk associated with trading decisions. The higher the forecasting accuracy, the lower the price uncertainty, and by extension the lower the market risk associated with each trading decision. Ceteris paribus, to reduce price uncertainty in statistical arbitrage trading strategies and increase profits of statistical arbitrage trading agents, novel forecasting methods which increase forecasting accuracies of short-term electricity market prices are needed.

Outlining the structure of the remaining chapter, Section 4.2 describes the literature review. Then, Section 4.3 introduces our benchmark models: autoregressive forecasting models. Section 4.4 offers brief descriptions of machine learning models. Finally, Section 4.5 concludes the chapter.

4.2 Literature Survey

The prevalence of non-renewable energy sources meant that forecasting studies of short-term electricity markets historically predominantly focused on the day-ahead market (DAM). Underlying the reasons why, stable supply meant that the final intersection of supply and demand could reasonably be predicted, with high accuracy, 24-hour-ahead of delivery. The recent transition from non-renewable to renewable sources of energy, however, is changing the status quo. Traded volume and liquidity on the continuous intraday market (CID) is growing, and the number of electricity price forecasting studies focusing on the CID is increasing. Below, recent examples of electricity price forecasting studies for the DAM, CID and balancing market (BAL) are outlined.

4.2.1 Forecasting Day-Ahead Market Prices

Several papers, including [10, 25–36], have studied DAM price forecasting. In [29, 30], lagged DAM prices and exogenous variables, such as load forecast, were found to be important features for forecasting DAM prices. In [27], lagged DAM prices were used together with neighbouring market prices. Using Belgian and French DAM prices, the forecasting performance of neural networks (NNs) was improved. Similarly, data from multiple markets were stacked together in [32]. The forecasting performance of NNs was also improved.

In [25, 28], the forecast accuracies of deep, ensemble, and statistical models were evaluated. Machine learning models were found to outperform statistical models in both studies. Elaborating, in [28] for example, NNs were found to outperform long short-term memory networks (LSTM) and gated recurrent units (GRUs) in forecasting Belgian DAM prices.

Other studies, such as [10, 31], have also compared the performances of deep and shallow machine learning models. In [10], deep models were found to outperform shallow models in forecasting Belgian DAM prices. In [31], GRUs were found to outperform all other neural network structures, such as NNs and LSTMs, and statistical methods, such as seasonal autoregressive integrated moving average (SARIMA) models and Markov models, in forecasting Turkish DAM prices. Contrasting performances of artificial networks, such as NNs, GRUs and LSTM, across [28, 31], differences in performance can be explained by model complexity, data characteristics, or training set sizes.

In [34–36], hybrid models were shown to outperform individual benchmark models. In [34], the hybrid model was harmonised using a wavelet transformation, an autoregressive moving average model, a kernel extreme learning machine (KELM), and self-adaptive particle swarm optimisation (SAPSO). In [35], the hybrid model was constructed with variational mode decomposition, SAPSO, SARIMA, and deep belief networks. Model accuracies were assessed across three different DAMs in [35]. Exogenous variables, however, were not included in this study. In [36], the hybrid model was structured using a local forecasting paradigm, a general regression neural network, coordinate delay and a harmony search algorithm. Unlike in [35], exogenous variables were included in this study. Model accuracies, however, were assessed only on one DAM in [36]. In [33], applications of spike forecasting were further explored. Using the Borderline-SMOTE method to balance the number of samples in different target classes, [33] increased the number of spike samples in training data; yielding DAM forecasting accuracy improvements. For a more detailed review of the DAM forecasting literature/best practices, we recommend interested readers read [25, 26].

4.2.2 Forecasting Continuous Intraday Market Prices

Several studies, such as [37–46], have studied CID price forecasting. Most of these studies investigated CID price drivers: including lagged CID prices, hour of the day variables, and day of the week variables. Exogenous variables, such as DAM prices, forecasts of load, generation, wind and solar power, as well as forecast errors were also investigated. Listing the variables that were found to have the most explanatory power, recent historic CID prices and DAM prices were found to be among the most important variables for forecasting German CID prices by [37]. Wind power forecast errors were found to be the most important features for the same market by [38].

The effect of wind and solar power forecasts was further investigated by [40, 41]. A 1 GWh error in wind power forecasts was found to have a greater influence on German CID prices than a 1 GWh error in solar power forecasts by [41]. Findings of [41] show that a 1 GWh negative wind power forecast error can be expected to raise the CID price by approximately 7 €/MWh at night and around 4 €/MWh during the day.

In [43], a model employing fundamental features as inputs was found to be able to explain roughly 75% of the CID price variance. The findings of [43] show that if the current residual load value is 1 GWh lower than the average load across the past four periods, then CID prices can be expected to fall by 0.295 €/MWh. Otherwise, they can be expected to rise by 0.676 €/MWh.

The forecast accuracies of LSTM networks and shallow machine learning models, such as extreme gradient boosting (XGB) and random forest (RF) models, were evaluated across the German market by [45]. LSTMs were found to outperform a RF model and slightly outperform XGB. Neural network-based forecasting models together with Lasso and linear regression models were further evaluated across the Turkish market in [46]. Neural network-based models were found to outperform the least absolute shrinkage and selection operator (LASSO) and linear regression models. Among neural-network models, recurrent neural networks were found to outperform artificial neural networks, meanwhile GRUs were found to outperform LSTMs.

4.2.3 Forecasting Balancing Market Prices

Compared to the DAM and CID, significantly fewer electricity price forecasting studies have focused on the BAL. This is because BAL prices are, by design, unpredictable; resulting from unforeseen stochastic price imbalances. Several studies, including [47–50], have nevertheless researched BAL price forecasting. In [47], the explanatory power of nineteen variables: including DAM prices, settlement periods, as well as load, production, wind power, and solar power, was investigated. For the UK market, the net imbalance volume, aggregated LOLP, aggregated de-rated margins, and month dummies were found to be among the most important features, with weights of 28.6%, 27.5%, 14.0%, and 8.9% respectively. In [48], benchmark models, such as autoregressive moving average (ARMA), autoregressive model (ARX) with exogenous inputs and naive Bayes, were investigated for the Nordic NO2 BAL. Models which include the balancing state feature were found to perform better than models without the feature.

In [47], the forecasting accuracies of GB, RF and XGB models were evaluated. XGB was found to outperform other machine learning models. In [49], a two-step probabilistic approach was evaluated. Belgian BAL prices were predicted after computing the net regulation volume state transition probabilities. This approach was found to outperform a probabilistic benchmark using the Gaussian processes and a deterministic benchmark using multi-layer perceptrons.

4.3 Autoregressive-Exogenous Models (ARXs)

ARXs are linear time series models capable of capturing both autoregressive and exogenous relationships driving stochastic processes. Formally, a single output ARX(n, m) model is defined according to (4.1).

$$y_t = a_0 + \sum_{i=1}^{n} a_i y_{t-i} + \sum_{i=1}^{m} b_i u_i + v_t, \tag{4.1}$$

where y_{t-i} is the model output at time $t-i$, u_i is i^{th} exogenous input, a_i and b_i are the i^{th} autoregressive and exogenous parameters respectively, and v_t is the residual noise. To train an ARX, LASSO is employed; optimising **a** and **b** according to (4.2).

$$\underset{\mathbf{a},\mathbf{b}}{\arg\min}\, \mathcal{L}(\mathbf{a},\mathbf{b};\mathbf{y},\mathbf{u}) = \mathbb{E}_{y_t \sim p_{\mathbf{y}}(y_t)}\left[\left(y_t - (a_0 + \sum_{i=1}^{n} a_i y_{t-i} + \sum_{i=1}^{m} b_i u_i)\right)^2\right] + \tau\left(\| \mathbf{a} \|_1 + \| \mathbf{b} \|_1 \right), \tag{4.2}$$

where τ is a regularisation parameter.

4.4 Machine Learning Models

4.4.1 Linear Regression (LR)

The most fundamental of linear models, LR [51] fits a straight line for a predictor, through a series of points by minimising the sum of squared errors between its targets and predictions. LR is sensitive to outliers and correlated features. Nevertheless, as LR is one of the primary machine learning models used for forecasting, it is included in our examination.

4.4.2 Huber Regression (HR)

Extending LRs, HR [52] is a linear model robust to response variable outliers. Unlike LR, HR optimises both an absolute and squared loss function, reducing the impact of outliers. To switch between loss functions, HR uses an epsilon hyperparameter. Despite implementing an enhanced optimisation procedure, HR remains sensitive to explanatory variable outliers and correlations.

4.4.3 Random Forest (RF)

RF [53] fits several decision trees on random samples of the data and averages them to obtain a final result. Individual trees are fit by recursively splitting the data in such a way that maximises the information gain.

4.4.4 AdaBoost (AB)

AB [54] is an adaptive boosting algorithm used to sequentially train an ensemble of weak learners. The algorithm begins by fitting a weak learner, and continues by training copies of this learner, placing a greater instance weight on incorrectly predicted values. The algorithm proceeds until the final model, a weighted sum of all trained weak learners, becomes a strong learner. Note that the AB algorithm used in this thesis trains an ensemble of decision trees.

4.4.5 Gradient Boosting (GB)

Another boosting algorithm, GB [55] focuses on sequentially improving model predictions by fitting copies of learners to residuals. Residual predictions are repeatedly added to model predictions until the sum of residuals stops decreasing. Similarly to AB, the GB algorithm utilised in this thesis trains an ensemble of decision trees.

4.4.6 A Scalable Tree Boosting System (XGB)

XGB [56] is yet another boosting algorithm and aware of sparsity and builds a tree based on this awareness. The key point of Sparsity-aware splitting is that the algorithm finds its way to only presence entries as a default. Chen et al. [56] found that Sparsity-aware splitting can make an algorithm 50 times faster than the simple gradient algorithm.

4.4.7 Deep Neural Networks

Deep models, or artificial neural networks (ANNs), consist of an input, output, and hidden layer. Of these three, the hidden layer is the most varied, with architectures differing in depth and layer make-up. Below, we describe some of the most fundamental layers and modules used in deep models.

Fully Connected Layer (FCL): Fully connected neurons, comprising of a linear regression with an added non-linearity [57], are stacked to build an FCL. FCLs can be used to approximate any continuous function [58], explaining why, with increasing computational power, they are frequently used in state-of-the-art price predictors.

ANN layers comprise of groups of interconnected nodes. Each node weights the outputs of previous layers according to $f(b+\mathbf{w}^{\mathsf{T}}\mathbf{z})$, where b is a bias term, $\mathbf{w}$ is a vector of trainable weights, $\mathbf{z}$ is a vector of previous layer outputs, and $f(.)$ is a differentiable activation function, such as a rectified linear unit (ReLU), capable of adding a non-linearity to a node's output. In an FCL, nodes in layer l are connected to all other nodes from layer $l-1$; computing the dot product of their outputs according to $f(b+\mathbf{w}^{\mathsf{T}}\mathbf{z}_{l-1})$. Each FCL yields $\mathrm{P}_{l-1}\times\mathrm{P}_l$ optimisable parameters, where P_l and P_{l-1} are the number of nodes in layers l and $l-1$.

Convolutional Layer (CONV): Locally connected neural networks, or convolutional neural networks (CNNs) [59], are used for feature mapping/extraction. The primary module used by these networks, CONV, works by sliding equally sized filters with trainable parameters across input data producing 2D activation maps. While FCLs tune the parameters of every neuron, CONVs implement parameter sharing to remain computationally feasible. Overall, CONVs have proved adept at identifying features in images [60] and time series [61], making them potentially very powerful modules for price forecasting.

In a CONV, multiple cascaded convolutions are applied across $\mathbf{z}_{l-1}$. Formally, kernels of size $(\mathrm{W}\times\mathrm{H})$, computing $f(b+\sum_{k}^{\mathrm{D}+k}\sum_{j}^{\mathrm{W}+j}\sum_{i}^{\mathrm{H}+i} w_{kji}z_{kji})$, where D is the depth of $\mathbf{z}_{l-1}$, are passed across $\mathbf{z}_{l-1}$. Unlike FCLs, CONVs implement parameter sharing, only optimising $\mathrm{C}\times\mathrm{W}\times\mathrm{H}\times\mathrm{D}$ parameters, where C is the number of CONV kernels.

Residual Module: A residual module, ResNet [62], adds the inputs from one module to the outputs of another module. It thus creates a direct identity mapping in a network between module inputs and outputs, combating both the vanishing gradient problem and the degradation problem, which otherwise impede the training of deep networks.

4.5 Concluding Remarks

In this chapter, the background knowledge required for forecasting electricity prices was summarised. The information covered is utilised in the ensuing chapters to build novel forecasting methods. The focus was placed on briefly describing the motivation for forecasting short-term electricity prices. Moreover, a literature review of forecasting studies, for each short-term market layer, was performed. Finally, various forecasting models were briefly explained.

Chapter 5

Forecasting Day-Ahead Market Prices: Technical Indicators

Day-ahead electricity market (DAM) volatility and price forecast errors have grown in recent years. Changing market conditions, epitomised by increasing renewable energy production and rising intraday market trading, have spurred this growth. If forecast accuracies of DAM prices are to improve, new features capable of capturing the effects of technical or fundamental price drivers must be identified. This chapter focuses on identifying/engineering technical features capable of capturing the behavioural biases of DAM traders. Technical indicators (TIs), such as Bollinger Bands, Momentum indicators, or exponential moving averages, are widely used across financial markets to identify behavioural biases. To date, TIs have never been applied to the forecasting of DAM prices. This chapter demonstrates how the simple inclusion of TI features in DAM forecasting can significantly boost the regression accuracies of machine learning models. Moreover, tailored TIs are identified for each of these models, highlighting the added explanatory power offered by technical features.

5.1 Introduction

DAM prices have historically been driven by fundamental drivers or fundamentals[1]; reflecting an intrinsic demand and supply of electricity. However, in recent years, regulatory changes, such as [63], have swept through markets provoking disruptive propagotary shocks. These shocks, which culminated from a growing need to generate cleaner and safer energy [64, 65], have indirectly boosted the impacts of technical price drivers or technicals[2] on the DAM and moved prices further from their intrinsic values. As a result, the identification of new model features, capable of capturing the residual impacts of technicals, has become necessary to forecast DAM prices accurately.

Because traders' decisions are not always perfectly rational, a greater willingness to distribute orders across day-ahead and intraday markets accentuated the impacts of technicals; moving prices away from their intrinsic values. Exemplifying this point, note that all trades on electricity markets are placed because of an underlying stochastic energy need. Whereas before the boom in renewable generation, traders almost exclusively utilised the DAM to meet this need in the short-term - resulting in planning decisions rooted principally in fundamentals - today, intraday opportunities offer traders many more ways to maximise their respective reward to risk ratio - resulting in more heterogeneous short-term planning decisions and the proliferation of technicals. Day-ahead and intraday substitution can thus be understood to have heightened the impacts of technicals on DAM prices.

To capture the impacts of technicals, an approach to forecasting future price movements using historic market data is needed. Technical analysis (TA) [66], widely used by practitioners to identify investment opportunities across financial markets, offers such an approach. Rooted in the theories of behavioural finance [67], TA, an analysis methodology, focuses on analysing statistical trends in historical market data to forecast future price movements. TA was established around three assumptions: prices move in trends; history repeats itself;

[1] Fundamentals are associated with the intrinsic value of a good, commodity, or security. Examples include goods' production costs, economic variables, etc. Fundamentals would drive prices in a world of perfectly rational investors, i.e. investors that always optimally maximise their utility.

[2] The term "technical" comes from the discipline of technical analysis (TA) [66]. Technicals are factors which move prices away from their intrinsic values. They result because in practice investors are not perfectly rational; investors often make seemingly sub-optimal decisions. Behavioural economics focuses on explaining how investors' biases, emotions, and other psychological factors influence decisions and, by extension, prices.

studying price fluctuations allows the prediction of future shifts in demand and supply [66]. In short, TA assumes that technicals form reoccurring patterns in market data which careful analysis can identify and predict. One of the ways in which TA identifies emerging price patterns is using technical indicators (TIs). TIs transform market data using various formulas. By transforming the data, TIs can facilitate the identification of complex price patterns; signalling when securities are overbought or oversold, when prices deviate from a central trend, or when they are near support and resistance lines - levels that prices bounce off of and break through. Moreover, in conjunction with price charts, TIs can provide leading indications of future price movements, offering additional explanatory power to models forecasting financial time series. Numerous studies, such as [68–70], have extensively demonstrated the explanatory power of TIs across stock markets. Given that technicals result from universally exhibited behavioural biases among investors, such studies motivate our examination of TIs with the DAM.

Summarising the principle contribution of this chapter, to the best of our knowledge, we are the first to evaluate the explanatory power of TIs in DAM price forecasting. This chapter demonstrates that the simple inclusion of TI features can significantly reduce linear, ensemble, and deep model regression forecast errors. Describing the structure of this chapter, examined TIs and the methodology are introduced in Sections 5.2 and 5.3 respectively. Subsequently, in Section 5.4 results are presented. Finally, in Section 5.5 the chapter is concluded.

5.2 Technical Indicators

While a plethora of TIs exists, none can guarantee the addition of explanatory power. This is because the success of individual TIs is domain and period-dependent. For instance, price-based trend-following indicators may be useful in times of high autocorrelation; however, they cease being indicative of future price moves when autocorrelation vanishes and trends become spurious [71]. With this in mind, TIs satisfying the following two criteria are chosen.

1. TI calculation only requires *close* prices.

2. TI inclusion is likely to improve predictive performance by highlighting oscillations or trends in DAM prices.

Criterion 1 is imposed by the properties of the DAM – neither *high*, *low*, nor *open* prices are available.[3] Criterion 2 stems from our desire to avoid examining overly intricate TIs which have been optimised for rare use-cases and evaluate TIs which are well suited for the DAM. DAM prices typically move between horizontal support and resistance lines, behaving comparably to ranging markets whose prices move within a band. On occasion, however, DAM prices breakout from this price band, and establish a trend; behaving comparably to trending markets. Overall, such varying behaviour spurs us to evaluate both oscillators, which vary around a central line or within set levels, and trend-following TIs, which measure the direction of a trend. The former TIs are well suited for ranging markets, while the latter TIs are optimised for trending markets.

Together Criteria 1 and 2 allow us to follow the suggestions of [69] by focusing our research on the simplest and most widely used TIs, which are likely to introduce more explanatory power than noise. Below, our list of chosen TIs are introduced and formulas for their calculation are provided. Note that throughout the chapter the following notations are used: t time, p_t price at time t, n lag-factor for $\forall n \in \mathbb{Z}$, s span for $\forall s \in \mathbb{Z}$, where n and s are hyperparameters tuned using grid-search during the modelling of DAM prices.

5.2.1 Simple Moving Average (SMA)

The SMA [66], a type of moving average, is the most fundamental TI. It is often used as a building block in the calculation of other compound indicators such as Bollinger Bands (BBANDs). The SMA captures trends by smoothing a price series using a lag-factor, n. A single SMA curve, either alone or in conjunction with the price series, can be used to forecast future price movements and generate trading ideas [73]. Specifically, the SMA can be used to identify support and resistance lines. Furthermore, SMA crossovers can be used to identify emerging price trends or consolidations [74].[4] The SMA is calculated according to (5.1).

$$\mathrm{SMA}(p_t, n) = \frac{p_t + \ldots p_{t-(n-1)}}{n} = \frac{1}{n}\sum_{i=0}^{n-1} p_{t-i}. \tag{5.1}$$

[3] Over a given period, the *open/close/high/low* price of a security is the initial/last/highest/lowest recorded trading price, respectively [72]. Together, these metrics can be used to convey primary information about a security's price movements over a specific period. Unlike other financial markets, however, the DAM only records a single clearing price for each hourly contract. This price is considered to be the *close* price.

[4] When the price crosses the SMA from above/below, it can signal that prices have peaked/troughed. A downward/upward trend may follow.

5.2.2 Exponential Moving Average (EMA)

The EMA [66] is a special type of moving average that exponentially averages historic prices. Through its weighting, the EMA can place greater significance on more recent price trends. This weighting distinguishes the EMA from the SMA and allows the EMA to more rapidly reflect immediate price movements. During periods of high volatility, placing more weight on more recent price moves can be an advantage. The EMA is calculated according to (5.2).

$$\mathrm{EMA}(p_t, s) = \frac{p_t + \alpha \times p_{t-1} + \ldots \alpha^t \times p_0}{1 + \alpha + \ldots \alpha^t}, \tag{5.2}$$

where $\alpha = \frac{s-1}{s+1}$ is the weighting term. α can be tailored to give more or less importance to the recent past.

5.2.3 Moving Average Convergence Divergence (MACD)

The MACD [66] is a trend-following momentum indicator comprising of three time series: the MACD 'Series', 'Signal', and 'Histogram'. These series can be used in tandem to formulate trading rules, for instance, extending the double crossover trading strategy [75]. To avoid introducing too many features at once, the MACD is split into its components, treating each as an individual indicator. These indicators are described below.

- **'Series'**: Calculated from two EMAs, the 'Series' [73] gives insight into price convergence, divergence, and crossover. The 'Series' reflects the difference between a fast (e.g. $s = 12$) and a slow (e.g. $s = 26$) EMA, capturing the second derivative of a price series. Using (5.2), the 'Series' is calculated according to (5.3).

  $$\mathrm{Series}(p_t, s_1, s_2) = \mathrm{EMA}(p_t, s_1) - \mathrm{EMA}(p_t, s_2), \tag{5.3}$$

  where $s_2 > s_1$.

- **'Signal'**: The 'Signal' [73] is an EMA of the 'Series'. It provides a lagging indication of crossovers between fast and slow EMAs. The 'Signal' is calculated according to (5.4), using (5.2) and (5.3).

  $$\mathrm{Signal}(p_t, s_1, s_2, s) = \mathrm{EMA}(\mathrm{Series}(p_t, s_1, s_2), s). \tag{5.4}$$

- **'Histogram'**: The 'Histogram' [73] is the difference between the 'Series' and 'Signal'. Mathematically, it can be interpreted as the fourth derivative of a price series, anticipating changes in the 'Series'. Using (5.3) and (5.4), the 'Histogram' is calculated according to (5.5).

$$\text{Histogram}(p_t, s_1, s_2, s) = \text{Series}(p_t, s_1, s_2) - \text{Signal}(p_t, s_1, s_2, s). \tag{5.5}$$

5.2.4 Moving Standard Deviation (MSD)

The MSD [76], measuring the rolling n day volatility of prices, is considered helpful in predicting the size of future price moves. The indicator anticipates periods of low volatility following periods of high volatility and vice versa. Using (5.1), the MSD is calculated according to (5.6).

$$\text{MSD}(p_t, n) = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(p_{t-i} - \text{SMA}(p_t, n))^2}. \tag{5.6}$$

5.2.5 Bollinger Bands (BBANDs)

BBANDs [66] consist of two bands: the BBAND$^+$, calculated according to (5.7), and the BBAND$^-$, calculated according to (5.8).

$$\text{BBAND}^+(p_t, n) = \text{SMA}(p_t, n) + 2 \times \text{MSD}(p_t, n), \tag{5.7}$$

$$\text{BBAND}^-(p_t, n) = \text{SMA}(p_t, n) - 2 \times \text{MSD}(p_t, n). \tag{5.8}$$

These bands, each two MSDs away from the SMA, indicate when a security is overbought (price above the BBAND$^+$) or oversold (price below the BBAND$^-$). BBANDs can be used to facilitate the prediction of future increases/decreases in volatility, and to identify technical signals such as the W-Bottom [77]. To avoid introducing too many features, two compound TIs derived from BBANDs, the %B and the Bandwidth, are explored.

- **%B**: The %B [78] scales the price series by the BBAND width. When the underlying security price equals the SMA, the %B equals 0.5. When the price is equal to the BBAND$^-$/BBAND$^+$, the %B equals 0/1 respectively. Similarly to BBANDs, the %B can be used to identify when prices are overbought or oversold, to predict future volatility and to generate trading ideas. Using (5.7) and (5.8), the %B is calculated according to (5.9).

$$\%B(p_t, n) = \frac{p_t - \text{BBAND}^-(p_t, n)}{\text{BBAND}^+(p_t, n) - \text{BBAND}^-(p_t, n)}. \tag{5.9}$$

- **Bandwidth**: The Bandwidth [78] measures the BBANDs divergence and is used to anticipate changing volatility and price breakouts. Using (5.1), (5.7), and (5.8), the Bandwidth is calculated according to (5.10).

$$\text{Bandwidth}(p_t, n) = \frac{\text{BBAND}^{+}(p_t, n) - \text{BBAND}^{-}(p_t, n)}{\text{SMA}(p_t, n)}. \tag{5.10}$$

5.2.6 Momentum (MOM)

The MOM [66] is a trend-following leading indicator. Elaborating, the MOM provides insight into price trends, acting as a signal to buy/sell when crossing above/below the zero line. Unlike the SMA, the MOM can peak or trough before the price, providing a forward-looking ('leading') trend prediction. As a forward-looking indicator, when the MOM peaks or troughs and begins to diverge from the main price trend, it can signal bearish or bullish divergence. The MOM is calculated according to (5.11).

$$\text{MOM}(p_t, n) = p_t - p_{t-n}. \tag{5.11}$$

5.2.7 Rate of Change (ROC)

The ROC [66, 73] is an oscillator, comparable to the MOM indicator, that expresses change as a percentage instead of an absolute value. As a standardised measure of change, the ROC can be used to identify overbought or oversold extremes that previously foreshadowed a trend reversal. Note that when above zero, the ROC indicates an overall uptrend, and when below zero, it indicates a downtrend. When prices are moving within a fixed corridor/range, the ROC remains near zero, confirming price consolidation. In these instances, the ROC provides little insight about future price movements. The ROC is calculated according to (5.12).

$$\text{ROC}(p_t, n) = \frac{p_t - p_{t-n}}{p_{t-n}}. \tag{5.12}$$

5.2.8 Coppock Curve (COPP)

The COPP [79] is a smoothed momentum oscillator. Although the COPP was originally developed to capture long-term price trends occurring in American equities, since its inception it has been used to identify both long and short-term trends in numerous markets, e.g. [73]. Using (5.2) and (5.12), the COPP is calculated according to (5.13).

$$\mathrm{COPP}(p_t, s, n_1, n_2) = \mathrm{EMA}\big(\mathrm{ROC}(p_t, n_1) + \mathrm{ROC}(p_t, n_2), s\big). \tag{5.13}$$

5.2.9 True Strength Index (TSI)

Providing trend insights, as well as indications of when a security is overbought or oversold, the TSI [80] is a smoothed momentum oscillator. Technical analysts often look for trend lines in the TSI to identify support and resistance price bands. Using (5.2), the TSI is calculated according to (5.14).

$$\mathrm{TSI}(p_t, s_1, s_2) = \frac{\mathrm{EMA}\big(\mathrm{EMA}(p_t - p_{t-1}, s_1), s_2\big)}{\mathrm{EMA}\big(\mathrm{EMA}(|p_t - p_{t-1}|, s_1), s_2\big)}. \tag{5.14}$$

5.3 Feature Engineering Approach to Day-Ahead Market Price Forecasting

This chapter focuses on demonstrating the explanatory power of TIs by boosting forecasting accuracies of Belgian DAM prices. More information about the Belgian DAM can be found at [81]. Below, data, data processing, TI calculation method, as well as model training and evaluation procedures are described.

5.3.1 Data

A dataset of historic DAM prices, spanning four and a half years, is gathered from [81]. Three and a half years of prices, from 01/01/2014 to 29/06/2017, are used for hyperparameter tuning and model training. A single year of prices, from 30/06/2017 to 30/06/2018, is used for testing. The summary statistics (in €/MWh) of the train/test data are: mean, 41.12/44.96 and standard deviation, 20.16/19.39.

5.3.2 Data Processing

Data processing is conducted to maximise the performances of the models. Firstly, using Min-Max scaling, both features and response variables are scaled.

Min-Max scaling bounds each value $a \in A$ between 0 and 1 according to the formula: $a^* = (a - A^{min})/(A^{max} - A^{min})$, where A^{min} and A^{max} are the minimum and maximum values of a set A, and a^* is the scaled value of a. Secondly, to capture significant serial correlation, an autocorrelation plot (ACF) is examined. ACFs plot lags on the x-axis and autocorrelation coefficients on the y-axes [82]. The plots visually facilitate the selection of an optimal look-back period. Capturing all significant terms in our ACF plot, a 6 day look-back period with an additional averaged 8-week look-back is selected. Formally, to forecast day $d+1$ prices, six prices from days d to $d-5$ along with the average of $d-6$, $d-13$, ..., $d-55$ prices are used.

5.3.3 Forecasting Models

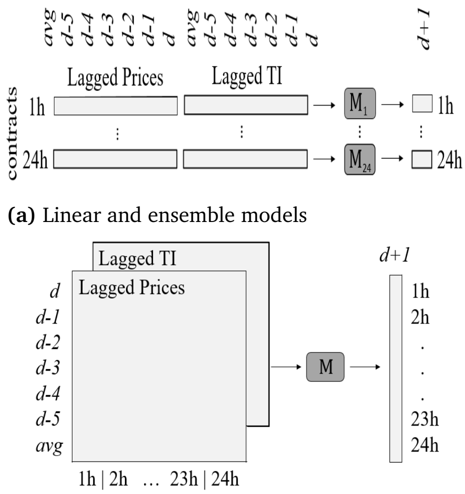
Significant performance gains, using multiple models, must be observed if we are to robustly demonstrate the added explanatory power TIs offer. With this in mind, a list of high-performing models is identified to evaluate TIs. After consulting [28, 83], machine learning (ML) models described in Section 4.4 are chosen. Our selected linear models are LR and HR, and ensemble models are RF, AB and GB. Additionally, deep models are used with: two FCLs (**2NN**); a single CONV (**CNN**); two CONVs (**2CNN**); two CONVs and a single FCL (**2CNN_NN**); seven CONV residual modules (**ResNet**).

5.3.4 TI Calculation

The TI formulas, presented in Section 5.2, require sequential inputs - i.e. continuous linear time. The DAM, however, is not sequential because it releases 24 prices simultaneously upon clearing. Consequently, the dataset is treated as an assortment of prices from 24 separate markets when calculating DAM TIs. Formally, h hour prices are used to calculate h hour TIs. For instance, the 12h SMA($n = 3$) at 09/11/2014 is calculated by taking the average of three 12h prices: the 07/11, 08/11, and 09/11. Finally, in order to identify tailored TIs for every model specified in Section 5.3.3, TI hyperparameters are optimised using grid-search separately for each model.

5.3.5 Model Training and Prediction

To be able to evaluate TI performance, each model introduced in Section 5.3.3 is trained twice: once as a benchmark model accepting only lagged prices, and



(a) Linear and ensemble models

(b) Deep models

Figure 5.1: Diagrams displaying TI model inputs - lagged prices concatenated with a lagged TI - and outputs. Per Section 5.3.2, d to $d-5$ represent the 6 day look-back period, while avg stands for the averaged 8 week look-back. **M** is a black box representation of ML models.

once as a TI model accepting both lagged prices and a lagged TI. Depending on the TI model, one of two approaches is used to concatenate inputs. Length-wise concatenation is used with linear and ensemble models, while channel-wise concatenation is used with deep models. Figure 5.1 visualises these approaches. Forecasting the DAM necessitates predicting 24 hourly prices. Because linear and ensemble models do not support multi-output prediction, but deep models do, to forecast the DAM different training methods are used for linear, ensemble, and deep models. These methods are visualised in Figure 5.1 for TI models. To forecast all 24 prices with linear and ensemble models, the data is split into 24 separate sets and train 24 single-output regressions. For instance, to predict the next day's 12h price, a benchmark model is trained using a set of lagged 12h prices. On the other hand, to forecast DAM prices with deep models, a multi-output regression is trained. Thus, to predict next days 01-24h prices, a benchmark deep model is trained using the entire training set of lagged prices.

Table 5.1: Grid-search selected model hyperparameters. Benchmark model performance is optimised to select these.

	Hyperparameters
LR	-
HR	epsilon:1.35
RF	n_estimators: 100
AB	n_estimators: 100, loss: square, learning rate: 0.1
GB	loss: huber
2NN	neuron_1: 500, neuron_2: 250, learning rate: 0.001
CNN	kernel: (1, 3), filter: 16, CONV layer: 1, learning rate: 0.001, dropout: 0.25
2CNN	kernel: (2, 3), filter: 13, CONV layer: 2, learning rate: 0.001, dropout: 0.25
2CNN_NN	kernel: (1, 3), filter: 32, CONV layer: 2, neuron: 123, learning rate: 0.001
ResNet	kernel: (3, 3), filter: 23, CONV layer: 7, learning rate: 0.001

To select optimal hyperparameters for each model, the training set is divided in two using a 3:1 ratio to get training and validation sets. Grid-search, which maximises benchmark model validation performance, is subsequently conducted using these sets. Table 5.1 presents grid-search selected optimal model hyperparameters. Note that scikit-learn [84] is used to run linear and ensemble models, and Keras [85] to run deep models. As a result, unless otherwise specified in Table 5.1, either scikit's, or Keras' default model hyperparameters are used.

5.3.6 Evaluation

To evaluate the explanatory power of TIs, benchmark and TI model performances are compared. In order to assess performance, various metrics measuring the discrepancies and similarities between a model's targets and predictions are computed. Firstly, the root mean squared error (RMSE) and the mean absolute error (MAE) are calculated for benchmark and TI models. By analysing the percentage change in these discrepancy metrics, %RMSE and %MAE, overall accuracy improvements are determinable. The %RMSE and %MAE are calculated according to (5.15).

$$\%\text{ME} = 100 \times \left(\frac{\text{ME}^{Bench} - \text{ME}^{TI}}{\text{ME}^{Bench}} \right), \tag{5.15}$$

where ME^{Bench} is the mean benchmark model error and ME^{TI} is the mean TI model error. A positive %RMSE and %MAE indicate that the inclusion of TI features, overall, reduces forecast errors.

Secondly, the Pearson correlation coefficient (PCC) is calculated for benchmark and TI models. The PCC is a similarity metric, whose square is equal to the coefficient of determination. By analysing the percentage change in this metric, %PCC, overall improvements in a model's goodness of fit are ascertainable. The %PCC is calculated according to (5.16).

$$\%\text{PCC} = 100 \times \Big(\frac{\text{PCC}^{TI} - \text{PCC}^{Bench}}{\text{PCC}^{Bench}}\Big). \tag{5.16}$$

A positive %PCC indicates that the inclusion of TI features, overall, increases a model's goodness of fit.

Finally, to determine the statistical significance of any performance improvements, one tailed Diebold-Mariano (DM) tests [86] with a mean squared error loss function are conducted. The DM test statistically evaluates differences between two models' forecasts by comparing their residuals, in our case $\{e_t^{Bench}\}_{t=1}^{T}$ and $\{e_t^{TI}\}_{t=1}^{T}$. The test converts a loss-differential, $l_t = g(e_t^{Bench}) - g(e_t^{TI})$ where $g(.)$ is the loss function, into an asymptotically normal DM statistic. Using the DM test, we can statistically evaluate whether TI forecast accuracies are equal to or worse than benchmark forecast accuracies (H_0: $\mathbb{E}(l_t) \leq 0$) or weather they are better (H_1: $\mathbb{E}(l_t) > 0$). We reject H_0 when DM values are above 2.33, indicating statistically significant performance improvement at a 1% significance level.

5.4 Results and Discussion

The repeated selection of a handful of TI hyperparameters, such as %B($\mathbf{n} = \mathbf{58}$), EMA($\mathbf{s} = \mathbf{6}$) and ROC($\mathbf{n} = \mathbf{49}$), by the grid-search optimisation, points to the prevalence of real and identifiable behavioural biases across the Belgian DAM. Our results, presented in Table 5.2 demonstrate that TI features can help in the identification of these biases by adding explanatory power and significantly boosting the regression accuracies of linear, ensemble, and deep models. On average, per the results in Table 5.2, the *best TIs* reduce forecast RMSEs by 3.28%, MAEs by 3.32%, and increase PCCs by 1.96%. These empirical metric improvements are statistically significant at a 1% level, yielding DM values above 2.33, in 9/10 cases.

Summarising TI performances across the ML models: the ROC increased 9 models' accuracies; EMA, 8; SMA, 7; MOM, 6; 'Series', 4; 'Signal', 4; 'Histogram', 4; MSD, 4; %B, 4; COPP, 4; TSI, 3; and Bandwidth, 1. Broadly, the Bandwidth was

Table 5.2: Summary results, showing the three best performing - lowest **RMSE** - TIs for each model. Raw evaluation metrics - **RMSE**, **MAE**, and **PCC** - measure TI model performances. Percentages - **%RMSE**, **%MAE**, and **%PCC** - measure TI model performance improvements. Bold **DM** values represent statistically significant performance improvements at a 1% significance level. The following hyperparameter (**HP**) keys are used: *=(n_1=18, n_2=24, s=18); **=(s_1=2, s_2=26, s=9); ***= (s_1=58, s_2=116, s=9); ****= (n_1=58, n_2=74, s=54).

		LR	**HR**	**RF**	**AB**	**GB**	**2NN**	**CNN**	**2CNN**	**2CNN_NN**	**ResNet**
Best TI	**TI**	%B	EMA	EMA	MOM	MOM	EMA	ROC	ROC	ROC	ROC
	HP	n=58	s=2	s=6	n=58	n=58	s=22	n=49	n=49	n=9	n=27
	RMSE	11.47	11.57	12.21	12.66	11.64	10.96	11.14	11.03	11.10	11.09
	%RMSE	4.49	4.50	1.66	5.42	3.74	4.09	2.73	2.39	2.36	1.38
	MAE	7.87	7.67	8.07	8.58	7.84	7.59	7.65	7.58	7.66	7.56
	%MAE	3.26	5.59	2.10	6.20	2.22	1.56	4.21	2.91	3.41	1.75
	PCC	0.79	0.79	0.76	0.74	0.79	0.81	0.80	0.81	0.81	0.80
	%PCC	2.92	2.75	1.08	3.79	2.80	1.85	1.40	1.19	0.95	0.82
	DM	**10.39**	**14.86**	1.74	**4.18**	**3.68**	**3.23**	**7.15**	**4.51**	**5.15**	**4.28**
Second-Best TI	**TI**	EMA	%B	MOM	ROC	%B	SMA	-	EMA	SMA	COPP
	HP	s=2	n=58	n=58	n=57	n=54	n=22	-	s=18	n=18	*
	RMSE	11.57	11.59	12.22	12.71	11.78	11.08	-	11.26	11.27	11.16
	%RMSE	3.73	4.35	1.57	5.01	2.66	3.00	-	0.37	0.91	0.77
	MAE	7.70	7.78	8.15	8.68	7.81	7.68	-	7.77	7.78	7.60
	%MAE	5.36	4.17	1.10	5.14	2.60	1.56	-	0.44	1.91	1.16
	PCC	0.79	0.79	0.77	0.73	0.78	0.81	-	0.80	0.80	0.80
	%PCC	2.55	2.89	1.57	3.23	1.71	1.38	-	0.06	0.01	0.40
	DM	**13.67**	**12.30**	1.23	**4.36**	1.61	2.08	-	0.16	1.36	**2.40**
Third-Best TI	**TI**	‘Histogram’	SMA	ROC	EMA	EMA	‘Histogram’	-	-	COPP	-
	HP	**	n=18	n=57	s=6	s=6	***	-	-	****	-
	RMSE	11.62	11.66	12.23	12.82	11.79	11.12	-	-	11.28	-
	%RMSE	3.25	3.76	1.50	4.19	2.53	2.66	-	-	0.75	-
	MAE	7.88	7.79	8.18	8.78	7.84	7.68	-	-	7.78	-
	%MAE	3.11	4.02	0.77	4.06	2.20	1.51	-	-	1.89	-
	PCC	0.79	0.79	0.76	0.73	0.78	0.80	-	-	0.80	-
	%PCC	2.22	2.43	0.97	3.01	1.67	1.13	-	-	0.01	-
	DM	**11.92**	**16.02**	0.56	**3.92**	**2.43**	**2.49**	-	-	1.32	-

found to be the worst-performing TI, while the ROC and the EMA were found to be the best.

5.5 Conclusion

This chapter evaluated the explanatory power of TIs by examining whether the inclusion of TI features could boost forecasting accuracies of DAM prices. Overall, it has been demonstrated that TIs can capture the residual impacts from traders’ behavioural biases; resulting in statistically significant reductions in forecast errors with ML models. Evaluated case study has identified four TIs well suited to forecasting Belgian DAM prices: the EMA, the %B, the MOM, and the ROC. More specifically, results advised using %B($n = 58$) and EMA($s = 2$) with linear models, MOM($n = 58$) with ensemble models, EMA($s = 22$) with NNs, and ROC($n = 49$) with CNNs. While it was found that TI performance is model dependent, the ROC and the EMA succeeded in reducing the RMSE and MAE of 90% and 80% of the ML models respectively.

Chapter 6

Forecasting Day-Ahead Market Prices: Data Augmentation

A model's expected generalisation error is inversely proportional to its training set size. This relationship can pose a problem when modelling multivariate time series, because structural breaks, low sampling rates, and high data gathering costs can severely restrict training set sizes, increasing a model's expected generalisation error by spurring regression model overfitting. Artificially expanding the training set size, using data augmentation methods, can, however, counteract the restrictions imposed by small sample sizes: increasing a model's robustness to overfitting and boosting out-of-sample prediction accuracies. While existing time series augmentation methods have predominantly utilised feature space transformations to artificially expand training set sizes and boost prediction accuracies, we propose using autoencoders (AEs), variational autoencoders (VAEs) and Wasserstein generative adversarial networks with a gradient penalty (WGAN-GPs) for time series augmentation. To evaluate our proposed augmentors, Belgian and Dutch day-ahead electricity market prices are predicted using both autoregressive models and artificial neural networks. As our proposed augmentors outperform existing augmentation methods, we strongly believe that both practitioners and researchers aiming to generate time series or reduce time series regression errors will find utility in this chapter.

6.1 Introduction

Sample sizes play a critical part in determining the optimal complexity of hypothesis sets and, by extension, the minimum attainable generalisation error. In multivariate time series analysis, the progression of time can often shift the target distribution. This shift increases the proportion of noise in historic data, making its inclusion in the learning process potentially detrimental. When data is scarce it is crucial to explore techniques capable of augmenting the sample. While numerous studies have demonstrated that augmentation can improve classification and regression accuracies, to date none have explored augmenting multivariate time series (i.e. time series with more than a single time-dependent variable) with exogenous variables despite the fact that the latter is commonly used by practitioners in regression analysis. This chapter aims to fill this gap by identifying methods capable of successfully augmenting multivariate time series for regression analysis.

6.1.1 Sample Sizes and Model Performance

Statistical learning and machine learning aim to identify patterns in data by minimising a model's generalisation error. Because the joint distribution of $\mathbf{x}$, a feature vector, and y, a scalar target output, $T = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$, is unknown, in practice, a direct minimisation of the generalisation error is impossible. Instead, in most statistical learning processes, an empirical error, and a regularisation term, such as the L2 norm, are minimised jointly.

Both bias plus variance decomposition and the Vapnik-Chervonesnkis (VC) theory justify the above statistical learning process. The bias plus variance decomposition specifies a framework for better understanding and analysing the generalisation error. This framework exposes an almost unavoidable trade-off, termed the bias-variance trade-off, between a model's bias in parameter estimation and variance of parameter estimates. The bias-variance trade-off can be used to show that the availability of training data determines the minimum attainable expected generalisation error. The VC theory expresses a generalised upper bound, termed the VC inequality, to the generalisation error. Similarly to the bias-variance trade-off, the VC inequality can be used to directly prove that a model's VC-dimension - i.e. a model's complexity captured by its number of effective parameters - is proportional to the number of training examples needed to attain a certain level of expected modelling performance - i.e. a certain generalisation bound [87]. Together the bias plus variance decomposition and VC inequality motivate studies which seek to reduce the expected generalisation error by expanding training set sizes.

Grounding theory in practice, note that studies, such as [88] for image classification, demonstrate that the relationship between model performance and data quantity is logarithmic. While, to the best of our knowledge, this relationship has yet to be extensively studied for regression models, back of the envelope calculations show that for every new training $(\mathbf{x}, y)$ example the expected generalisation error of, for instance, a linear regression modelling a linear target function decreases by $(eJ+e)/(N^2+N)$, where e is the best approximation error, J is the number of features, and N is the number of samples [87]. The above examples further underscore the importance of dataset sizes in minimising the expected generalisation loss, as well as the potential performance improvements that an expansion of the training set can yield.

6.1.2 Data Augmentation

Linking modelling performance to data augmentation, in instances where it is impractical or even impossible to expand real training dataset sizes, due to costs, data scarcity, or time series structural breaks, it is worthwhile to consider whether augmented data could be used in place of additional real training data to boost modelling performance. While, to the best of our knowledge, a generalised proof establishing a direct relationship between the expected generalisation error and number of augmented data points has yet to be theoretically formulated, numerous studies, such as [89–96], empirically demonstrate that using augmented data can boost both classification and regression accuracies comparable to the addition of real data. Below, the augmentation methods explored in these studies are introduced.

Commencing with augmentation methods that have been utilised to boost classification accuracies, feature space augmentors, exploiting simple transformations such as symmetry, position, or style, have been observed to successfully generate data for both image and time series classification problems [89, 90]. For instance, [90] observed a 5.1% accuracy increase in Parkinson's disease motor state classification, with a convolutional neural network, after augmenting wearable sensory data using random rotations. Despite this success, because of temporal relationships in time series, it must be stressed that applying feature space augmentations to time series can be risky. This is because, for augmentation methods to be useful they have to generate meaningful data originating from the underlying target distribution. Whilst we can observably conclude that images generated using feature space transformations continue to be meaningful, we cannot readily conclude this with time series.

Beyond feature space transformations, researchers have successfully applied augmentors utilising generative autoencoders to classification problems [91, 92]. Autoencoders [97] are artificial neural networks that learn to encode, and subsequently decode model inputs, by performing unsupervised representation learning. Because encoding transforms input features from a feature space X to a latent space Z, autoencoders can generate data by applying perturbations in Z, or by performing Bayesian inference and sampling from a latent distribution. These methods, to the best of our knowledge, have never been used to augment multivariate time series; however, both have a potential theoretical advantage over feature space transformations that make them worth examining.

Generative adversarial networks (GANs) [98] have also been utilised to boost classification accuracies [93, 99]. GANs, comprising of a generator G and a discriminator D, differ from generative autoencoders in their training approach. While the latter is trained by minimising the divergence between model inputs and outputs directly, GANs use D to indirectly train G. The process is comparable to a game between a forger and a detective, and allows G to learn to generate data from the distribution of the real data. Although, to the best of our knowledge GANs have never been used to augment multivariate time series, studies such as [93, 94] demonstrate their ability to generate meaningful data and boost forecasting performances. Tran et al. [94] observed deep model classification accuracy improvements of 6%, across MNIST, CIFAR-10 and CIFAR-100 datasets, surpassing feature space transformation performances.

Finally describing augmentation methods for time series regression, to the best of our knowledge two studies, [95, 96], have thus far attempted to boost regression accuracies using augmentation methods. In particular, [95] proposed generating univariate time series by replacing the error of deseasonalised and detrended time series with bootstrapped errors, while [96] proposed generating the same series by sampling Markov Chain Monte Carlo parameters and forecasting time series paths using those parameters. Across the M3-competition dataset, [95] demonstrated that the bootstrapping augmentation method, combined with bagging, significantly improves the prediction accuracies of exponential smoothing models. Across the same dataset, [96] observed a 5.7% reduction in symmetrical mean absolute percentage errors of long short-term memory recurrent neural networks. Whether these augmentation methods could be adapted to readily and successfully generate multivariate time series with exogenous inputs is uncertain.

Elaborating, for example, to generate multivariate time series using [95], one would have to augment multiple univariate time series; replacing the errors of multiple deseasonalised and detrended time series with correlated bootstrapped errors. To achieve this without introducing an excessive amount of noise, cross correlations between the time series would probably have to be modelled. Moreover, the time series would have to be of the same length, with the same sampling periodicity. As the above, in and of itself, constitutes a new data augmentation method, as it significantly extends the use case presented in [95], we consider any attempts to implement it beyond the scope of our examination.

6.1.3 Motivation and Contributions

The literature review above exposes how researchers have thus far concentrated more on developing augmentation methods for classification than regression. Particularly, augmentation methods for multivariate time series with exogenous variables have to date not been researched. Motivated by a desire to fill this void, this chapter concentrates on developing universally applicable augmentation methods for both univariate and multivariate time series regression analysis. Because of their prevailing use in classification studies as well as their ease of implementation, three feature space augmentors - jittering, scaling, and magnitude-warping - are evaluated. Similarly, because of their significant achievements with classifiers, their capacity to model the underlying distribution of data, and their ability to generate data without any domain knowledge, tailored model-based augmentors - autoencoders (AEs), variational autoencoders (VAEs), and Wasserstein GANs with a gradient penalty (WGAN-GP) - are developed and evaluated.

To fully explore the effectiveness of the above-mentioned augmentation methods, their performance impact on autoregressive models with exogenous inputs (ARX) and two artificial neural networks (ANN) are examined. ARXs are chosen both to demonstrate the methods impacts on linear model performances, and because they have been shown to achieve reasonable DAM benchmark forecast accuracies. ANNs are chosen because they have been shown to outperform other state-of-the-art statistical and machine learning models in DAM forecasting [28]. Summarising the principle contributions of this chapter, to the best of our knowledge, we are the first to:

- explore the augmentation of multivariate time series with exogenous variables,
- utilise feature space transformations, AEs, VAEs and WGAN-GPs for regression augmentation,
- apply augmentation methods to the forecasting of DAM prices.

Outlining the structure of this chapter, in Sections 6.2-6.5 the theories and models underpinning our augmentation methods are described. In Section 6.6 our methodology and case study are outlined. In Section 6.7 the results are analysed. Finally, in Section 6.8 the chapter is concluded.

6.2 Kullback-Leibler Divergence

The Kullback-Leibler divergence (D_{KL}) is an asymmetric statistical distance measure, measuring the divergence between two distributions. The forward D_{KL} is calculated as: $D_{KL}\big(p_X(\mathbf{x})||p_{\tilde{X}_\omega}(\tilde{\mathbf{x}})\big) = \sum_j p_X(x_j)\ln\frac{p_X(x_j)}{p_{\tilde{X}_\omega}(\tilde{x}_j)}$, where p_X and $p_{\tilde{X}_\omega}$ are real and generated data distributions respectively. The reverse D_{KL} is calculated as $D_{KL}\big(p_{\tilde{X}_\omega}(\tilde{\mathbf{x}})||p_X(\mathbf{x})\big)$. As later demonstrated, the weights, $\boldsymbol{\omega}$, of autoencoders and GANs can be optimised by minimising the forward D_{KL} and reverse D_{KL} respectively. The impacts of minimising these metrics on data generation are highlighted below.

By reformulating (6.1), it can be shown that minimising the forward D_{KL} is equivalent to performing maximum likelihood estimation [59].

$$\begin{aligned}\arg\min_{\boldsymbol{\omega}} D_{KL}(p_X(\mathbf{x})\,||\,p_{\tilde{X}_\omega}(\tilde{\mathbf{x}})) &= \arg\min_{\boldsymbol{\omega}} \mathbb{E}_{\mathbf{x}\sim p_X}[-\log p_{\tilde{X}_\omega}(\tilde{\mathbf{x}}) + \log p_X(\mathbf{x})] \quad (6.1)\\ &= \arg\min_{\boldsymbol{\omega}} \mathbb{E}_{\mathbf{x}\sim p_X}[-\log p_{\tilde{X}_\omega}(\tilde{\mathbf{x}})] - \mathcal{H}(p_X(\mathbf{x}))\\ &= \arg\max_{\boldsymbol{\omega}} \mathbb{E}_{\mathbf{x}\sim p_X}[\log p_{\tilde{X}_\omega}(\tilde{\mathbf{x}})],\end{aligned}$$

where $\mathcal{H}(p_X(\mathbf{x}))$ is the entropy of real data, which is known and constant. The consequence of minimising the forward D_{KL} is a 'mean seeking' approximation, which spurs a model to cover the support of p_X, assigning a high probability mass where p_X is high, while centring $p_{\tilde{X}_\omega}$ around the mean of p_X.

In contrast, minimising the reverse D_{KL} encourages 'mode seeking' approximations of p_X by assigning a high probability mass to the mode of the real distribution [59]. Equation (6.2) formally expresses this.

$$\arg\min_{\boldsymbol{\omega}} D_{KL}(p_{\tilde{X}_{\boldsymbol{\omega}}}(\tilde{\mathbf{x}}) \,||\, p_X(\mathbf{x})) = \arg\min_{\boldsymbol{\omega}} \mathbb{E}_{\tilde{\mathbf{x}} \sim p_{\tilde{X}_{\boldsymbol{\omega}}}}[-\log p_X(\mathbf{x})] - \mathcal{H}(p_{\tilde{X}_{\boldsymbol{\omega}}}(\tilde{\mathbf{x}})), \quad (6.2)$$

where $\mathcal{H}(p_{\tilde{X}_{\boldsymbol{\omega}}}(\tilde{\mathbf{x}}))$ is the entropy of generated data. Equation (6.2) highlights that when minimising the reverse D_{KL}, a generative model learns to assign low probabilities to $p_{\tilde{X}_{\boldsymbol{\omega}}}$ where p_X is low.

6.3 Autoencoders

Autoencoders are lossy networks that learn to encode and subsequently decode model inputs. Generally, autoencoders are implemented in a $J/U/J$ or bottleneck architecture, where J is the dimension of input and output layers, U is the dimension of hidden layers/latent space, and $J > U$. Under such a structure the encoder extracts the most salient features from the input vector, $\mathbf{x}$; transforming inputs from the feature space $X \in \mathbb{R}^J$ to an unstructured and generally more compact latent space $Z \in \mathbb{R}^U$. To find the optimal encoder and decoder parameters, $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ respectively, autoencoders are trained by minimising a reconstruction loss according to (6.3) [100].

$$\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{E}_{\mathbf{x} \sim p_X(\mathbf{x})}\, \ell(\mathbf{x}, D_{\boldsymbol{\theta}}(E_{\boldsymbol{\phi}}(\mathbf{x}))), \quad (6.3)$$

where $E_{\boldsymbol{\phi}}(.)$, $D_{\boldsymbol{\theta}}(.)$, and $\ell(\mathbf{x},.)$ are encoding, decoding, and distance functions respectively, and $\mathbf{x}$ is a vector sampled from the training set $X = \{\mathbf{x}_1, ..., \mathbf{x}_\mathrm{N}\}$. As [101] argues and as further explained in Section 6.6.7, it is the dimensionality reduction, which occurs during encoding, that makes latent space transformations more likely to generate data from the underlying distribution than feature space transformations.

6.4 Variational Autoencoders

VAEs [102] are a special class of autoencoder, modelling observed and latent variable probability distributions while learning structured latent space representations $\mathbf{z}$ of real observations $\mathbf{x}$. To learn these representations, VAEs use variational inference to approximate the true posterior, $p(\mathbf{z}|\mathbf{x})$, with a variational posterior $q_{\lambda_v}(\mathbf{z}|\mathbf{x})$, where λ_v is a collection of variational parameters. From a neural net perspective, VAEs perform variational expectation maximisation to optimise the parameters ϕ of an inference network $q_{\phi}(\mathbf{z}|\mathbf{x})$ that outputs λ_v. VAE encoders are called inference networks because they parametrise the inference of a true posterior. VAE decoders, performing the parametrised probabilistic decoding $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, are called generation networks.

Deriving the VAE objective function [102], note that directly approximating the true distribution of $\mathbf{x}$ as: $p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})dz = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})dz$ is intractable because it requires evaluating all configurations of $\mathbf{z}$. However, by reformulating the above using Bayes' rule, a tractable objective function representing the lower bound of $\mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})} \log p_{\boldsymbol{\theta}}(\mathbf{x})$ can be derived. Specifically, to optimise the parameters of the inference and generation networks, ϕ and $\boldsymbol{\theta}$ respectively, the value function $\mathcal{V}$ is maximised according to (6.4).

$$\arg\max_{\phi,\boldsymbol{\theta}} \mathcal{V}(\boldsymbol{\theta},\phi;\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_{\phi}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}(\mathbf{x}|\mathbf{z})]-D_{KL}[q_{\phi}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]}. \tag{6.4}$$

Since the value function $\mathbf{V}(\boldsymbol{\theta},\phi;\mathbf{x}) = \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}) - D_{KL}[q_{\phi}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})]\right] \leq \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x})]$ [103], maximising $\mathcal{V}(\boldsymbol{\theta},\phi;\mathbf{x})$ is equivalent to minimising the forward $D_{KL}[q_{\phi}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})]$. This is because, as the forward D_{KL} approaches 0, $\mathcal{V}$ approaches $\mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})} \log p_{\boldsymbol{\theta}}(\mathbf{x})$. Linking $\mathcal{V}$ to autoencoders, by taking the negative of $\mathcal{V}$ similarities and differences between VAE and autoencoder objective functions can be identified. Formally, the negative of $\mathcal{V}$ is equal to (6.5).

$$\arg\min_{\boldsymbol{\theta},\phi} \mathbf{L}(\boldsymbol{\theta},\phi;\mathbf{x}) = -\left[\mathbb{E}_{\mathbf{z}\sim q_{\phi}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}(\mathbf{x}|\mathbf{z})]-D_{KL}[q_{\phi}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]}\right]. \tag{6.5}$$

In (6.5), the first term on the right measures the expected negative log-likelihood of $\mathbf{x}$. This term is equivalent to an autoencoder's reconstruction loss. The second term measures the information loss from variational approximation: it is a regulariser term controlling the structuredness of $\mathbf{z}$ and distinguishing VAEs from autoencoders.

6.5 Wasserstein Generative Adversarial Networks

GANs are adversarial networks that learn to generate samples from the underlying probability distribution of data without explicitly modelling said distribution. They do this by pitting two neural networks G_{ϕ}, a generator, and $D_{\boldsymbol{\theta}}$, a discriminator, against each other in a minmax game. To train $D_{\boldsymbol{\theta}}$, [104] proposed maximising the negative cross-entropy of a binary classifier, outputting variable y, and separating real ($\mathbf{x} \sim p_X(\mathbf{x}|y=1)$) and generated ($\tilde{\mathbf{x}} \sim p_{G_\phi}(\tilde{\mathbf{x}}|y=0)$) data. Formally, $\boldsymbol{\theta}$ and ϕ are optimised according to (6.6) and (6.7).

$$\arg\max_{\boldsymbol{\theta}} \mathcal{V}_D(\phi,\boldsymbol{\theta};\mathbf{x},\mathbf{z}) = \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}[\log D_{\boldsymbol{\theta}}(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_Z(\mathbf{z})}[\log(1 - D_{\boldsymbol{\theta}}(G_\phi(\mathbf{z})))], \tag{6.6}$$

$$\arg\max_{\phi} \mathcal{V}_G(\phi,\boldsymbol{\theta};\mathbf{z}) = \mathbb{E}_{\mathbf{z}\sim p_Z(\mathbf{z})}[\log D_{\boldsymbol{\theta}}(G_\phi(\mathbf{z}))]. \tag{6.7}$$

Equating GANs to VAEs, by treating the binary targets y of $D_{\boldsymbol{\theta}}$ as observed variables, and the inputs $\acute{X} = \{\mathbf{x},\tilde{\mathbf{x}}\}$ of $D_{\boldsymbol{\theta}}$ as latent variables, [103] established a connection between a GAN's objective functions and variational expectation maximisation. Similarly to a VAE's inference network, [103] highlighted that G_ϕ can be thought of as performing posterior inference; with the variational posterior $p_\phi(\acute{X}|y)$ approximating the posterior $q^r(\acute{X}|y) \propto q_{\boldsymbol{\theta}_0}(1-y|\acute{X})\left[\mathbb{E}_{y\sim p_Y(y)}\, p_{\phi_0}(\acute{X}|y)\right]$, where $\boldsymbol{\theta}_0$ and ϕ_0 are D and G weights from the previous training iteration respectively, and, $q_{\boldsymbol{\theta}_0}(y|\acute{X})$ is equal to $D_{\boldsymbol{\theta}_0}(\acute{X})$, as shown in (6.8).

$$\begin{aligned}\nabla_\phi \mathbb{E}_{\acute{X}\sim p_\phi(\acute{X}|y)p(y)} \log q_{\boldsymbol{\theta}_0}(1-y|\acute{X}) = \nabla_\phi JSD\Big(p_{G_\phi}(\tilde{\mathbf{x}}|y=0)||p_X(\mathbf{x}|y=1)\Big)\\ - \nabla_\phi \mathbb{E}_{y\sim p(y)}\Big[D_{KL}(p_\phi(\acute{X}|y)||q^r(\acute{X}|y)\Big].\end{aligned} \tag{6.8}$$

In (6.8), the JSD term, whose impact becomes negligible once the reverse D_{KL} is sufficiently minimised, is upper bounded by the reverse D_{KL} term [103]. The optimisation of G_ϕ according to (6.7) is thus equivalent to the minimisation of the reverse D_{KL}.

There are two critical drawbacks with optimising ϕ according to (6.7): unstable gradient updates and mode collapse [105]. For a fixed G_ϕ, as $D_{\boldsymbol{\theta}}$ improves and approaches optimality ($\boldsymbol{\theta} \to \boldsymbol{\theta}^*$) the gradient norm of the objective function - $\|\nabla_\phi \log D_{\boldsymbol{\theta}}\big(G_\phi(\mathbf{z})\big)\|$ - rapidly explodes. An instability of gradient updates spurs network saturation/instability and increases the complexity of hyperparameter tuning. Further, as minimising the reverse D_{KL} encourages 'mode seeking' approximations of p_X, optimising ϕ according to (6.7) can provoke mode collapse, which occurs when G_ϕ, with varying input vectors, begins generating data centred at a single mode of a complex multimodel dataset.

Wasserstein GANs (WGANs) [106] are a variant of traditional GANs that offer an alternative training approach to [104] centred around the optimisation of a more stable objective function: the Wasserstein distance. The Wasserstein distance, a symmetric measure of distribution similarity, is mathematically defined in its primal form as the infimum (greatest lower bound) energy cost of transforming $p_{\tilde{X}}$ into p_X. Unlike the JSD and reverse D_{KL}, the Wasserstein distance is continuous everywhere and yields smooth and meaningful gradients even when the manifolds of p_X and $p_{\tilde{X}}$ have disjoint supports. Moreover, it limits mode collapse, by dissuading 'mode seeking' approximations of p_X. Because computing the infimum is intractable, in practice a dual form of the Wasserstein distance, expressed in (6.9), is computed.

$$W(p_X, p_{\tilde{X}}) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}[f(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}}\sim p_{\tilde{X}}(\tilde{\mathbf{x}})}[f(\tilde{\mathbf{x}})], \tag{6.9}$$

where sup is the supremum (least upper bound), $f : X \to \mathbb{R}$ is a 1-Lipschitz function with $|f(x_1) - f(x_2)| \leq |x_1 - x_2|$. Arjovsky et al. [106] derived the above objective function using the Kantorovich-Rubinstein duality; turning a minimisation problem with an infimum into a maximisation problem with a supremum. Formalising the WGAN training procedure, using f in place of D, ϕ and $\boldsymbol{\theta}$ are optimised according to (6.10).

$$\arg\min_{\phi}\max_{\boldsymbol{\theta}} W(\phi, \boldsymbol{\theta}; \mathbf{x}, \mathbf{z}) = \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}[f_{\boldsymbol{\theta}}(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p_Z(\mathbf{z})}[f_{\boldsymbol{\theta}}(G_{\phi}(\mathbf{z})))]. \tag{6.10}$$

Note, $f_{\boldsymbol{\theta}}$ must satisfy the Lipschitz constraint, because only Lipschitz continuous functions produce feasible/optimal solutions for both the dual/primal forms of the Wasserstein loss respectively. To satisfy the Lipschitz constraint, [106] proposed using gradient clipping when optimizing $\boldsymbol{\theta}$. Alternatively, a gradient penalty, underlined in (6.11), may be added to the objective function [107].

$$\arg\min_{\boldsymbol{\theta}} W(\boldsymbol{\theta}; \mathbf{x}, \tilde{\mathbf{x}}) = \mathbb{E}_{\tilde{\mathbf{x}}\sim p_{\tilde{X}}(\tilde{\mathbf{x}})}[f_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})] - \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}[(f_{\boldsymbol{\theta}}(\mathbf{x}))] + \underbrace{\lambda\,\mathbb{E}_{\hat{\mathbf{x}}\sim p_{\hat{X}}(\hat{\mathbf{x}})}[(\| \nabla_{\hat{\mathbf{x}}} f_{\boldsymbol{\theta}}(\hat{\mathbf{x}}) \|_2 - 1)^2]}_{\text{gradient penalty}}, \tag{6.11}$$

where λ is the gradient penalty coefficient and $\hat{\mathbf{x}}$ is an interpolated randomly sampled vector: $\alpha\tilde{\mathbf{x}} + (1-\alpha)\mathbf{x}$ with $0 \leq \alpha \leq 1$. The gradient penalty ensures that the gradient norm of the objective function does not exceed 1, without requiring extensive hyperparameter tuning.

Table 6.1: Summary statistics of Belgian and Dutch DAM prices (in €/MWh).

Belgian DAM Prices						*Dutch DAM Prices*					
Training		*Validation*		*Test*		*Training*		*Validation*		*Test*	
Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
36.78	23.69	44.59	21.62	55.27	23.54	32.29	11.35	39.31	12.76	52.53	15.18

6.6 Data Augmentation to Day-Ahead Market Price Forecasting

6.6.1 Data

As a case study, to evaluate the robustness of our augmentation methods, Belgian and Dutch DAM prices are predicted. To forecast these, lagged DAM prices [10, 28], day-ahead grid load forecasts [29], day-ahead available generation forecasts [28] and meteorological features, namely actual and day-ahead forecasts of temperature, wind speed, solar irradiance, and precipitation, are used. From [108, 109] dataset is gathered. Data from 01/01/2016 to 31/12/2016 is used for model training and data augmentation (training), data from 01/01/2017 to 31/12/2017 for hyperparameter tuning (validation), and data from 01/01/2018 to 31/12/2018 for augmentation evaluation (test). Summary statistics of DAM prices are presented in Table 6.1. Training, validation, and test means (**Mean**) and standard deviations (**SD**) are presented.

Analysing Table 6.1, it is observed that the dataset of DAM prices displays varying statistical characteristics across countries (Belgium/The Netherlands) and time (training/validation/test). On average, the Mean and SD of DAM prices are higher in Belgium than The Netherlands. Finally, the Mean and SD of DAM prices are observed to generally increase across the training, validation, and test sets, i.e. across time.

6.6.2 Data Processing

To facilitate data augmentation using generator networks, Min-Max normalisation, which binds data in the range $[0, 1]$, is used. Such data can be readily generated by applying a sigmoid activation function across final layer network outputs as in [110].

6.6.3 Feature Selection

Many researchers have modelled the DAM as a multiple output regression problem. The DAM, however, is chosen to model as a single output regression problem, forecasting the prices of every DAM hourly contract independently. This allows us to: (1) tailor our modelling approach to the varying statistical characteristics of every contract, (2) capture any price drivers which may be unique to a specific contract, and (3) avoid using too many features in the forecasting of any single contract. Feature selection is used to shrink an initial feature pool of 2160 features to a reduced feature space of at most 35 features. Our initial feature pool, for any DAM contract, consists of the features mentioned in Section 6.6.1 spanning a seven-day-lagged period. Specifically, the total of 2160 features comprises of 1440[1] meteorological features, 384[2] generation and load features, and 336[3] price features. For example, the Dutch feature pool contains seven-day-lagged 00h-23h: Dutch prices, Belgian prices, Dutch generation and load forecasts, and Dutch meteorological features. Note that Belgian and Dutch prices are used together for both the Dutch and Belgian feature pools to increase the explanatory variability coming from neighbouring countries.

To understand why our modelling input size is constrained to at most 35 features refer back to our discussion of the generalisation bound in Section 6.1.1. The VC-bound[4] for a regression can be expressed according to (6.12) [111]:

$$\mathbb{P}\left(E_{\text{out}}(h) \leq \frac{E_{\text{in}}(h)}{\left(1 - c\sqrt{\gamma}\right)_{+}}\right) = 1 - \delta, \tag{6.12}$$

where $\gamma = a/N\,[VC(H) + VC(H)\ln(bN/VC(H)) - \ln(\delta/4)]$, a, b, c are constants, N is the sample size, E_{out} and E_{in} are the generalisation and empirical errors, $1-\delta$ is the probability that the bound holds, and $VC(H)$ is the VC-dimension of the family of regression models $\mathcal{F} = \{f(\mathbf{x}, h)\} : \mathbb{R}^J \to \mathbb{R}$ indexed by $h \in H$. The VC-bound exposes how changes in $VC(H)$ and N impact the expected generalisation error. Because the generalisation bound follows the same monotonicity as

[1] 360 temperature features, 360 wind speed features, 360 solar irradiance features and 360 precipitation features. Each of the above consists of 7 * 24 (seven-day) lagged actuals + 7 * 24 (seven-day) lagged forecasts + 24 next day forecasts.

[2] 192 generation features and 192 load features; each consisting of 7 * 24 (seven-day) lagged forecasts + 24 next day forecasts.

[3] 168 Belgian and 168 Dutch DAM prices; each consisting of 7 * 24 (seven-day) lagged actuals.

[4] VC-theory assumes that sample points are independently and identically distributed (i.i.d.). While DAM prices are not uniformly i.i.d., in determining the maximum number of features this assumption is made to obtain reasonable bounds.

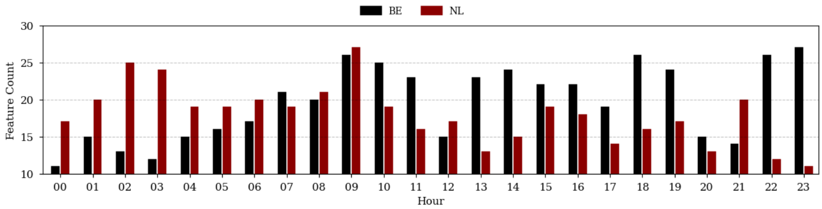


Figure 6.1: Visualisation presenting the number of selected features for every Belgian (BE) and Dutch (NL) DAM hour denominated contract. On average, roughly 20 features are selected for Belgian contracts and 18 for Dutch contracts.

the VC-bound, (6.12) can be used to determine the N needed to train a family of models $\mathcal{F}$, or the $\mathcal{F}$ that a constrained N can reasonably train. Numerous studies, such as [87], indicate that by applying a rule of 10, mathematically expressed as $10 \times VC(H) \leq N$, a reasonable generalisation error - i.e. a level where the VC-bound is meaningful with a probability close to 1 - can be attained. The rule of 10 can either be applied to determine an appropriate N for a given $VC(H)$ or the reverse. Given our N of 365, the rule of 10 suggests that an $\mathcal{F}$ with a $VC(H) \leq 36$ is best suited to our case study.

Linking the $VC(H)$ to our final modelling feature size, it can be shown that the $VC(H)$ of a J-dimensional linear classifier and regression is equal to $J+1$ - i.e. its degrees of freedom. The $VC(H)$ of a real-valued function $\mathcal{F} = \{f(\mathbf{x}, h)\}$ is equal to the $VC(H)$ of its respective binary classifier $\mathbb{1}(f(\mathbf{x}, h) - [f(\mathbf{x}, h)] > 0)$ [51]. Because the expressiveness and by extension $VC^J_{NN}(H)$ of J-dimensional real-valued neural networks is at least as large as the $VC^J_{LR}(H)$ of a J-dimensional linear regression, a limit can be determined for the maximum number of feature inputs for our case study. Formally, $\mathcal{F}^J_{LR} \subseteq \mathcal{F}^J_{NN}$ implies that $[VC^J_{LR}(H) = J+1] \leq VC^J_{NN}(H)$, and therefore $J \leq VC^J_{LR}(H)) \leq VC^J_{NN}(H)$. From the above it can be discerned that strictly no more than 35 feature inputs, J, should be selected by our feature selection method.

Similarly to [112], random forests (RF) are used to identify and select features with the greatest explanatory power. RF, an ensemble machine learning model, fits numerous decision trees to random samples of a training dataset. Because each tree is trained to recursively split data by maximising an information gain, estimates of a feature's importance can be computed. This property, as well as

RF's computational speed, and robustness to outliers and noise [113], make RF an effective feature selection method.

The RF feature selection method is trained on the training set and tuned by minimising the R^2 across the validation set. A feature importance cut-off that prevents the RF method from selecting more than 35 features is set. Figure 6.1 displays the total number of features selected for every Belgian and Dutch DAM contract. One-day-lagged and seven-day-lagged dependent variables are frequently selected, capturing the well-known seosonalities in DAM prices. For example, the following 17 features are selected for the Dutch 14h contract: BE^{11}_{-7}, BE^{14}_{-7}, NL^{14}_{-7}, generation_fc^{14}_{-7}, BE^{15}_{-7}, NL^{15}_{-7}, NL^{16}_{-7}, NL^{17}_{-2}, NL^{14}_{-1}, NL^{16}_{-1}, BE^{17}_{-1}, NL^{17}_{-1}, NL^{18}_{-1}, load_fc^{14}_{-1}, and wind_fc^{14}_{-1}. The NL^{14}_{-7} and generation_fc^{14}_{-7} represent the seven-day-lagged Dutch 14h DAM price and seven-day-lagged Dutch 14h generation forecast respectively. While both the one-day-lagged and seven-day-lagged dependent variables are selected for the Dutch 14h contract, as our RF feature selection method is not constrained, no lagged dependent variables are selected in 6 out of 48 cases: the Belgian: {00h, 01h, 14h}, and the Dutch: {00h, 10h, 13h}.

6.6.4 Forecasting Models

To establish reasonable benchmark forecast accuracies DAM prices are modelled using ARXs. Further, following [28], DAM prices are modelled using a two-layer neural network (**2NN**) consisting of two intermediate FCLs. Finally, to evaluate the performance impacts of augmentation on convolutional neural networks, DAM prices are modelled using a joint three-layer network (**2CNN_NN**) consisting of two intermediate CONVs, two optional max-pooling or average-pooling layers, and a single intermediate FCL. Layers are stacked to identify non-linear relationships, and prevent an explosion in the number of network parameters. Early stoppage, L2 regularisation, dropout, ReLU activation functions, Adam, learning rate scheduling, and learning rate decay are used to combat node saturation, speed-up model training, and reduce the likelihood of overfitting.

Bayesian optimisation is used to identify optimal architectures and hyperparameters of every model. Models are trained on the training set and evaluated on the validation set. The hyperparameters are selected based on the best validation score, i.e. the lowest root mean squared error. Note that the augmented data is not used for optimisation. Samples of selected hyperparameters are presented in Tables 6.2 - 6.3. For ARX, bayesian optimisation is

Table 6.2: Optimised ARX model hyperparameters for the Belgian 15h and 17h, and Dutch 06h and 22h.

	ARX (*BE 15h*)	**ARX** (*BE 17h*)	**ARX** (*NL 06h*)	**ARX** (*NL 22h*)
Regularisation Parameter	1.78×10^{-3}	1.70×10^{-3}	1.29×10^{-4}	2.11×10^{-4}
Maximum Iterations	1450	1175	925	1425
Tolerance	2.40×10^{-4}	6.42×10^{-4}	1.49×10^{-5}	3.66×10^{-4}

Table 6.3: ANN hyperparameters for the Belgian 15h and 17h, and Dutch 06h and 22h.

	2NN (*BE 17h*)	**2CNN_NN** (*BE 15h*)	**2NN** (*NL 22h*)	**2CNN_NN** (*NL 06h*)
Hidden Layers	[500, 500]	[(1, 5, 48), AvgPool, (1, 3, 32), 12]	[97, 43]	[(1, 4, 8), (1, 4, 32), 64]
Dropout	0.25	0.00	0.00	0.00
Learning Rate	5.62×10^{-3}	9.07×10^{-4}	1.36×10^{-3}	1.27×10^{-4}

used to select the L1 regularisation parameter $\sim U[0,1]$, the maximum number of training iterations $\sim U[800, 825, 850, \ldots, 1600]$, and the optimisation tolerance $\sim U\big[\log(10^{-5}), \log(10^{-3})\big]$. For ANN, bayesian optimisation is used to identify the ANN hidden layers, the dropout $\sim U[0, 0.25, 0.5]$, and the learning rate $\sim U\big[\log(10^{-4}), \log(2 \times 10^{-2})\big]$.

Note, in Table 6.3 CONVs are represented using tuples specifying kernel height and width, and the number of filter outputs. FCLs are represented using integers specifying neuron count. An average pooling layer, using a one dimensional kernel of size 2 with a stride of 2, is represented using the acronym AvgPool. Explaining the above notation with an example: the NL 06h 2CNN_NN consists of two CONVs, with one dimensional kernels and channel outputs of size 4 and 4, and 8 and 32 respectively, connected to a single FCL with 64 neurons. This network is specified as [(1, 4, 8), (1, 4, 32), 64] in Table 6.3.

Early stoppage and learning rate scheduling and decay are employed while training ANNs. ANN training continues until either an early stoppage criterion, requiring a minimum training loss improvement of 1e-6 over 22 epochs, is satisfied, or the maximum number of training epochs, 200, is reached. The learning rate is reduced by a factor of 10 after every 50 epochs. Additionally, it is reduced by a factor of $4/3$ after 7 epochs of the training loss plateauing.

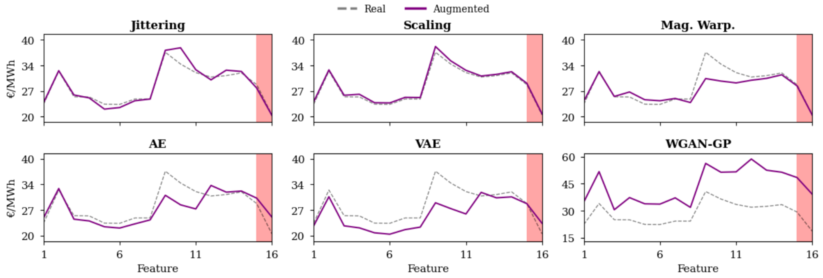


Figure 6.2: Generated time series examples for the Belgian 01h contract. Note that $\mathbf{x}$ consists of lagged prices (in €/MWh) for this contract. The training (Real) time series is used as an input in the generation of the augmented series. The violet band displays how $\tilde{\mathbf{t}}$ is sliced to give $\tilde{\mathbf{x}}$, in the feature range $[1,15]$, and $\tilde{y}$, in the feature range $(15,16]$.

6.6.5 Basics of Data Augmentation

When augmenting multivariate time series, both X, real explanatory variables, and Y, real response variables, can be simultaneously used to generate $\tilde{X}$, augmented explanatory variables, and $\tilde{Y}$, augmented response variables. X and Y represent measurements indexed through time, forming a set of time series $\{\mathbf{t} \in T \mid \mathbf{t} = \mathbf{x} \cup y,\ \forall \mathbf{x} \in X,\ y \in Y\}$. An ability to generate a set of augmented time series $\{\tilde{\mathbf{t}} \in \tilde{T} \mid \tilde{\mathbf{t}} = \tilde{\mathbf{x}} \cup \tilde{y},\ \forall \tilde{\mathbf{x}} \in \tilde{X},\ \tilde{y} \in \tilde{Y}\}$ simplifies augmentation by alleviating the need for any pseudo-labeling of $\tilde{X}$. Examples of generated time series are presented in Figure 6.2. Below Sections 6.6.6-6.6.9 describe how, given a set of normalised training inputs T, normalised augmented outputs $\tilde{T}$ can be generated using a multitude of augmentation methods.

6.6.6 Feature Space Augmentation

From [90], jittering, scaling and magnitude-warping are chosen. Briefly describing how these methods generate data, jittering adds varying amounts of noise ($\epsilon \sim N(0, \sigma^2 I)$) to $\mathbf{t}$. Scaling generates data by multiplying $\mathbf{t}$ by a random scalar ($s \sim N(1, \sigma^2)$). Magnitude-warping, similarly to scaling, multiplies $\mathbf{t}$ by a smoothly-varying random curve with r knots and a standard deviation σ. To select the optimal hyperparameters σ and r, Bayesian optimisation is employed. Table 6.4 in Section 6.6.10 presents examples of selected hyperparameters.

6.6.7 Autoencoder Augmentation

AEs, introduced in Section 6.3, are utilised to augment multivariate time series. Similarly to [110], as a distance function, $l(\mathbf{x},.)$, to measure the reconstruction loss in (6.3) the binary cross-entropy (BCE) is chosen. It is found that using the BCE, an asymmetric loss function, outperformed using a symmetric loss, such as the mean squared error.[5] The BCE a special instance of the cross-entropy $\mathcal{H}(p_X(\mathbf{x}), p_{\tilde{X}}(\tilde{\mathbf{x}})) = -\sum_j p_X(x_j) \log p_{\tilde{X}}(\tilde{x}_j)$, is calculated according to (6.13).

$$\begin{aligned}\mathrm{BCE}(\mathbf{x}, D_{\boldsymbol{\theta}}(E_{\boldsymbol{\phi}}(\mathbf{x}))) = -\sum_{j=1}^{J} \big[p_X(x_j) \log p_{\tilde{X}}(D_{\boldsymbol{\theta}}(E_{\boldsymbol{\phi}}(x))_j) \\ + \big(1 - p_X(x_j)\big) \log\big(1 - p_{\tilde{X}}(D_{\boldsymbol{\theta}}(E_{\boldsymbol{\phi}}(\mathbf{x}))_j)\big), \qquad (6.13)\end{aligned}$$

where J is the dimensionality of X. Optimising the BCE pointwise is equivalent to optimising the forward D_{KL}: $D_{KL}(p_X(\mathbf{x}) \,||\, p_{\tilde{X}}(\tilde{\mathbf{x}})) = \mathcal{H}(p_X(\mathbf{x}), p_{\tilde{X}}(\tilde{\mathbf{x}})) - \mathcal{H}(p_X(\mathbf{x}))$, because $\mathcal{H}(p_X(\mathbf{x}))$, the entropy of p_X, is known and constant. For our AEs' training, Leaky ReLU with a negative slope ζ is used to combat neuron saturation, early stoppage and L2 regularisation with a regularisation parameter τ to limit model overfitting, and Adam to speed-up convergence. The search spaces of ζ and τ for Bayesian optimisation can be found in Section 6.6.10.

AEs, once trained, can generate time series, $\tilde{\mathbf{t}}$, by applying perturbations to encoder outputs in the latent space Z according to $\tilde{\mathbf{t}} = D_{\hat{\boldsymbol{\theta}}}(g(E_{\hat{\boldsymbol{\phi}}}(\mathbf{t})))$, where $g: Z \to \mathbb{R}^J$ is a perturbation function. This is advantages because encoding model inputs from the feature space T to Z increases the relative volume occupied by the real distribution p_T [101]. To allow tailored data generation, optimal

[5] To demonstrate why an asymmetric loss function, specifically the BCE, is better suited to modelling p_X than a symmetric loss function such as the mean squared error (MSE), below the forecasting losses ($\mathcal{L}$) and gradient magnitudes ($|\nabla|$) for two model predictions are calculated. Firstly, the $\mathcal{L}$ and $|\nabla|$ are considered when a model underpredicts a scaled target value (0.1) in the left tail of p_X by 0.075. Using the MSE, record that $\mathcal{L}^{MSE}_{0.025} = (0.025 - 0.1)^2 = 0.0056$ and $|\nabla|^{MSE}_{0.025} = |2\tilde{x}_j - 2x_j| = 0.15$, while with the BCE record that: $\mathcal{L}^{BCE}_{0.025} = -0.1 \log(0.025) - 0.9 \log(0.975) = 0.392$ and $|\nabla|^{BCE}_{0.025} = |-x_j/\tilde{x}_j + (1-x_j)/(1-\tilde{x}_j)| = 3.077$. Secondly, the $\mathcal{L}$ and $|\nabla|$ are considered when the model overpredicts the same target by 0.075. The MSE again records a loss of 0.0056 and gradient magnitude of 0.15, while the BCE yields: 0.3474 and 0.519. It is clear that the MSE treats both underprediction and overprediction of target values in the left tail of p_X identically ($\{\mathcal{L}^{MSE}_{0.025}, |\nabla|^{MSE}_{0.025}\} = \{\mathcal{L}^{MSE}_{0.175}, |\nabla|^{MSE}_{0.175}\}$), while the BCE places an emphasis on correcting underprediction ($\{\mathcal{L}^{BCE}_{0.025}, |\nabla|^{BCE}_{0.025}\} > \{\mathcal{L}^{BCE}_{0.175}, |\nabla|^{BCE}_{0.175}\}$). Notice also that BCE $|\nabla|$ dominate those of the MSE. Such asymmetric weight updating is desirable when modelling a non-uniform probability distribution as it promotes mass covering, while strictly preventing outlier generation. The results are: fewer generated data points near 0 and 1, and more generated data points near the central tendency.

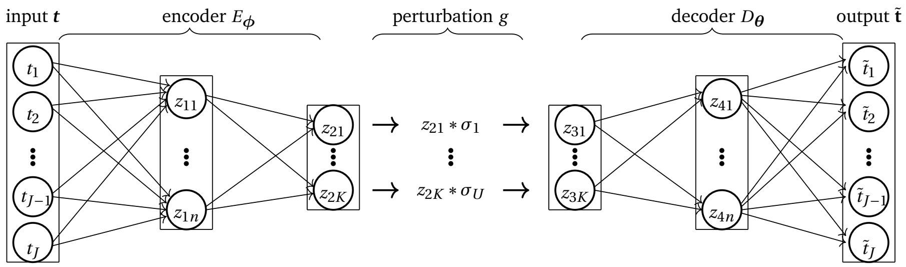


Figure 6.3: AE network architecture showing the input-output relationship for multivariate time series generation. Encoder and decoder are drawn with one hidden layer for simplicity. Each, however, usually consists of more hidden layers.

encoding and decoding functions for every DAM hourly contract are identified using Bayesian optimisation. Table 6.5 in Section 6.6.10 presents examples of encoding and decoding functions.

As shown in Figure 6.3, encoder outputs are scaled by a scaling matrix $S \sim N(1, diag(v^{\top}))$, where $v^{\top}$ is equal to the column-wise standard deviation (SD) of encoder outputs, $[\sigma_1, \ldots \sigma_U]$, multiplied by γ. Formally, a scaling latent space perturbations is applied according to: $g(\mathbf{z}) = S\mathbf{z}$. Bayesian optimisation is used to identify an optimal γ. See Section 6.6.10 for the search space of γ.

6.6.8 Variational Autoencoder Augmentation

Introducing our VAE architectures, all inference networks use a single latent variable layer with a normal prior ($p(\mathbf{z}) \sim N(0, I)$) and variational parameters λ_v: μ and σ^2. These parameters are approximated as: $N(\mathbf{z}|\mu_{\phi}(\mathbf{t}), \sigma^2_{\phi}(\mathbf{t}))$, where $\mu_{\phi}(\mathbf{t})$ and $\sigma^2_{\phi}(\mathbf{t})$ are parametrised latent variational estimates of λ_v. To train our VAEs, a variant of (6.5) is used. The weight of the forward D_{KL} regulariser in (6.5) is adjusted by a Bayesian optimised parameter δ [114]. Additionally, an L2 penalty, weighted by a Bayesian optimised parameter τ, is added. For a minibatch of m time series ($\{\mathbf{t}^i\}_{i=1}^{m} \sim p_T$), our VAE is trained according to (6.14).

$$\underset{\boldsymbol{\theta},\boldsymbol{\phi}}{\arg\min}\,\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{t}) = \frac{1}{m}\sum_{i=1}^{m}\left(-\left(\sum_{j=1}^{J}\boldsymbol{\omega}_{i,j}\right)+\delta\left(-\frac{1}{2}\sum_{u=1}^{U}\Upsilon_{i,u}\right)\right)+\tau(\parallel \boldsymbol{\theta} \parallel_2^2 + \parallel \boldsymbol{\phi} \parallel_2^2), \quad (6.14)$$

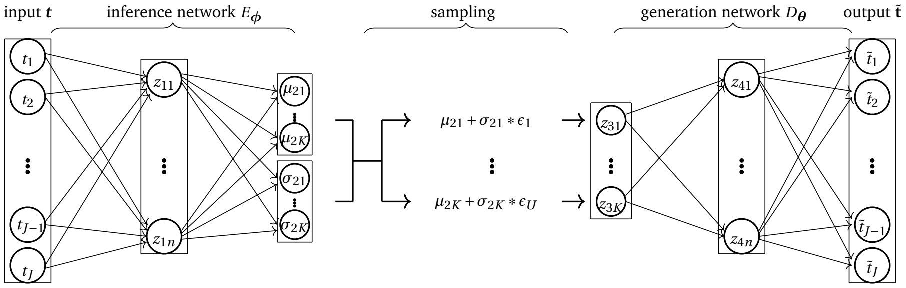


Figure 6.4: VAE network architecture showing the input-output relationship for multivariate time series generation. Inference and generation networks are drawn with one hidden layer for simplicity. Each, however, usually consists of more hidden layers. Note that a vector $\epsilon \sim N(0,I)$ yields $\{\epsilon_1,...,\epsilon_U\}$.

where $\boldsymbol{\omega}_{i,j} = p_T(t_j^i)\log p_{\boldsymbol{\theta}}(t_j^i|\mathbf{z}^i)+\big(1-p_T(t_j^i)\big)\log\big(1-p_{\boldsymbol{\theta}}(t_j^i|\mathbf{z}^i)\big)$, $\Upsilon_{i,u} = 1+\log\sigma_\phi(\mathbf{t}^i)_u^2 - \mu_\phi(\mathbf{t}^i)_u^2 - \sigma_\phi(\mathbf{t}^i)_u^2$, and J/U are the dimensionalities of feature/latent variable outputs respectively [102]. In (6.14), the first term on the right is equivalent to the BCE, while the second term is a closed-form expression of the forward D_{KL}. Bayesian optimisation is used to identify optimal inference and generation networks for each DAM contract. Table 6.5 in Section 6.6.10 presents examples of optimal inference and generation networks. Similarly to AEs, Leaky ReLU with a negative slope ζ, early stoppage, and Adam are applied. The search spaces of δ, τ and ζ for Bayesian optimisation can be found in Section 6.6.10. Once trained, by minimising the above VAE loss, the prior ($p(z) \sim N(0,I)$) is sampled and passed through the generation network to yield VAE augmented data as shown in Figure 6.4.

6.6.9 Wasserstein Generative Adversarial Network Augmentation

Summarising three generators that motivated our WGAN-GP generation approach, [116] proposed generating outputs, $\tilde{\mathbf{x}}_2$, by performing $G_\phi(\mathbf{x}_1) \to \tilde{\mathbf{x}}_2$: inputs, $\mathbf{x}_1$, are passed into G_ϕ and dropout is used to provide noise to G_ϕ. Similarly, [117] proposed generating outputs by performing $\mathcal{T}_\phi(\mathbf{x}_1) \to \tilde{\mathbf{x}}_2$ using a transformation network comparable to an autoencoder, $\mathcal{T}_\phi$, in place of G_ϕ. Finally, [118] proposed generating outputs by performing $G_\phi(\mathbf{z},\sigma) \to \tilde{\mathbf{x}}$. Noise, $\epsilon \sim N(0,\sigma^2 I)$, is added in every layer of G_ϕ to improve energy-based GAN train-

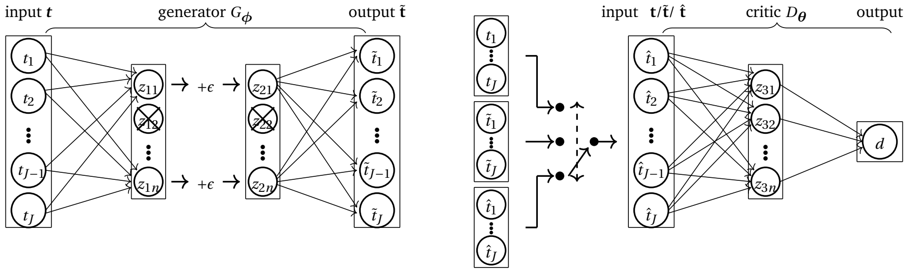


Figure 6.5: WGAN-GP network architecture showing the input-output relationship separately for a generator and a critic. Crosses in generator nodes i.e. z_{1n_1-1} represent our use of dropout. Meanwhile $\hat{\mathbf{t}} = \alpha\mathbf{t} + (1-\alpha)\tilde{\mathbf{t}}$, where $\alpha \sim U[0,1]$. The generator and the critic are drawn with one hidden layer for simplicity. Each, however, usually comprises of more than one hidden layer. To obtain the WGAN-GP loss, all inputs, namely t, $\tilde{t}$, and $\hat{t}$, must individually be passed through the critic.

ing stability. Using elements of [116–118], we propose performing $G_\phi(\mathbf{t},\sigma) \rightarrow \tilde{\mathbf{t}}$ to augment time series. Our G_ϕ accepts time series $\mathbf{t}$ as inputs and adds noise, $\epsilon \sim N(0,\sigma^2 I)$, to intermediate layer outputs. Modifying (6.10) and (6.11), our WGAN-GP is trained according to (6.15) and (6.16).

$$\begin{aligned}\arg\min_{\boldsymbol{\theta}} W_D(\boldsymbol{\theta};\mathbf{t},\tilde{\mathbf{t}}) &= \mathbb{E}_{\tilde{\mathbf{t}}\sim p_{\tilde{T}}(\tilde{\mathbf{t}})}[D_{\boldsymbol{\theta}}(\tilde{\mathbf{t}})] - \mathbb{E}_{\mathbf{t}\sim p_T(\mathbf{t})}[(D_{\boldsymbol{\theta}}(\mathbf{t}))] \\ &\quad + \lambda\mathbb{E}_{\hat{\mathbf{t}}\sim p_{\hat{T}}(\hat{\mathbf{t}})}[(\parallel \nabla_{\hat{\mathbf{t}}} D_{\boldsymbol{\theta}}(\hat{\mathbf{t}}) \parallel_2 - 1)^2], \qquad (6.15)\\ \arg\min_{\boldsymbol{\phi}} W_G(\boldsymbol{\phi};\mathbf{t}) &= -\mathbb{E}_{\mathbf{t}\sim p_T(\mathbf{t})}[D_{\boldsymbol{\theta}}(G_\phi(\mathbf{t},\sigma))] + \tau \parallel \phi \parallel_2^2, \qquad (6.16)\end{aligned}$$

where $D_{\boldsymbol{\theta}}$, referred to as the critic in WGANs, approximates the 1-Lipschitz function $f : T \rightarrow \mathbb{R}$. The L2 penalty, $\tau \parallel \phi \parallel_2^2$, is added to the generator loss to further improve training stability. During a non-exhaustive empirical evaluation it is found that L2 regularisation facilitated WGAN-GP training and increased the diversity of generated data.

Algorithm 6.1 presents our proposed method for training WGAN-GP time series augmentors. The hyperparameters referenced in Algorithm 6.1 and further described in Section 6.6.10 are selected using Bayesian optimisation for every DAM hourly contract. Similarly to [107], the negative critic loss, $-W_D(\boldsymbol{\theta};\mathbf{t},\tilde{\mathbf{t}})$, is used in our convergence criterion. Once trained, G_ϕ is employed to gen-

Algorithm 6.1 WGAN-GP for multivariate time series augmentation. The need to control both training duration and identify early instances of mode collapse is weighed against the need to avoid impeding model convergence. Following [107], default variables $\lambda = 10$ and $n_{critic} = 5$ are used.

```
Require: learning rates $\alpha_g$ and $\alpha_d$, gradient penalty coefficient $\lambda$, regularisation parameter $\tau$, standard deviations $\sigma$, minimum spread coefficient $\psi$, batch size $m$, early stopping parameters $k$, number of training iterations $K$, the number of critic updates per generator update $n_{critic}$
Require: initial discriminator parameters $\boldsymbol{\theta}$, initial generator parameters $\boldsymbol{\phi}$
        $k = 1, \dots, K$ $n = 1, \dots, n_{critic}$ $i = 1, \dots, m$
Sample training time series $\mathbf{t} \sim P_T$, and a random number $\alpha \sim U[0,1]$
$\tilde{\mathbf{t}} \leftarrow G_{\boldsymbol{\phi}}(\mathbf{t}, \sigma_g)$
$\hat{\mathbf{t}} \leftarrow \alpha \mathbf{t} + (1-\alpha)\tilde{\mathbf{t}}$
$\mathcal{L}^{(i)} \leftarrow D_{\boldsymbol{\theta}}(\tilde{\mathbf{t}}, \sigma_d) - D_{\boldsymbol{\theta}}(\mathbf{t}, \sigma_d) + \lambda(\| \nabla_{\hat{\mathbf{t}}} D_{\boldsymbol{\theta}}(\hat{\mathbf{t}}, \sigma_d) \|_2 - 1)^2$
$\boldsymbol{\theta} \leftarrow \mathrm{RMSprop}\left(\nabla_{\boldsymbol{\theta}} \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}^{(i)}, \boldsymbol{\theta}, \alpha_d\right)$
Sample a batch from the training time series $\{\mathbf{t}^{(i)}\}_{i=1}^{m} \sim P_T$
$\boldsymbol{\phi} \leftarrow \mathrm{RMSprop}\left(\nabla_{\boldsymbol{\phi}} \frac{1}{m} \sum_{i=1}^{m} [-D_{\boldsymbol{\theta}}(G_{\boldsymbol{\phi}}(\mathbf{t}^i, \sigma_g), \sigma_d)] + \tau \| \boldsymbol{\phi} \|_2^2, \boldsymbol{\phi}, \alpha_g\right)$ $k > k_{check}$
$\tilde{T} \leftarrow G_{\boldsymbol{\phi}}(T, \sigma)$ $\sum_{j=1}^{J} \mathrm{SD}(\tilde{T}) < \psi \sum_{j=1}^{J} \mathrm{SD}(T)$ or $-W_D(\boldsymbol{\theta}; T, \tilde{T})$ has not set new low for $k_{break}$
early stopping break
```

Noise, $\epsilon \sim N(0, \sigma^2 I)$, is added element-wise to intermediate G layer outputs and, similarly to [115], optionally to D.

erate multivariate time series. Example G_{ϕ} and $D_{\boldsymbol{\theta}}$ network architectures are presented in Figure 6.5 and Table 6.5.

Across a randomly selected sample of DAM contracts, the training stability of WGAN-GPs is empirically compared with the training stability of vanilla GANs, GANs with soft real and fake labels, WGANs, and context encoders, which generate regions of a target and use a compound loss function comprising of a reconstruction loss plus an adversarial loss. Overall, WGAN-GPs were found to be the most stable and easy to train. They suffered significantly fewer instances of mode collapse and were less sensitive to small hyperparameter changes.

6.6.10 Model Architectures and Hyperparameters for Augmentation Methods

Table 6.4 shows a sample of selected hyperparameters for feature space augmentors, introduced and described in Section 6.6.6. Bayesian optimisation is employed to select $\log(\sigma) \sim U\left[\log(9 \times 10^{-3}), \log(10^{-1})\right]$ and $r \in 4, \dots, 8$.

Table 6.5 presents a sample of optimal architectures for our model-based augmentation methods: AE, VAE, and WGAN-GP, described in Sections 6.6.7- 6.6.9. To combat neuron saturation, leaky ReLU activations, with a negative slope

Table 6.4: Optimised feature space augmentation hyperparameters for the Belgian 13h and Dutch 17h.

	Jittering	**Scaling**	**Mag. Warp.**
2NN (BE 13h)	$\sigma = 1.78 \times 10^{-2}$	$\sigma = 4.52 \times 10^{-2}$	$r = 4$, $\sigma = 3.20 \times 10^{-2}$
2CNN_NN (NL 17h)	$\sigma = 1.01 \times 10^{-2}$	$\sigma = 1.22 \times 10^{-2}$	$r = 6$, $\sigma = 1.51 \times 10^{-2}$
ARX (BE 13h)	$\sigma = 3.67 \times 10^{-2}$	$\sigma = 9.87 \times 10^{-2}$	$r = 7$, $\sigma = 8.98 \times 10^{-2}$

Table 6.5: Optimised model-based augmentation network architectures for the Belgian 13h and Dutch 17h and 22h.

	AE		**VAE**		**WGAN-GP**	
	Encoding	**Decoding**	**Inference**	**Generation**	**Generator**	**Critic**
2NN (BE 13h)	[25, 13, 4]	[9, 17, 24]	[17, 12, 2]	[7, 12, 24]	[27, 24]	[32, 21]
2CNN_NN (NL 17h)	[9, 3]	[9, 17, 15]	[31, 3]	[7, 12, 15]	[52, 31, 15]	[31, 1]
ARX (NL 22h)	[17, 5]	[7, 12, 13]	[31, 3]	[17, 13]	[27, 13]	[27, 13, 1]

ζ, are applied to intermediate layer outputs of both the generator and critic. For AEs, Bayesian optimisation is used to select: $\zeta \sim U[0,\ 2 \times 10^{-1}]$; $\log(\gamma) \sim U[\log(10^{-4}), \log(10^{-2})]$; and $\log(\tau) \sim U[\log(10^{-5}), \log(5 \times 10^{-3})]$. For VAEs, it is used to select: $\zeta \sim U[0, 2 \times 10^{-1}]$; $\log(\delta) \sim U[\log(10^{-4}), \log(7.5 \times 10^{-3})]$; and $\log(\tau) \sim U[\log(10^{-5}), \log(5 \times 10^{-3})]$. For WGAN-GPs, it is used to select: learning rate for the generator $\log \alpha_g \sim U[\log(10^{-5}),\ \log(2 \times 10^{-2})]$; learning rate for the discriminator $\log(\alpha_d) \sim U[\log(10^{-5}), \log(2 \times 10^{-2})]$; standard deviation of added noise for the generator $\log(\sigma_g) \sim U[\log(10^{-4}),\ \log(10^{-1})]$; standard deviation of added noise for the discriminator $\sigma_d \sim U[0, 10^{-2}]$; negative slope $\zeta \sim U[0, 2 \times 10^{-1}]$, and regularisation $\log(\tau) \sim U[\log(10^{-5}), \log(10^{-1}))]$. Hyperparameters impacting WGAN-GP training duration are set by trial and error to: m=32; K=3000; k_{check}=360; k_{end}=720; k_{break}=600; and $\psi = 2.5 \times 10^{-1}$.

6.6.11 Evaluation Setup

This chapter aims to determine whether data augmentation can significantly boost the prediction accuracies of multivariate time series regression models. To achieve this aim, across the test set, an empirical and statistical analysis are conducted for regression models trained with (Aug: $[T, \tilde{T}]$, size = $2N_{train}$) and without (Bench: $[T]$, size = N_{train}) augmentation. The specifics of our empirical and statistical evaluation procedures are detailed below.

Describing our empirical evaluation procedure, to assess the overall impacts of data augmentation, in Section 6.7 the mean absolute errors (MAE), the root mean squared errors (RMSE), and the symmetric mean absolute percentage errors (sMAPE) of regression forecasts with and without augmentation are computed and evaluated. Additionally, similarly to [89], Win / Tie / Loss scores are computed to measure an augmentor's efficiency. Instances where benchmark RMSEs decrease/increase across the test set are classified as Wins/Losses. Instances where an augmentor fails to improve upon benchmark RMSEs across the validation set are classified as Ties.

Describing our statistical evaluation procedure, to determine the statistical significance of any accuracy improvements, one-sided Wilcoxon signed-rank tests [119] are performed. The Wilcoxon test is a nonparametric test used to compare distributions of paired samples. This test is used to compare Bench and Aug RMSEs: $\{\text{RMSE}_i^{bench}\}_{i=1}^n$ and $\{\text{RMSE}_i^{aug}\}_{i=1}^n$ respectively. The test assumes symmetry between positive and negative sample differences: $d_i = \text{RMSE}_i^{aug} - \text{RMSE}_i^{bench}, \ \forall \ i \in \{1, \ldots, n\}$. To compute the Wilcoxon statistic, the rank sums of positive and negative differences are calculated, and their minimum is taken. Note, following the recommendations of [120], the Pratt ranking method is used to obtain conservative estimates of the Wilcoxon statistic. A one-sided Wilcoxon test is performed with the null hypothesis H_0: $\text{Median}(\{d_i\}_{i=1}^n) \geq 0$ and alternative hypothesis H_1: $\text{Median}(\{d_i\}_{i=1}^n) < 0$. When the Wilcoxon statistic translates to a p-value less than 0.05, H_0 is rejected, and H_1 is accepted. A p-value less than 0.05 indicates a statistically significant performance improvement at a 5% level.

6.7 Numerical Results and Discussion

To highlight overall performance changes, summary augmentation results are presented in Table 6.6. Mean forecast error percentage changes, %Δ **E**, for each country and evaluation model pair, are calculated according to: $100 \times [(\text{E}_{aug} - \text{E}_{bench}) \div \text{E}_{bench}]$, where E_{aug} and E_{bench} are the augmentation and ***Bench.*** mean forecast errors respectively, and E is either the MAE, RMSE, or sMAPE. To calculate **Summary** forecast error percentage changes, %Δ **E**s are averaged across the six evaluation cases. Bold p-values, from Wilcoxon tests, indicate statistically significant performance improvements at a 5% level. Gray highlights show the best results for each row.

Table 6.6: Augmentation results displaying average percentage changes in benchmark (***Bench.***) mean forecast errors: specifically MAEs, RMSEs, and sMAPEs.

			Jittering	**Scaling**	**Mag. Warp.**	**AE**	**VAE**	**WGAN-GP**	***Bench.***
Belgian DAM Prices	*2NN*	**%Δ MAE**	-0.36	-0.94	-2.23	-3.26	-4.07	-3.96	*9.26*
		%Δ RMSE	-0.98	-1.15	-2.14	-3.18	-3.50	-4.36	*16.47*
		%Δ sMAPE	-0.24	-1.03	-2.28	-3.49	-4.52	-4.04	*17.27*
		Win / Tie / Loss	6/13/5	4/16/4	12/7/5	15/4/5	21/1/2	15/1/8	-
		p-value	0.38	0.42	**0.03**	**0.01**	**0.00**	0.12	-
	2CNN_NN	**%Δ MAE**	0.39	0.58	1.90	-3.82	-5.79	-3.03	*9.30*
		%Δ RMSE	0.25	0.57	1.74	-2.72	-2.92	-3.54	*16.38*
		%Δ sMAPE	0.76	0.60	1.99	-3.27	-5.07	-2.41	*17.09*
		Win / Tie / Loss	7/12/5	4/16/4	5/10/9	16/1/7	17/0/7	16/0/8	-
		p-value	0.33	0.55	0.93	**0.01**	**0.01**	**0.03**	-
	ARX	**%Δ MAE**	0.57	-0.70	0.50	-1.96	-2.80	-6.03	*10.02*
		%Δ RMSE	0.22	-0.54	0.25	-1.58	-2.37	-4.23	*18.25*
		%Δ sMAPE	0.57	-0.55	0.43	-1.73	-2.77	-5.82	*18.25*
		Win / Tie / Loss	6/7/11	14/5/5	11/0/13	17/1/6	17/1/6	15/2/7	-
		p-value	0.84	**0.01**	0.66	**0.00**	**0.00**	0.06	-
Dutch DAM Prices	*2NN*	**%Δ MAE**	0.43	0.52	0.19	-0.76	-0.94	-1.33	*6.69*
		%Δ RMSE	0.28	0.15	-0.05	-0.64	-0.94	-1.13	*9.53*
		%Δ sMAPE	0.40	0.56	0.30	-0.72	-0.78	-1.15	*12.96*
		Win / Tie / Loss	6/13/5	7/10/7	6/11/7	16/2/6	14/1/9	16/0/8	-
		p-value	0.38	0.65	0.55	**0.04**	**0.02**	**0.05**	-
	2CNN_NN	**%Δ MAE**	-0.89	-0.02	0.45	-2.27	-0.60	-1.77	*6.90*
		%Δ RMSE	-0.55	0.08	0.42	-1.95	-0.79	-2.32	*9.73*
		%Δ sMAPE	-0.77	0.25	0.76	-2.35	-0.94	-1.85	*13.36*
		Win / Tie / Loss	9/15/0	4/16/4	5/13/6	17/4/3	16/2/6	19/1/4	-
		p-value	**0.00**	0.53	0.70	**0.00**	**0.03**	**0.00**	-
	ARX	**%Δ MAE**	0.05	-1.07	-0.67	-1.31	-2.18	-1.71	*6.92*
		%Δ RMSE	-0.08	-1.14	-0.94	-1.00	-2.52	-2.47	*9.97*
		%Δ sMAPE	0.22	-1.00	-0.47	-1.20	-1.98	-1.72	*13.37*
		Win / Tie / Loss	9/13/2	13/8/3	13/5/6	16/2/6	17/0/7	19/0/5	-
		p-value	**0.03**	**0.00**	**0.02**	**0.00**	**0.00**	**0.00**	-
Summary		**%Δ MAE**	0.03	-0.27	0.02	-2.23	-2.73	-2.97	*8.18*
		%Δ RMSE	-0.14	-0.34	-0.12	-1.85	-2.17	-3.01	*13.39*
		%Δ sMAPE	0.16	-0.19	0.12	-2.13	-2.67	-2.83	*15.39*
		Win / Tie / Loss	43/73/28	46/71/27	52/46/46	97/14/33	102/5/37	100/4/40	-
		p-value	0.08	**0.01**	0.27	**0.00**	**0.00**	**0.00**	-

The results indicate that not all augmentation methods can significantly boost the regression accuracies of multivariate time series models. While AEs, VAEs, and WGAN-GPs, on average, reduce benchmark summary MAEs, RMSEs, and sMAPEs by more than 2%, yielding p-values < 0.05, jittering and magnitude-warping fail to significantly improve summary benchmark MAEs and sMAPEs, yielding p-values ≥ 0.05. Beyond the summary results, AEs, VAEs, and WGAN-GPs are observed to significantly boost benchmark performances in 6/6, 6/6 and 4/6 evaluation cases respectively. Moreover, they are observed to improve 58.33% to 87.4% of benchmark RMSEs; yielding %Δ RMSEs between -4.36% and -0.64%. Meanwhile, jittering, scaling and magnitude-warping are observed to yield p-values < 0.05 in 2/6, 2/6, and 2/6 evaluation cases, and improve 16.67% to 58.33% of benchmark RMSEs; producing %Δ RMSEs between -2.14% and 1.74%. Overall, out of the augmentors evaluated, VAEs and WGAN-GPs achieve the greatest forecast error improvements. VAEs produce the highest number of Wins, while WGAN-GPs produce the greatest average %Δ MAE, %Δ RMSE, and %Δ sMAPE improvements. To facilitate further analyses, Belgian, and Dutch ARX, 2NN, and 2CNN_NN augmentation performances are analysed separately below.

To explain the relative underperformance of jittering, scaling, and magnitude-warping, remember that the success of feature space augmentation is data-dependent. Broadly, feature space augmentation either increase forecast accuracies by spurring the identification of long-term trends or decrease forecast accuracies by adding too much noise and scrambling any potential trends in time series. Analysing average ARX %Δ RMSE improvements, scaling (-0.84%) is found to perform better than magnitude-warping (-0.35%) and jittering (0.07%). We postulate that scaling, in contrast to jittering, successfully improves linear model performances, because it better regulates the amount of noise added to input time series by transforming both trends and residual errors.

In the case of deep models, it must also be noted that feature space augmentations can increase overfitting by not adding enough noise; extending training times without facilitating the identification of non-linear long-term trends. Analysing average ANN %Δ RMSE improvements, jittering (-0.25%) is found to perform better than magnitude-warping (-0.09%) and scaling (-0.01%). We postulate that, with ANNs, scaling, in contrast to jittering, generally fails to generate sufficiently distinct time series, i.e. add enough noise, to meaningfully facilitate the identification of non-linear long-term trends.

Beyond feature space augmentation, in Table 6.6, overwhelming evidence that model-based augmentors consistently generate meaningful time series capable of significantly boosting the regression accuracies of both ARXs and ANNs is found. All model-based augmentors decrease Belgian, and Dutch ARX, 2NN, and 2CNN_NN MAEs, RMSEs, as well as sMAPEs. Moreover, they yield more than 14 (58.33%) Wins across all evaluation cases. Analysing performance differences between Belgian and Dutch evaluation cases, AEs, VAEs, and WGAN-GPs are observed to reduce Belgian forecasting errors more than Dutch errors. Varying feature counts, used in the forecasting of Belgian and Dutch DAM prices, may explain these differences. Because Dutch prices are, on average, forecasted using fewer features, it is reasonable to postulate that less complex ANNs are used to forecast Dutch prices. Elaborating, per Section 6.1.1, decreased model complexity reduces the need for large training sets and, by extension, the expected forecast error reduction attainable from data augmentation.

Finally, analysing model-based augmentation performances across ARXs and ANNs, on average, model-based augmentors are observed to reduce ARX MAEs, RMSEs, and sMAPEs by 2.67%, 2.36%, and 2.54%, and ANN MAEs, RMSEs, and sMAPEs by 2.63%, 2.33%, and 2.55% respectively. While similar ARX and ANN improvements are surprising, note that they mostly result from WGAN-GP's Belgian ARX performance. On average, AEs and VAEs achieve higher forecast error reductions with ANNs than ARXs.

6.8 Conclusion

This chapter demonstrated that multivariate time series augmentation methods can significantly boost the regression accuracies of both autoregressive models with exogenous inputs and artificial neural networks. While jittering, scaling, and magnitude-warping generally struggled to improve a majority of forecast errors, AEs, VAEs, and WGAN-GPs were found to significantly reduce a majority of forecast errors; on average reducing benchmark MAEs by 2.23%, 2.73% and 2.97% respectively. Taking every result into consideration, VAEs and WGAN-GPs were found to be our best and most stable individual multivariate time series augmentors, decreasing 70.83% and 69.44% of benchmark errors.

Chapter 7

Forecasting Day-Ahead Market Prices

Price forecast errors for the day-ahead market (DAM) have grown in recent years. To reduce these forecast errors, we propose using technical indicator features described in Chapter 5, and data augmentation methods devised in Chapter 6. This chapter evaluates the impacts of technical indicators and data augmentation methods on DAM forecast errors. The chapter additionally evaluates whether ensemble methods that average multiple forecasts, obtained using technical indicators and data augmentation, could be used to further boost forecasting accuracies of DAM prices.

7.1 Introduction

To ensure the profitability of autonomous agents, accurate forecasts of day-ahead market (DAM) prices are needed. To attain accurate DAM forecasts, neural networks, which have been shown to outperform statistical models [10, 25, 28], are employed throughout this chapter. Additionally, we propose using technical indicators and data augmentation methods.

Chapter 5 underlined how the use of technical indicator features, such as moving averages and Bollinger bands, can improve the accuracy of machine learning models. Technical features assist in the identification of behavioural biases of DAM traders. Meanwhile, Chapter 6 highlighted how data augmentation methods, using autoencoders and generative adversarial networks, can be used to boost forecast accuracies.

This chapter compares forecasting results attained using technical indicators and data augmentation methods. Further, the chapter evaluates ensemble methods that average the forecasts obtained using these methods. To the best of our knowledge, we are the first to employ both technical indicator features and data augmentation methods to forecast DAM prices.

Note that while methods implemented in Chapters 5 and 6 were tested using data spanning from 2017 to 2018, in this chapter the same methods, together with our newly proposed ensemble methods, are tested using data spanning 2020. The use of a more recent dataset further highlights the veracity of previously tested methods, and demonstrates the methods intransigence to potential data drift.

Outlining the structure of this chapter, the methodology for forecasting DAM prices is described in Section 7.2. In Section 7.3 the results are outlined. Finally, in Section 7.4 the chapter is concluded.

7.2 Ensemble Approach to Day-Ahead Market Price Forecasting

The data gathering and processing steps employed in the forecasting of DAM prices are outlined below. Additionally, the evaluation procedure is described, and the DAM forecasting results are analysed.

7.2.1 Data

Dutch and Belgian DAM prices, spanning from 01/01/2016 to 11/12/2020, are collected together with load and generation day-ahead forecasts from the ENTSO-E Transparency Platform [22]. This data is subsequently split into training and test data. Data from 01/01/2016 to 01/01/2020 are used for training, while data from 01/01/2020 to 11/12/2020 are used for testing. Comparably

Table 7.1: Summary statistics for Belgian and Dutch DAM prices in €/MWh.

	mean		*standard deviation*	
	train	**test**	**train**	**test**
BE	43.95	31.04	22.93	16.38
NL	41.31	31.46	14.67	15.08

to previous chapters, the testing period is selected to span one year. It is also selected to immediately follow the training period for each DAM contract.

Note that hyperparameter optimisation is not performed in this chapter. Instead, features and optimised parameters from Chapters 5 and 6 are used directly. Consequently, no validation set is required.

Summary statistics of DAM prices for the training and test sets are shown in Table 7.1. From Table 7.1 we discern that Belgian DAM prices in the test set have a lower standard deviation than historic DAM prices.

Analysing train and test set differences further, note that Table 7.1 additionally highlights differences in mean DAM prices. The means of Belgian and Dutch DAM prices are higher across the training set than test set. This contrasts the trend observed across train and test data in Chapters 5 and 6.

7.2.2 Data Processing

Min-max scaling is applied across a 30-day rolling window to scale features. A 30-day window is selected to counteract market seasonalities.

7.2.3 Prediction

The forecasting accuracies of a two-layer neural network (2NN), consisting of two intermediate fully connected layers, and a joint three-layer network (2CNN_NN), consisting of two intermediate convolutional layers and a single intermediate fully connected layer, are assessed. Parameters and features from Chapters 5 and 6 are used. L2 regularisation, ReLU activation functions and Adam are employed to improve training stability.

7.2.4 Ensemble Forecasts of Day-Ahead Market Prices

The aforementioned forecasting models are evaluated with technical indicator (TI) feature inputs from Chapter 5, and with the addition of augmented data from Chapter 6. The best performing TIs and augmentors from these chapters are utilised in this chapter. Following Chapter 5, for instance, exponential moving average TI, i.e. EMA($s = 22$), is evaluated with 2NN forecasting models and rate of change TI, i.e. ROC($n = 9$), is evaluated with 2CNN_NN models. Following Chapter 6 three augmentation methods using autoencoders (AE), variational encoders (VAE) and Wasserstein generative adversarial networks with a gradient penalty (WGAN-GP) are evaluated. Each of these individual methods is considered as a benchmark against their combined ensemble method.

Ensemble forecasts, obtained by averaging the forecasts from multiple models, are assessed. Evaluated ensemble forecasts are listed below:

- **AE+VAE:** To calculate AE+VAE ensemble forecasts, firstly, a forecasting model, either 2NN or 2CNN_NN, is trained using both real data and AE augmented data. Secondly, a forecasting model is trained using both real and VAE augmented data. Finally, the forecasts from these models are averaged according to (7.1) to obtain AE+VAE forecasts.

$$\hat{p}^{\text{NN}}_{\text{AE+VAE}} = 1/2(\hat{p}^{\text{NN}}_{\text{AE}} + \hat{p}^{\text{NN}}_{\text{VAE}}), \tag{7.1}$$

  where NN refers to either a 2NN or 2CNN_NN forecasting model, $\hat{p}^{\text{NN}}_{\text{AE}}$ are forecasts obtained using a model trained using real and AE augmented data, $\hat{p}^{\text{NN}}_{\text{VAE}}$ are forecasts obtained using a model trained using real and VAE augmented data, and $\hat{p}^{\text{NN}}_{\text{AE+VAE}}$ are ensemble AE+VAE forecasts.

- **AE+WGAN-GP:** In order to calculate AE+WGAN-GP ensemble forecasts, firstly, a forecasting model is trained using real data and AE augmented data. Secondly, a forecasting model is trained using real and WGAN-GP augmented data. Finally, the forecasts from these models are averaged according to (7.2) to obtain AE+WGAN-GP forecasts.

$$\hat{p}^{\text{NN}}_{\text{AE+WGAN-GP}} = 1/2(\hat{p}^{\text{NN}}_{\text{AE}} + \hat{p}^{\text{NN}}_{\text{WGAN-GP}}), \tag{7.2}$$

  where $\hat{p}^{\text{NN}}_{\text{WGAN-GP}}$ are forecasts obtained using a model trained with both real, and WGAN-GP augmented data, and $\hat{p}^{\text{NN}}_{\text{AE+WGAN-GP}}$ are ensemble AE+WGAN-GP forecasts.

- **VAE+WGAN-GP:** To calculate VAE+WGAN-GP ensemble forecasts, firstly, a forecasting model is trained using real data and VAE augmented data. Secondly, a forecasting model is trained using real and WGAN-GP augmented data. Finally, the forecasts from these models are averaged according to (7.3) to obtain VAE+WGAN-GP forecasts.

  $$\hat{p}^{\text{NN}}_{\text{VAE+WGAN-GP}} = 1/2(\hat{p}^{\text{NN}}_{\text{VAE}} + \hat{p}^{\text{NN}}_{\text{WGAN-GP}}), \tag{7.3}$$

  where $\hat{p}^{\text{NN}}_{\text{VAE+WGAN-GP}}$ are ensemble VAE+WGAN-GP forecasts.

- **AE+TI:** To calculate AE+TI ensemble forecasts, firstly, a forecasting model is trained using real data and AE augmented data. Secondly, a forecasting model is trained using TI features as inputs. Finally, the forecasts from these two models are averaged according to (7.4).

  $$\hat{p}^{\text{NN}}_{\text{AE+TI}} = 1/2(\hat{p}^{\text{NN}}_{\text{AE}} + \hat{p}^{\text{NN}}_{\text{TI}}), \tag{7.4}$$

  where $\hat{p}^{\text{NN}}_{\text{TI}}$ is the NN forecast using TI features as inputs, and $\hat{p}^{\text{NN}}_{\text{AE+TI}}$ is the AE+TI ensemble forecast.

- **VAE+TI:** To calculate VAE+TI ensemble forecasts, firstly, a forecasting model is trained using real data and VAE augmented data. Secondly, a forecasting model is trained using TI features as inputs. Finally, the forecasts from these models are averaged according to (7.5).

  $$\hat{p}^{\text{NN}}_{\text{VAE+TI}} = 1/2(\hat{p}^{\text{NN}}_{\text{VAE}} + \hat{p}^{\text{NN}}_{\text{TI}}), \tag{7.5}$$

  where $\hat{p}^{\text{NN}}_{\text{VAE+TI}}$ is the VAE+TI ensemble forecast.

- **WGAN-GP+TI:** To calculate WGAN-GP+TI ensemble forecasts, firstly, a forecasting model is trained using real data and WGAN-GP augmented data. Secondly, a forecasting model is trained using TI features as inputs. Finally, the forecasts from these models are averaged according to (7.6) to obtain WGAN-GP+TI forecasts.

  $$\hat{p}^{\text{NN}}_{\text{WGAN-GP+TI}} = 1/2(\hat{p}^{\text{NN}}_{\text{WGAN-GP}} + \hat{p}^{\text{NN}}_{\text{TI}}), \tag{7.6}$$

  where $\hat{p}^{\text{NN}}_{\text{WGAN-GP+TI}}$ is the WGAN-GP+TI ensemble forecast.

- **AE+VAE+WGAN-GP:** After training forecasting models with real data, and AE, VAE and WGAN-GP augmented data, AE+VAE+WGAN-GP ensemble forecasts are calculated according to (7.7).

$$\hat{p}^{\text{NN}}_{\text{AE+VAE+WGAN-GP}} = 1/3(\hat{p}^{\text{NN}}_{\text{AE}} + \hat{p}^{\text{NN}}_{\text{VAE}} + \hat{p}^{\text{NN}}_{\text{WGAN-GP}}), \tag{7.7}$$

  where $\hat{p}^{\text{NN}}_{\text{AE+VAE+WGAN-GP}}$ are AE+VAE+WGAN-GP ensemble forecasts.

- **AE+VAE+WGAN-GP+TI:** After training forecasting models with real and AE/VAE/WGAN-GP augmented data, and after training a forecasting model with TI features as inputs, AE+VAE+WGAN-GP+TI ensemble forecasts are determined according to (7.8).

$$\hat{p}^{\text{NN}}_{\text{AE+VAE+WGAN-GP+TI}} = 1/4(\hat{p}^{\text{NN}}_{\text{AE}} + \hat{p}^{\text{NN}}_{\text{VAE}} + \hat{p}^{\text{NN}}_{\text{WGAN-GP}} + \hat{p}^{\text{NN}}_{\text{TI}}), \tag{7.8}$$

  where $\hat{p}^{\text{NN}}_{\text{AE+VAE+WGAN-GP+TI}}$ are AE+VAE+WGAN-GP+TI forecasts.

7.2.5 Evaluation

The mean absolute errors (MAE) are used to evaluate forecasting accuracy. Additionally, to facilitate relative evaluation, naive benchmark forecasts are computed according to (7.9):

$$\hat{p}^{\text{BENCH}}_{d+1} = 1/2(p^{dam}_{d} + p^{dam}_{d-1}). \tag{7.9}$$

where p^{dam}_{d} is DAM prices on day d. The benchmark is a two-day moving average of DAM prices.

7.3 Results and Discussion

The forecasting accuracies of models are summarised in Table 7.2. Analysing the results, the benchmark, BENCH, is observed to yield the highest average MAE of 8.07. Meanwhile, TI is observed to yield an average MAE of 7.56; 6.20% lower than BENCH. Improving forecasting accuracies further, data augmentation methods, namely AE, VAE, and WGAN-GP, are observed to yield average MAEs of 6.47, 6.45, and 6.39 respectively; up to 20.82% lower than BENCH. The high performance of data augmentation methods highlights the importance of the training set size in reducing the generalisation error of DAM forecasts.

Table 7.2: MAE results on the test set for DAM price forecasting methods.

	NL		*BE*		**average**
	2NN	**2CNN_NN**	**2NN**	**2CNN_NN**	
AE	5.70	5.62	7.27	7.28	6.47
VAE	5.72	5.69	7.04	7.33	6.45
WGAN-GP	5.78	5.65	6.89	7.23	6.39
TI	6.54	7.61	8.15	7.96	7.57
AE+VAE	5.58	5.53	6.71	6.81	6.16
AE+WGAN-GP	5.60	5.52	6.59	6.84	6.14
VAE+WGAN-GP	5.61	5.55	6.47	6.80	6.11
AE+TI	5.54	6.04	6.55	6.57	6.18
VAE+TI	5.55	6.12	6.37	6.48	6.13
WGAN-GP+TI	5.54	6.08	6.42	6.56	6.15
AE+VAE+WGAN-GP	5.56	5.49	6.40	6.63	6.02
AE+VAE+WGAN-GP+TI	5.37	5.62	6.14	6.24	5.84
BENCH		7.75		8.38	8.07

Combining forecasts from the above methods boosts accuracies even further. For example, AE+WGAN-GP, which generates forecasts by averaging AE and WGAN-GP forecasts, yields an average MAE of 6.14. This is 5.10% and 3.91% lower than AE and WGAN-GP methods respectively. Overall, AE+VAE+WGAN-GP+TI is observed to yield the lowest average MAE of 5.84. The ensemble method, using both TI features and data augmentation, outperforms TI by 22.85%, and AE, VAE, and WGAN-GP on average by 9.27%.

7.4 Conclusion

In this chapter, day-ahead market (DAM) prices were predicted by employing both technical indicator features and data augmentation methods. Data augmentation methods, namely autoencoders (AE), variational autoencoders (VAE) and Wasserstein generative adversarial networks with a gradient penalty (WGAN-GP), were found to outperform technical indicators (TI). Overall, however, an ensemble method, AE+VAE+WGAN-GP+TI, averaging AE, VAE, WGAN-GP, and TI model forecasts was found to outperform all other evaluated forecasting methods.

Chapter 8

Forecasting Continuous Intraday Market Prices

Continuous intraday market (CID) trading and volatility have grown in recent years. Increasing renewable energy production has spurred this growth, but induced price uncertainty in CID trading strategies. To reduce uncertainty and increase the profitability of trading strategies, more accurate CID price forecasts are needed. If forecast accuracies of CID prices are to improve, new features and models capable of capturing the effects of continuous trading must be identified. This chapter evaluates novel features that have been shown to have high explanatory power in forecasting CID prices. Moreover, the chapter compares the forecast accuracies of machine learning models using these inputs.

Note that while most CID studies only focus on forecasting the volume-weighted average price of trades (VWAP), this chapter focuses on predicting elements of the trade book (TB) and limit-order book (LOB). For instance, the VWAP, the lowest traded price (LOW) and the highest traded price (HIGH), the average best ask price (ASK), the average best bid price (BID), and the average mid-price (MID) are all predicted.

8.1 Introduction

In earlier chapters, we highlighted that the transition from non-renewable to renewable sources of energy in recent years has negatively impacted price forecast accuracies. Traded volume and liquidity on the CID have grown. And the number of trading algorithms/agents operating on the CID has increased. To improve agents' decision-making, increase profits, and reduce overall risk exposure, accurate CID price predictions are needed.

In Chapter 4, the literature review highlighted that CID price forecasting studies have thus far focused predominantly on forecasting the VWAP of the ID_3 index[1] using readily available features, such as the lagged VWAP, and seasonal and exogenous features. By not using other types of features from the LOB and TB, and by not predicting all CID prices, existing studies have thus far ignored or failed to account for significant continuous trading characteristics.

Given the potential to increase the performance of CID trading agents by expanding the feature input space and forecasting all CID prices, this chapter focuses on feature engineering novel features by extracting statistical information from the LOB and TB. Additionally, the focus is placed on forecasting all CID prices, instead of just the VWAP of the ID_3 index. Summarising the three contributions of this chapter, to the best of our knowledge, we are the first to:

- predict CID prices: VWAP of the whole trading session, LOW, HIGH, ASK, BID and MID,
- engineer novel features using the LOB and TB for forecasting CID prices,
- evaluate machine learning models, such as the least absolute shrinkage and selection operator (LASSO), random forest (RF), gradient boosting (GB), deep neural networks (DNN) and an ensemble model, for forecasting CID prices.

Outlining the structure of this chapter, CID prices are introduced in Section 8.2. The methodology for forecasting CID prices is described in Section 8.3. In Section 8.4, we analyse the results from our CID forecasting studies. Finally, in Section 8.5 the chapter is concluded.

[1] The VWAP of the ID_3 index is calculated only for the last three hours of a trading session.

8.2 Continuous Intraday Market Prices

TB prices capture information about executed trades. Given N, i.e. the number of transactions executed for a contract, TB prices are defined as:

- **The Volume-Weighted Average Price of Trades (VWAP):** The VWAP is the ratio between the cumulative traded amount and the cumulative traded volume over an entire trading session for a contract. Formally, the VWAP is calculated according to (8.1).

$$p^{vwap} = \sum_{i=1}^{N}(p_i^{traded} \times q_i^{traded}) \;/\; \sum_{i=1}^{N} q_i^{traded}, \tag{8.1}$$

  where p^{traded} and q^{traded} are the traded price and quantity for one transaction respectively.

- **The Highest Traded Price (HIGH):** The HIGH is the highest traded price across a trading session for a contract. Formally, the HIGH is calculated according to (8.2).

$$p^{high} = \max\,\{p_1^{traded},\dots p_N^{traded}\} \tag{8.2}$$

- **The Lowest Traded Price (LOW):** The LOW is the lowest traded price across a trading session for a contract. Formally, the LOW is calculated according to (8.3).

$$p^{low} = \min\,\{p_1^{traded},\dots p_N^{traded}\} \tag{8.3}$$

LOB prices have the potential to capture significant CID price drivers. Figure 2.4 presents a snapshot of the LOB. Assuming the CID trading session is discretised according to $t \in [1, T]$, where T is the total number of time steps for a contract, LOB prices are formally defined as:

- **The Average Best Ask Price (ASK):** The ASK is the average best ask price over an entire contract's trading session. Formally, the ASK is calculated according to (8.4).

$$\bar{p}^a = 1/T \sum_{t=1}^{T} p_t^a \tag{8.4}$$

- **The Average Best Bid Price (BID):** The BID is the average best bid price over an entire contract's trading session. Formally, the BID is calculated according to (8.5).

$$\bar{p}^b = 1/T \sum_{t=1}^{T} p_t^b \tag{8.5}$$

- **The Average Mid-Price (MID):** The MID is the average mid-price over an entire contract's trading session. The MID is calculated according to (8.6).

$$\bar{p}^{mid} = 1/T \sum_{t=1}^{T} p_t^{mid} \tag{8.6}$$

8.3 Feature Engineering Approach to Continuous Intraday Market Price Forecasting

To forecast CID prices, the use of machine learning models that take novel features as inputs is proposed in this chapter. To this end, various steps and procedures, such as data gathering, feature engineering, data processing, cross-validation, feature selection, modelling and evaluation, are implemented. These steps and procedures are outlined below.

8.3.1 Data

LOB and TB data spanning from 01/01/2020 to 11/12/2020 is queried from Scholt Energy [109].[2] The data originates from the Single Intraday Coupled Market (CID), available for the Dutch Market. Day-ahead forecasts of temperature, wind speed, solar irradiance and precipitation are also gathered from Scholt Energy [109]. Finally, load and generation day-ahead forecasts are collected from the ENTSO-E Transparency Platform [22].

Table 8.1 presents summary statistics for CID prices. These prices show similar averages (mean) and standard deviations (std). However, HIGH has a higher mean and std than LOW. Similarly, ASK has a higher mean and std than BID.

[2] Please note that some contracts were missing from the dataset. Also, note that some contracts were removed from the data set for failing data quality tests.

Table 8.1: Summary statistics for CID prices in €/MWh.

	VWAP	MID	BID	ASK	LOW	HIGH
mean	30.45	31.17	28.64	33.71	24.86	34.10
std	24.35	20.82	19.88	22.70	18.29	30.17

8.3.2 Feature Engineering

Using the LOB of each hourly contract, CID price drivers are gathered. To have an impact, LOB features must capture relevant information required to accurately forecast CID prices. Detailing our proposed LOB features:

- **# bid orders:** The total number of submitted bid orders for a contract.
- **# ask orders:** The total number of submitted ask orders for a contract.
- **# orders:** The total number of submitted orders for a contract.
- **# revisions:** The total number of important revisions for a contract. Note that a revision number is updated when the LOB changes, e.g. a new order is received. Important revision numbers that change either the best ask price or the best bid price are considered and the remaining revision numbers are excluded.
- **min of p^a:** The best offered ask price, the lowest price among all submitted ask orders, for a contract.
- **std of p^a:** The standard deviation of all submitted ask order prices for a contract.
- **max of p^b:** The best offered bid price, the highest price among all submitted bid orders, for a contract.
- **std of p^b:** The standard deviation of all submitted bid order prices for a contract.
- $\bar{p}^{mid}$ **hour index-**1/2/3/4**:** Hour indices refer to specific time intervals of a trading session. For instance, hour index-1 is the last hour of a trading session, i.e. $[T-1hour,\ T]$. Hour index-i is $[T-ihour,\ T-i+1hour]$. The

$\bar{p}^{mid}$ hour index-i is calculated by averaging all mid-prices across hour index-i.

- $\bar{p}^{spread}$**:** The price spread between the best ask price and the best bid price p^{spread} at time t is defined in Section 2.3.2. The average price spread $\bar{p}^{spread}$ for a contract is calculated as $\bar{p}^{spread} = 1/T \sum_{t=1}^{T} p_t^{spread}$.

- 1$^{\text{st}}$ **quantile of cumulative ask quantities:** The 25th percentile (lower quartile) of the cumulative ask quantity distribution.

- 2$^{\text{nd}}$ **quantile of cumulative ask quantities:** The 50th percentile (median) of the cumulative ask quantity distribution.

- 3$^{\text{rd}}$ **quantile of cumulative ask quantities:** The 75th percentile (upper quartile) of the cumulative ask quantity distribution.

- 1$^{\text{st}}$ **quantile of cumulative bid quantities:** The 25th percentile (lower quartile) of the cumulative bid quantity distribution.

- 2$^{\text{nd}}$ **quantile of cumulative bid quantities:** The 50th percentile (median) of the cumulative bid quantity distribution.

- 3$^{\text{rd}}$ **quantile of cumulative bid quantities:** The 75th percentile (upper quartile) of the cumulative bid quantity distribution.

Using the TB of each hourly contract, information about trades is also extracted. Detailing our proposed TB features:

- p^{vwap} **of exports:** The vwap calculated based on exported traded prices for a contract.

- p^{vwap} **of imports:** The vwap calculated based on imported traded prices for a contract.

- p^{vwap} **of locals:** The vwap calculated based on traded prices within a country for a contract.

- p^{vwap} **hour index-**1/2/3/4**:** The vwap during a specific hour index for a contract.

- **spread** p^{high} **and** p^{low}**:** The spread between p^{high} and p^{low}.

- **open price:** The first traded price for a contract.
- **std of traded prices:** The standard deviation of traded prices for a contract.
- **traded volume:** The total traded volume for a contract.
- **traded volume hour index-1/2/3/4:** The total traded volume during a specific hour index for a contract.
- **exported traded volume:** The total exported traded volume to other countries for a contract.
- **imported traded volume:** The total imported traded volume from other countries for a contract.
- **locally traded volume:** The total traded volume within a country for a contract.
- **# transactions:** The total number of transactions for a contract.
- **binary import:** The binary import feature is 0 if a country exports more than imports for a contract. Otherwise, it is 1.
- **binary trade hour index-1/2/3/4:** The binary trade feature is 1 if a country trades with another country during a specific hour index for a contract. Otherwise, it is 0.

Additionally, other possible price drivers are obtained by adding exogenous features to the input space. Forecasts of wind speed, temperature, solar irradiance, precipitation, and Dutch and Belgian DAM prices are evaluated. Seasonal features such as day of the week, hour of the day, holiday, and month of the year are evaluated as well. These categorical seasonal features are processed using one-hot encoding.

8.3.3 Data Processing

Min-Max scaling is used for each feature. Remember that min-max scaling bounds each value $a \in A$ between 0 and 1 according to the formula: $a^* =$

$(a - A^{min})/(A^{max} - A^{min})$, where A^{min} and A^{max} are the minimum and maximum values of a set A, and a^* is the scaled value of a.

8.3.4 Cross Validation

Describing training, validation, and testing procedures, given a need to accumulate forecasts for the entire dataset (from 01/01/2020 to 11/12/2020) an iterative procedure is employed to obtain test forecasts for all 49 weeks of inputs. In one iteration, the available data is split into roughly 48 weeks of training and 1 week of test data. 48-fold cross-validation is subsequently employed across the training data; yielding an optimised model, which is used to generate forecasts of CID prices for the test data. This process is repeated 49 times until forecasts are obtained for the entire dataset.

8.3.5 Feature Selection

Feature selection commences by removing highly correlated features from the data set using the Pearson correlation coefficient as a reference. Highly correlated features, with a correlation larger than 0.8, are removed. Known electricity market characteristics are accounted for by adding day-lagged and hour-lagged features. For instance, to forecast VWAP of contract/expiry 16 on day $d+1$, i.e. p^{vwap} of h_{16}^{d+1}, a fourteen-day-lagged period h_{16}^{d-13}, … h_{15}^{d}, h_{16}^{d}, h_{17}^{d} is considered for each feature.

The feature selection process is finalised by removing the least important features from the data set using the least absolute shrinkage and selection operator (LASSO) [121]. Note that LASSO parameters, alpha $\in (0,\ 10]$ and tolerance $\in (10^{-5},\ 10^{-1}]$, are optimised.

8.3.6 Model Training and Prediction

The ability of regression models, such as LASSO, random forest (RF) [53], gradient boosting (GB) [55] and deep neural networks (DNN), to predict CID prices is evaluated. Each model has a unique hyperparameter set to be optimised by minimising the forecast error across the validation set. For example, LASSO has the alpha $\in (0,\ 10]$ and tolerance $\in (10^{-5},\ 10^{-1}]$, RF has the number of estimators (# estimators) $\in [30,\ 200]$, and GB has the maximum depth (max depth) $\in [2,\ 14]$. Similarly, DNN has the number of hidden layers (# layers) $\in [1,\ 4]$, the size of these layers (# neurons) $\in [32,\ 2048]$ and the learning rate $\in [10^{-4},\ 10^{-1}]$.

Table 8.2: Selected average number of features and LASSO parameters.

	VWAP	MID	BID	ASK	LOW	HIGH
# features	37	42	42	42	43	30
alpha	0.21	0.19	0.27	0.20	0.23	0.32
tolerance	0.06	0.05	0.05	0.06	0.05	0.05

Note that DNN models are constructed with fully connected layers, ReLU activation functions and Adam optimisers. Additionally, L2 regularisation is implemented to reduce the possibility of overfitting.

An ensemble method, which averages all evaluated forecasts, as shown in (8.7), is further considered.

$$\hat{p}^{\mathrm{ENSEMBLE}} = 1/4(\hat{p}^{\mathrm{LASSO}} + \hat{p}^{\mathrm{GB}} + \hat{p}^{\mathrm{RF}} + \hat{p}^{\mathrm{DNN}}) \tag{8.7}$$

In summary, four individual models, namely LASSO, RF, GB, and DNN, and one ensemble model are evaluated. The hyperparameters of the individual models are optimised using 48-fold cross-validation.

Note that in this thesis, CID prices are predicted before 12:00. CID forecasts are thus obtained before DAM settlement.

8.3.7 Evaluation

Mean absolute error (MAE) is used to evaluate the accuracy of price forecasts. Furthermore, a naive benchmark is considered to facilitate a relative evaluation of forecast accuracies. As a benchmark, a two-day moving average, calculated according to (8.8), is computed.

$$\hat{p}^{\mathrm{BENCH}}_{d+1} = 1/2(p_d + p_{d-1}), \tag{8.8}$$

where p is either p^{vwap}, $\bar{p}^{mid}$, $\bar{p}^{b}$, $\bar{p}^{a}$, p^{low}, or p^{high}.

Table 8.3: Importance checklist for the trade book (TB) features.

	VWAP	MID	BID	ASK	LOW	HIGH
p^{vwap}	✓				✓	✓
p^{vwap} of exports	✓	✓	✓	✓		✓
p^{vwap} of imports						✓
p^{vwap} of locals		✓		✓		
p^{vwap} hour-2		✓	✓		✓	✓
p^{vwap} hour-3	✓	✓	✓	✓	✓	
p^{vwap} hour-4						✓
p^{high}				✓		✓
p^{low}	✓	✓	✓	✓	✓	✓
spread p^{high} and p^{low}	✓					✓
open price	✓	✓			✓	✓
std of traded prices						
traded volume						
traded volume hour-1						
traded volume hour-2						
traded volume hour-3						
traded volume hour-4						
exported traded volume						
imported traded volume						
locally traded volume						
# transactions						
binary import	✓	✓	✓	✓	✓	✓
binary trade hour-1	✓	✓	✓	✓		✓
binary trade hour-2	✓	✓	✓	✓	✓	✓
binary trade hour-3	✓	✓	✓	✓	✓	✓
binary trade hour-4	✓	✓	✓	✓	✓	✓

8.4 Results and Discussion

8.4.1 Feature Selection

Table 8.2 presents optimisation results for the feature selection method described in Section 8.3.5. On average, 40 features are selected to predict CID prices. Below, the selected features for forecasting each CID price are detailed. Note that fourteen-day-lagged, seven-day-lagged, two-day-lagged and one-day-lagged LOB and TB features are often selected.

Table 8.4: Importance checklist for the limit order book (LOB) features.

	VWAP	MID	BID	ASK	LOW	HIGH
# bid orders	✓	✓	✓	✓	✓	✓
# ask orders	✓				✓	✓
# orders	✓					✓
# revisions	✓	✓	✓	✓	✓	✓
min of p^a	✓	✓	✓	✓	✓	✓
std of p^a						
max of p^b		✓				
std of p^b						
$\bar{p}^{mid}$		✓				
$\bar{p}^{mid}$ **hour-1**						
$\bar{p}^{mid}$ **hour-2**						
$\bar{p}^{mid}$ **hour-3**						
$\bar{p}^{mid}$ **hour-4**	✓	✓	✓	✓	✓	✓
$\bar{p}^{mid}$ **hour-5**						
$\bar{p}^{b}$			✓		✓	
$\bar{p}^{a}$		✓	✓	✓		
$\bar{p}^{spread}$						
1**st quantile of cumulative ask quantities**						
2**nd quantile of cumulative ask quantities**						
3**rd quantile of cumulative ask quantities**						
1**st quantile of cumulative bid quantities**						
2**nd quantile of cumulative bid quantities**						
3**rd quantile of cumulative bid quantities**						

As Table 8.3 shows, features reflecting traded price information are frequently selected. While the literature focuses on p^{vwap}, the results show that other TB prices, such as p^{high}, p^{low} and the open price, can aid in predicting CID prices.

Similarly, binary features are also found to have high explanatory power. The binary import feature, which is 0 if a country exports more energy than it imports, and 1 otherwise, for example, is found to be an important feature in predicting all CID prices, highlighting how countries' import/export behaviours can drive both TB and LOB prices.

Unlike the traded price and binary features, features reflecting traded volume are, however, not found to be important in predicting CID prices. A potential

Table 8.5: Importance checklist for the exogenous features.

	VWAP	MID	BID	ASK	LOW	HIGH
wind speed forecast	✓	✓	✓	✓	✓	✓
temperature forecast		✓	✓		✓	✓
solar irradiance forecast	✓	✓	✓	✓	✓	✓
precipitation forecast	✓			✓		✓
Dutch DAM price forecast	✓	✓	✓	✓	✓	✓
Belgian DAM price forecast		✓				

Table 8.6: Importance checklist for the seasonal features.

	VWAP	MID	BID	ASK	LOW	HIGH
holiday	✓	✓	✓	✓	✓	✓
day of the week	✓	✓	✓	✓	✓	✓
hour of the day	✓	✓	✓	✓	✓	✓
month of the year	✓	✓	✓	✓	✓	✓

reason for this may be that market participants/agents avoid using volume-related features in their decision-making processes.

As Table 8.4 shows, four LOB features, namely # bid orders, # revisions, min of p^a and $\bar{p}^{mid}$ hour-4, are found to be important features in predicting all CID prices. The # bid orders and # revisions offer information about the liquidity of the CID and proved to be important CID price drivers. While the success of these features, to some extent, is expected, the success of min of p^a and $\bar{p}^{mid}$ hour-4 is surprising. The importance of min of p^a shows the powerful effect of the lowest offered ask price on CID prices throughout a trading session. The importance of $\bar{p}^{mid}$ hour-4 indicates very active CID trading in this hour index compared to the other hours of a trading session.

As Table 8.5 presents, most exogenous features are found to be important features in predicting CID prices. Three features, namely wind speed forecasts, solar irradiance forecasts and Dutch DAM price forecasts, are found to be important features in predicting all CID prices. This result was expected since wind and solar forecasts, and DAM prices are widely used in the literature when forecasting CID prices.

Table 8.7: Selected average parameters.

		VWAP	MID	BID	ASK	LOW	HIGH
LASSO	alpha	0.21	0.18	0.27	0.20	0.22	0.33
	tolerance	0.06	0.06	0.05	0.04	0.06	0.05
GB	max depth	6	5	5	5	5	6
RF	# estimators	115	104	110	115	105	115
DNN	learning rate	0.007	0.007	0.005	0.007	0.004	0.007
	# layers	2	2	2	2	2	2
	# neurons	(272, 408)	(210, 404)	(287, 484)	(365, 617)	(301, 514)	(215, 400)

Finally analysing seasonal features, as Table 8.6 shows, features capturing seasonal effects are found to be important in predicting all CID prices. The results show that even holidays and month of the year features are important predictors. Among seasonal features, day of the week and hour of the day dummies were expected to be important, owing to their widespread use in the literature. Holiday and month of year dummies, however, were not expected to be important, owing to their infrequent use in the literature.

8.4.2 Prediction

Table 8.7 shows hyperparameter optimisation results. The table shows that similar hyperparameters are frequently selected by evaluated models in predicting CID prices. DNNs, for instance, with a two hidden layer architecture are more frequently selected.

Table 8.8 presents the forecast accuracies of evaluated models. Overall, BENCH is found to yield the highest average MAE of 11.67. Using novel features as inputs, individual machine learning models, LASSO, GB, and RF, for example, yield average MAEs of 8.09, 8.03, and 8.37 respectively; outperforming BENCH by roughly 30%. Among individual models, DNN is the best model and obtains the lowest average MAE of 7.62; outperforming BENCH by 35%. ENSEMBLE, which takes the average of all model forecasts, further improves forecast accuracies; yielding the lowest average MAE of 7.44.

8.5 Conclusion

In this chapter, continuous intraday market (CID) prices, namely the volume-weighted average price of trades (VWAP), the lowest traded price (LOW), the

Table 8.8: MAE results on the test set.

	LASSO	GB	RF	DNN	ENSEMBLE	BENCH
VWAP	8.27	8.22	8.58	7.78	7.59	12.25
MID	7.43	7.40	7.75	6.97	6.83	10.59
BID	7.14	7.09	7.41	6.77	6.47	10.40
ASK	8.37	8.40	8.75	8.00	7.75	11.98
LOW	7.83	7.70	8.14	7.22	7.25	11.50
HIGH	9.51	9.36	9.59	8.95	8.73	13.27
average	8.09	8.03	8.37	7.62	7.44	11.67

highest traded price (HIGH), the average best ask price (ASK), the average best bid price (BID), and the average mid-price (MID), were predicted. To predict these prices, feature engineering was utilised to extract statistical information from the limit order book (LOB) and trade book (TB). Subsequently, feature selection, using LASSO, was employed to evaluate the importance/explanatory power of created features.

Overall, the feature selection process revealed that exogenous features, such as wind speed forecasts, solar irradiance forecasts, day-ahead market price forecasts, and seasonal features, such as holiday, day of the week, hour of the day and month of the year, are important predictors of CID prices. Additionally, the process revealed that novel features capturing price information are more important than features capturing volume information. Among TB features, the VWAP of exported trades, the VWAP of hour-3, the LOW, the binary import, and binary trade hours-1/2/3/4 were found to be the most important predictors in forecasting a majority of CID prices. Among LOB novel features the # bid orders, the # revisions, the min of p^a, and the $\bar{p}^{mid}$ hour-4 were found to be the most important features in predicting all CID prices.

Using selected feature inputs, the forecasting accuracies of a range of machine learning models were evaluated. LASSO, random forest (RF) and gradient boosting (GB) were found to outperform a naive benchmark. Deep neural networks (DNN), meanwhile, were found to outperform all other individual models. Ensemble modelling, i.e. taking the average of all model forecasts, was found to further improve CID forecast accuracies.

Chapter 9

Forecasting Balancing Market Prices

Accurately predicted balancing market (BAL) prices are needed to ensure the profitability of arbitrage trading strategies. To predict BAL prices, this chapter proposes and evaluates a range of machine learning models, such as the least absolute shrinkage and selection operator (LASSO), random forest (RF), gradient boosting (GB), extreme gradient boosting (XGB), deep neural networks (DNN), and ensemble models. Moreover, the chapter also proposes using technical indicator (TIs) features as inputs for forecasting BAL prices.

9.1 Introduction

Day-ahead forecasting of BAL prices is challenging because the BAL is designed, as a measure of last resort, to handle stochastic and unforeseen demand and supply imbalances. Despite this, because BAL price forecasts are used as inputs in the decision-making processes of continuous intraday market (CID) trading agents, accurate forecasts are needed.

As the literature review from Chapter 4 highlights, to date, BAL price forecasting research has predominantly focused on the comparison of simple machine learning and statistical models. More research is needed to identify the best BAL forecasting practices.

With the above in mind, in this chapter, the existing literature is extended by evaluating linear-based, tree-based, and neural network-based models. As the BAL has different pricing systems across European countries, the focus is placed solely on predicting Dutch BAL prices.

Chapter 5 highlighted the explanatory power of TI features in day-ahead market (DAM) price forecasting. Consequently, as an additional research goal, the explanatory power of TI features in forecasting BAL prices is explored.

Summarising the two contributions of this chapter, to the best of our knowledge, we are the first to:

- explore using TI features in forecasting BAL prices,
- evaluate and compare an exhaustive range of machine learning models, such as the least absolute shrinkage and selection operator (LASSO), random forest (RF), gradient boosting (GB), extreme gradient boosting (XGB), deep neural networks (DNN), and ensemble models, in forecasting BAL prices.

Outlining the structure of this chapter, the methodology for forecasting BAL prices is described in Section 9.2. In Section 9.3, the results are analysed. Finally, in Section 9.4 the chapter is concluded.

9.2 Balancing Market Price Forecasting Using Machine Learning Models

To forecast BAL prices using machine learning models, various steps and procedures, such as data gathering, feature engineering, data processing, feature selection and modelling, are followed/employed. These steps and procedures are described below.

Table 9.1: Summary statistics for BAL take and feed prices in €/MWh.

	mean			*standard deviation*		
	train	**val**	**test**	**train**	**val**	**test**
TAKE	43.69	42.56	35.51	42.13	37.01	52.72
FEED	40.57	40.39	31.96	40.80	36.42	51.90

9.2.1 Data

BAL-related features, such as the take price, feed price, upward incident reserve, downward incident reserve and regulation state - spanning from 01/01/2016 to 11/12/2020 - are gathered from TenneT [19], an electricity transmission system operator (TSO). Additionally, Dutch and Belgian day-ahead market (DAM) prices are collected, together with load and generation day-ahead forecasts, from the ENTSO-E Transparency Platform [22]. Finally, day-ahead forecasts of temperature, wind speed, solar irradiance and precipitation are gathered from Scholt Energy [109].

Data is split into training, validation, and test data. Data from 01/01/2016 to 01/01/2019 is used for training. Data from 01/01/2019 to 01/01/2020 is used for hyper-parameter tuning and feature selection. Finally, data from 01/01/2020 to 11/12/2020 is used for testing.

The hourly BAL take price (p^{take}) and feed price (p^{feed}) are computed by averaging four quarter-hourly BAL take and feed prices. Summary statistics of these prices are shown in Table 9.1. These summary statistics show that while the mean of BAL prices has decreased from the training set to the test set, the volatility has increased significantly.

9.2.2 Feature Engineering

Exogenous features, including the generation forecast, load forecast, solar irradiance forecast, temperature forecast, wind speed forecast, imbalance volume, purchase volume, upward power, downward power, upward incident reserve, downward incident reserve, incentive component, regulation state, volume, Dutch DAM prices, and Belgian DAM prices are considered. Highly correlated features with a Pearson correlation larger than 0.8 are removed.

Though the BAL is less prone to seasonality effects than other short-term electricity markets, known electricity market characteristics are accounted for by adding day-lagged and hour-lagged features. For instance, to forecast p^{take} of h_{16}^{d+1}, a fourteen-day-lagged period h_{16}^{d-13}, … h_{15}^{d}, h_{16}^{d}, h_{17}^{d} is considered for each feature. Categorical features, such as month of the year, week of the year, day of the week, hour of the day and holiday, are also considered.

Finally, several TI features, such as the exponential moving average (EMA) from Section 5.2.2, momentum (MOM) from Section 5.2.6, Bollinger bands (%B) from Section 5.2.5, are considered. Specifically, EMA(s=4), EMA(s=24), MOM(n=4), MOM(n=24), %B(n=4) and %B(n=24) are calculated separately for p^{take} and p^{feed}.

9.2.3 Data Processing

Min-Max scaling is used for each feature. Min-max scaling bounds each value $a \in A$ between 0 and 1 according to the formula: $a^* = (a - A^{min})/(A^{max} - A^{min})$, where A^{min} and A^{max} are the minimum and maximum values of a set A, and a^* is the scaled value of a.

9.2.4 Feature Selection

Taking all aforementioned features from Section 9.2.2, a feature selection approach is deployed to remove the least important features from the input space. Detailing the feature selection steps, the method firstly calculates mutual information scores between each feature and the target. The calculation of mutual information scores requires nonparametric approaches based on entropy estimates from k-nearest neighbor distances [122]. Overall, if the dependency between two random variables is high, then the mutual information is high. Otherwise, the mutual information is low. The mutual information of two independent random variables, for instance, is zero.

After calculating mutual information scores, the feature selection method selects $k \in [1, 100]$ features with the highest scores. Underlining the selection process, the mean absolute error (MAE) of linear regression is calculated across the validation set for each k. The lowest resulting MAE offers the optimal number of important features k^*.

9.2.5 Model Training and Prediction

The ability of regression models, such as LASSO, random forest (RF), gradient boosting (GB), extreme gradient boosting (XGB), and deep neural networks (DNN), to predict BAL prices is evaluated. Each model has a unique set of hyperparameters to be optimised. By minimising the MAE across the validation set, optimal hyperparameters for each model are identified.

Specifying the hyperparameter sets for each model, LASSO optimises alpha $\in (0,\ 10]$ and tolerance $\in (10^{-5},\ 10^{-1}]$. RF optimises the number of estimators (# estimators) $\in [30,\ 200]$. GB optimises the maximum depth (max depth) $\in [2,\ 14]$. XGB optimises max depth $\in [2,\ 14]$ and # estimators $\in [30,\ 200]$. DNN optimises the number of hidden layers (# layers) $\in [1,\ 4]$, the size of these layers (# neurons) $\in [32,\ 2048]$, and the learning rate $\in [10^{-4},\ 10^{-1}]$. Note that DNN models employ fully connected layers, ReLU activation functions, and the Adam optimiser. Additionally, L2 regularisation and dropout are used to reduce the possibility of overfitting.

Finally note that an ensemble method, which averages all evaluated forecasts following (9.1), is additionally considered.

$$\hat{p}^{\text{ENSEMBLE}} = 1/5(\hat{p}^{\text{LASSO}} + \hat{p}^{\text{RF}} + \hat{p}^{\text{GB}} + \hat{p}^{\text{XGB}} + \hat{p}^{\text{DNN}}) \tag{9.1}$$

9.2.6 Evaluation

MAE is used to evaluate the accuracy of price forecasts. Furthermore, a naive benchmark is considered to facilitate a relative evaluation of forecast accuracies. As a benchmark, a two-day moving average is calculated according to (9.2).

$$\hat{p}^{\text{BENCH}}_{d+1} = 1/2(p_d + p_{d-1}), \tag{9.2}$$

where p is either p^{take} or p^{feed}.

9.3 Results and Discussion

The results of the feature selection step employed in the evaluation of forecasting BAL prices are outlined below. Then, the performance of forecasting models is compared and analysed.

Table 9.2: Selected hyperparameters.

	LASSO alpha	RF # estimators	GB max depth	XGB max depth; # estimators	DNN # neurons; learning rate
TAKE	0.1	105	2	4; 30	[511, 409]; 0.005
FEED	0.1	130	2	2; 30	[1008, 943, 41, 121]; 0.0009

9.3.1 Feature Selection

The feature selection method, which is described in Section 9.2.4, selected 75 and 90 features when forecasting p^{take} and p^{feed} respectively. Significant overlap between these two selections was identified. Elaborating, out of the evaluated categorical features the week of the year, the hour of the day, and the month of the year were all found to be important when forecasting both p^{take} and p^{feed}. Similarly, out of the evaluated TI features, EMA(s=4), EMA(s=24), %B(n=4) and %B(n=24) were found to be important for predicting both BAL prices. The feature selection method discarded MOM(n=4) and MOM(n=24) when forecasting both p^{take} and p^{feed}.

Among exogenous variables, generation forecast, load forecast, temperature forecast, purchase volume, and Dutch and Belgian DAM prices were found to be important when forecasting both BAL prices. The solar irradiance forecast and imbalance volume were, however, found to only be important when forecasting p^{take}. Similarly, the wind forecast and the upward power were found to only be important when forecasting p^{feed}. Seven-day-lagged, two-day-lagged and one-day-lagged exogenous features were often selected; combating electricity market seasonality.

Finally, note that numerous lagged p^{take} features, from a fourteen-day-lagged period, were selected when forecasting p^{take}. Numerous lagged p^{feed} features were selected when forecasting p^{feed}.

9.3.2 Prediction

Table 9.2 presents the results from the hyperparameter optimisation described in Section 9.2.5. The table shows that similar hyperparameters are frequently selected by the evaluated models when predicting BAL prices. LASSO, for instance, chooses an alpha of 0.1 when forecasting both p^{take} and p^{feed}.

Table 9.3: MAE results on the test set.

	LASSO	RF	GB	XGB	DNN	ENSEMBLE	BENCH
TAKE	32.05	37.05	35.16	35.74	31.96	33.63	39.36
FEED	30.96	35.10	33.57	31.35	30.07	31.56	38.69
average	31.51	36.08	34.37	33.55	31.02	32.60	39.03

DNN, however, chooses different hyperparameters when forecasting both p^{take} and p^{feed}. A DNN with two hidden layers is selected when predicting p^{take}, and four hidden layers when predicting p^{feed}.

Table 9.3 shows the forecast accuracies of all the evaluated models. Overall, BENCH is found to yield the highest average MAE of 39.03. Individual machine learning models, LASSO, RF, GB and XGB, yield average MAEs of 31.51, 36.08, 34.37 and 33.55 respectively; outperforming BENCH by roughly 13%. Similar to [47], XGB is found to outperform both GB and RF. Among individual models, DNN is identified as the best model; obtaining the lowest average MAE of 31.02. It thus outperforms BENCH by 19%. ENSEMBLE, which takes the average of all model forecasts, fails to further improve upon DNN forecast accuracies; yielding an average MAE of 32.60.

9.4 Conclusion

In this chapter, balancing market (BAL) prices, namely take and feed prices, were predicted. To predict these prices, feature engineering was utilised, and technical indicator (TI) features were created. Further, to evaluate the importance of engineered features in predicting BAL prices, feature selection was employed. Among TI features, EMA and %B were found to be the most important predictors in forecasting BAL prices.

Using the selected features as inputs, the forecasting accuracies of a range of machine learning models were evaluated. LASSO, random forest (RF), gradient boosting (GB), and extreme gradient boosting (XGB) were found to outperform an evaluated naive benchmark model. Deep neural networks (DNN), meanwhile, were found to outperform all other models; yielding the lowest BAL forecast errors.

Part III

Statistical Arbitrage Trading on Electricity Markets

Chapter 10

Background: Statistical Arbitrage Trading on Electricity Markets

This chapter provides background information about statistical arbitrage trading (SAT): the third part of this thesis. The chapter conducts a literature review, and briefly describes SAT and reinforcement learning.

10.1 Introduction

The transition from non-renewable to renewable sources of energy has significantly increased short-term trading volumes; impacting the day-ahead market (DAM), continuous intraday market (CID) and balancing market (BAL). Although DAM volumes continue to exceed CID volumes, greater renewable production and CID liquidity have significantly upended the planning decisions of traders, boosting their willingness to distribute orders across short-term electricity markets. There are, however, risks inherited in market inefficiencies and illiquidities for traders. Trading strategies increasing market efficiency and liquidity are greatly needed to reduce traders' risks.

Arbitrage trading strategies offer an opportunity to reduce price differences across markets, and increase market liquidity and efficiency while yielding profits for traders [1–7]. Arbitrage opportunities emerge when price differences between markets emerge. Alternatively, they can also arise when prices deviate from their long term means. Arbitrage traders (or intelligent arbitrage agents) try to profit from such price differences without exposing themselves to long-term physical commitments. Note that intelligent arbitrage agents adopt statistical arbitrage trading (SAT), which is an autonomous trading strategy that employs intelligent learning methods.

The third part of this thesis focuses on developing novel SAT strategies for short-term electricity markets. The aim is to optimise the risk-reward ratios of these trading strategies. Before outlining novel strategies, however, this chapter focuses on introducing the required background information.

Outlining the structure of the remainder of this chapter, Section 10.2 conducts a literature review of SAT strategies that have been applied to short-term electricity markets. Subsequently, Section 10.3 describes SAT for short-term electricity markets, and Section 10.4 introduces reinforcement learning. Finally, Section 10.5 concludes the chapter.

10.2 Literature Survey

Most SAT, or virtual bidding, studies that have explored maximising the profits of purely financial traders, have focused on exploiting the price differences between the DAM and BAL. Among these studies, [123] and [124] followed a stochastic optimisation approach to exploit price differences. Similarly, [125] implemented a min-max two-level optimisation model, [126] a data-driven approach, [127] a machine-learning approach and [2] an online-learning algorithm. Each method has been shown to yield positive returns across United States markets.

Analysing studies that considered the CID, [128] explored the profitability of arbitraging between the CID and BAL using a rule-based trading method. As part of the strategy, a short or long position was opened on the CID, before being closed on the BAL. To decide whether to place an initial long or short position, [128] forecast demand. Imbalance volume was additionally predicted. If the forecast of imbalance volume was long, then a short position on the CID was taken and vice versa. By executing this trading rule every 30 minutes with a fixed 2 MWh quantity for the British market, [128] was able to achieve annual revenues of £73407.

Not all evaluated CID trading strategies have centred on arbitrage trading. Some, such as [129–131], have focused on evaluating bidding strategies for asset-backed physical traders, i.e. energy producers. Others, such as [132–134], have focused on bidding strategies for storage device operators. Among these studies, [132,133] have implemented deep reinforcement learning (DRL) methods. Bertrand and Papavasiliou [132] evaluated how REINFORCE, a policy gradient DRL algorithm, could be used to optimise bidding strategies. On the German market, their proposed method surpassed the profitability of a rolling intrinsic policy by 17.8%. Boukas et al. [133] investigated a value function approximation method, an asynchronous distributed version of the fitted Q iteration, to maximise the total return. On the German market, their proposed method improved the profitability of a rolling intrinsic policy by 2.7%. A state selection process, however, was not conducted, and as a result, their study was limited because state engineering and selection processes are vital to function approximators in DRL methods. The optimal number of important states (features) leads to better forecasts and decisions.

10.3 Statistical Arbitrage Trading (SAT)

10.3.1 SAT Across the Intraday and Balancing Markets

Initiator CID traders can place multiple limit orders of varying sizes and prices. Theoretically, initiator traders have an infinite-dimensional action space. Aggressor CID traders, on the other hand, have a low-dimensional action space by trading available orders, i.e. $\{B, S, H\}$, where B buys the best ask order, S sells the best bid order and H holds. In this thesis, agents which behave as aggressor traders are developed.

The aggressor agent starts with a volume of $v_{t_0} = 0$. At any time, this agent can open a long position ($v_t > 0$) with a buy action or a short position ($v_t < 0$) with a sell action on the CID. As shown in Figure 10.1, such an agent, for example, opens a long position and trades continuously $[B, B, H, S, H]$ (left) and

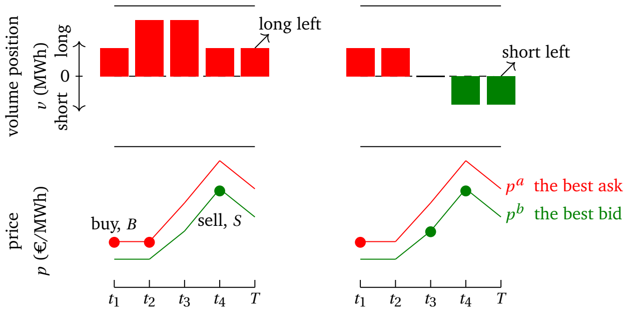


Figure 10.1: Two examples of arbitrage aggressor trading opportunities on the CID for any h. Trading timeline is discretised into 5 for a simple visualisation.

$[B,H,S,S,H]$ (right).[1] Assume q_t^a and q_t^b are 1 MWh. This agent's total bought quantity $\sum q^a$ is then 2 MWh (left) and 1 MWh (right), and total sold quantity $\sum q^b$ is 1 MWh (left) and 2 MWh (right). Agent's total traded quantity $\sum q = min\{\sum q^a, \sum q^b\}$ is 1 MWh, and the remaining quantity, i.e. final position, $\sum q^a - \sum q^b = v_T$ is 1 MWh (left) and -1 MWh (right).

Even though hundreds of transactions are executed during the trading session, some positions are not closed on the CID, either because a counter-party is not found or because closure on the BAL is desired, such as in [128]. In these situations, open positions are cleared on the BAL. BAL prices are announced for each quarter-hour. For hourly contracts, four quarterly balancing prices are averaged separately for short and long positions, p^{take} and p^{feed} respectively. Note that BAL prices display significantly higher volatility ranging from negative prices to high positive prices.

The arbitrage trading strategy aims to maximise profit PnL. On the CID, the agent receives cash $\sum_t p_t^b \times q_t^b$ from selling electricity and pays cash $\sum_t p_t^a \times q_t^a$ for buying electricity. The cash made on the CID is then $C^{cid} = \sum_t p_t^b \times q_t^b - \sum_t p_t^a \times q_t^a$. The remaining quantity $\sum_t q_t^a - \sum_t q_t^b = v_T$ is closed on the BAL. PnL is thus

[1] Remember from Chapter 2.3, at time step $t \in [t_1, T]$, the best ask price p_t^a is the lowest of all available ask prices for the CID. Meanwhile, the best bid price p_t^b is the highest of all available bid prices. Note that their available quantities are q_t^a and q_t^b respectively.

calculated as in (10.1).

$$PnL = \begin{cases} C^{cid} + v_T \times p^{take} - TC, & \text{if } \sum_t q_t^b > \sum_t q_t^a \\ C^{cid} + v_T \times p^{feed} - TC, & \text{if } \sum_t q_t^b < \sum_t q_t^a \\ C^{cid} - TC, & \text{otherwise} \end{cases} \tag{10.1}$$

where TC is the trading cost: $TC = 0.232 \times \sum q$. Note that to factor in the frictions associated with trading, a market operator is assumed to charge traders €0.232 for opening and subsequently closing a position of 1 MWh.

10.3.2 SAT Across Short-Term Electricity Markets

By trading across the DAM, CID and BAL, arbitrage traders aim to maximise profits, PnL, without taking or unwinding long-term physical commitments. Diving into the decisions an arbitrager faces, a trader can open a DAM position v_0 (in MWh) either by buying of $v^{max} > 0$ or short selling $v^{min} < 0$, where v^{min} and v^{max} are pre-defined limits for short and long positions respectively. A trader pays $-v^{max} \times p^{dam}$ for a long position and receives $-v^{min} \times p^{dam}$ for a short position. This represents the cash received or spent on the DAM: C^{dam}.

Having opened a DAM position, an arbitrager can subsequently partially or fully close out the position on the CID. Elaborating, a trader can buy the best ask order (B), sell the best bid order (S), or hold (H) at any time step $t \in [1, T]$ during the CID trading session. Figure 10.2, for example, shows a trader who starts with $v_1 = v^{max}$ - the open position brought forward from the DAM - and continuously trades on the CID, performing $\{H, S, S, S, S, H, B, B, H, \dots\}$. This trader pays $p_t^a \times q_t^a$ for buy decisions and receives $p_t^b \times q_t^b$ from sell decisions. The total cash paid or received by trading on the CID is C^{cid}. Note that the total arbitraged quantity is: $\sum_{t=1}^T q_t = min\{\sum_{t=1}^T q_t^a, \sum_{t=1}^T q_t^b\}$. Meanwhile, the outstanding quantity and final position is $\sum_{t=1}^T q_t^a - \sum_{t=1}^T q_t^b = v_T$.

When the CID trading window closes at time step $t = T$, any outstanding open position the arbitrager holds is automatically settled on the BAL. The cash made on the BAL, C^{bal}, is calculated as $v_T \times p^{take}$ or $v_T \times p^{feed}$.

Summarising the above, the profit PnL of the arbitrage trader is calculated according to (10.2).

$$PnL = C^{dam} + C^{cid} + C^{bal} - TC, \tag{10.2}$$

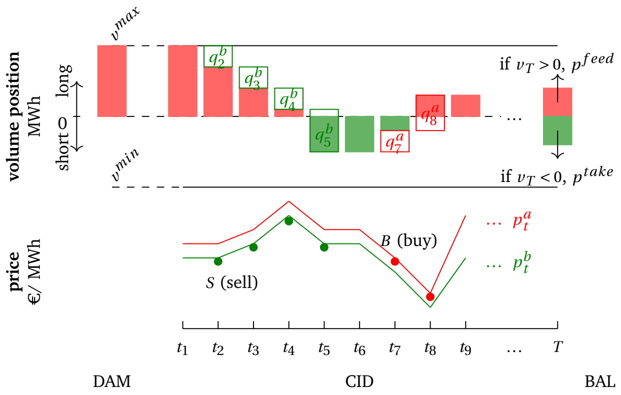


Figure 10.2: An example of arbitrage trading for an example hourly contract h. The continuous trading timeline of CID is split into discrete time steps $t \in [1, T]$.

where

$$C^{dam} = \begin{cases} -v^{max} \times p^{dam}, & \text{if } v_0 > 0 \\ -v^{min} \times p^{dam}, & \text{otherwise} \end{cases}$$

$$C^{cid} = \sum_{t=1}^{T} p_t^b \times q_t^b - \sum_{t=1}^{T} p_t^a \times q_t^a$$

$$C^{bal} = \begin{cases} v_T \times p^{take}, & \text{if } v_T < 0 \\ v_T \times p^{feed}, & \text{if } v_T > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$TC = 0.116 \times (\sum_{t=1}^{T} q_t^a + \sum_{t=1}^{T} q_t^b)$$

Note that TC is the trading cost assumed to be charged by a market operator.

10.4 Reinforcement Learning

RL [135] aims to solve a sequential decision-making problem. In the context of RL, a decision is defined as an *action* and a decision-maker as an *agent*. At time

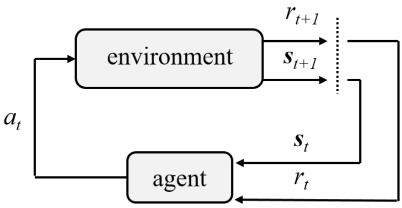


Figure 10.3: Agent's interaction with the environment.

step $t \in [1,\ T]$, an agent executes its action a_t after receiving a state measurement $\mathbf{s}_t$. A *state* is a vector that holds all the necessary information about an environment in order to enable an optimal decision. Based on an agent's action, the agent receives immediate scalar feedback with a *reward* signal $r_{t+1} \in \mathbb{R}$. The interaction described above and Figure 10.3 proceeds iteratively for an *episode* starting from the initial state $\mathbf{s}_1$ to the terminal state $\mathbf{s}_T$.

An RL system is designed to find an optimal behaviour by maximising the total reward over a given number of episodes $e \in [1,\ e^{max}]$. The behaviour, a mapping from states to actions, is defined by the *policy* function as shown in (10.3).

$$\pi(a|\mathbf{s}) = \mathbb{P}[a_t = a \mid \mathbf{s}_t = \mathbf{s}] \tag{10.3}$$

A stochastic policy function outputs a probability for each action. An agent decides whether to exploit - by choosing the action with the highest probability - or explore - by choosing a random action.

The expected total reward, which is accumulated until the terminal state $\mathbf{s}_T$, is defined by a *value* function. While a value function evaluating a state is called the *state-value* function V, a value function evaluating an action from a state is called the *action-value* function Q. Under a policy π, V_π and Q_π are given in (10.4).

$$\begin{aligned} V_\pi(\mathbf{s}) &= \mathbb{E}_\pi[R_t \mid \mathbf{s}_t = \mathbf{s}] \\ Q_\pi(\mathbf{s}, a) &= \mathbb{E}_\pi[R_t \mid \mathbf{s}_t = \mathbf{s}, a_t = a] \end{aligned} \tag{10.4}$$

where $R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots + \gamma^{T-1} r_T$ is the return and $\gamma \in [0,1]$ is the discount factor. As shown in (10.5), the maximum values among all policies yield optimal value functions, V^* and Q^*, and thus an optimal policy π^*.

$$\begin{aligned} V^*_{\pi^*}(\mathbf{s}) &= \max_{\pi} V_{\pi}(\mathbf{s}) \\ Q^*_{\pi^*}(\mathbf{s}, a) &= \max_{\pi} Q_{\pi}(\mathbf{s}, a) \end{aligned} \tag{10.5}$$

10.5 Concluding Remarks

In this chapter, background knowledge required for SAT, the focus of the third part of the thesis, was introduced. Motivations for exploring SAT strategies were detailed. Moreover, a literature review was conducted, and the fundamentals of SAT and reinforcement learning were described. The subsequent chapters in this part of the thesis use the above background information to build novel SAT strategies which optimise the risk-reward ratio.

Chapter 11

Trading Across the Intraday and Balancing Markets

This chapter focuses on statistical arbitrage trading (SAT) opportunities involving the continuous exploitation of price differences arising during an intraday trading period with the option of closing positions on the balancing market (BAL). The chapter aims to maximise the reward-risk ratio of an autonomous trading strategy. To find an optimal trading policy, we propose utilising the asynchronous advantage actor-critic (A3C) algorithm, a deep reinforcement learning method, with function approximators of two-headed shared deep neural networks. A risk-constrained trading strategy is enforced by limiting the maximum allowed position, and conduct state engineering and selection processes. A novel reward function and goal-based exploration, i.e. behaviour cloning, are introduced. Our methodology is evaluated on a case study using the limit order book of the European single intraday coupled market (CID) available for the Dutch market area. The majority of hourly products on the test set return a profit. We expect this chapter to benefit electricity traders, renewable electricity producers and researchers who seek to implement state-of-art intelligent trading strategies.

11.1 Introduction

Arbitrage traders, otherwise referred to as virtual traders, try to profit from price differences without exposing themselves to long term physical commitments. Several studies, such as [1–7], have shown that arbitrage trading can reduce price differences between electricity markets while yielding profits for arbitrage traders. In this chapter, the profitability of statistical arbitrage trading across the CID and BAL, introduced in Section 10.3.1, is investigated.

Trading decisions, namely buying, selling and holding, need to be automated when exploiting statistical arbitrage to optimise the risk-reward ratio. Deep reinforcement learning (DRL), employed by [136, 137] to solve similar decision processes, can be used to automate SAT decisions. Elaborating, various DRL methods, such as value-based, policy-based and actor-critic, can be used because of their consideration of future outcomes when making decisions and their potential to solve sequential decision and control problems.

Actor-critic methods combine the benefits of value-based and policy-based methods and suffer from fewer shortcomings [138,139]. Policy-based methods suffer from high variance during back-propagation. Actor-critic methods reduce this variance. Combining actor-critic methods with generalised advantage estimation further reduces the variance of gradient updates [138]. The most successful algorithm in the space of advantage actor-critic methods has been shown to be the asynchronous advantage actor-critic (A3C) method [139].

A3C has thus far been successfully applied to equity trading on the stock markets [140,141] and wind energy trading on the reserve electricity markets [142]. To the best of our knowledge, the use of A3C, in the context of SAT on the CID, has not been explored yet. Considering the above-mentioned advantages of A3C and similar successful implementations of A3C across other markets, it should be possible to successfully apply A3C to SAT on the CID.

As the review from Chapter 10 highlights, the literature on CID trading is growing; however, papers focusing purely on financial trading are still scarce. Given the potentials for increasing market stability, this chapter focuses on developing a SAT strategy. While a SAT strategy has already been explored by [128] using a trading rule with forecasts, intelligent methods should be further developed and investigated. A DRL method implemented with the A3C algorithm is utilised to maximise the reward-risk ratio of the arbitrage trading strategy.

An agent capable of trading continuously on the CID with the opportunity to close an outstanding position on the BAL is developed. Summarising the three contributions of this chapter, to our knowledge, we are the first to:

- utilise A3C algorithm in developing a risk-constrained arbitrage trading strategy on the CID,

- conduct state engineering and state selection processes for a DRL-based arbitrage trading strategy on the CID,

- propose a novel reward function and behaviour cloning (goal-based exploration) for a DRL-based arbitrage trading strategy on the CID.

Outlining the structure of this chapter, details of our algorithmic trading method are provided in Section 11.2. In Section 11.3, our case study and its results are outlined. Finally, the chapter is concluded in Section 11.4.

11.2 Asynchronous Advantage Actor-Critic Method to Intraday Market Trading

11.2.1 Discrete Time Steps

The continuous trading timeline is discretised, i.e. time step $t \in [t_1, T]$. Typically in the literature, fixed time intervals are used to separate time steps. For example, the time interval Δt between t and $t+1$ is set to 15 minutes in [133], 5 minutes in [143] and 1 minute in [144]. This approach, however, is not sufficiently flexible for our arbitrage strategy. When a profitable ask/bid order is submitted to the CID, the agent should buy/sell this order sooner than its competitors. Additionally, the agent should adjust itself to trading sessions' quiet early hours using fewer time steps and busy final hours using more time steps.

To create flexible time steps, in this thesis the time interval Δt is defined by the order book revision number, i.e. update number for each change in the order book. Most of the updates, however, are not important and change neither the best ask price nor the best bid price. These revisions are dropped and only important updates are kept to improve training performance and reduce training time. Figure 11.1, for example, is plotted after this process.

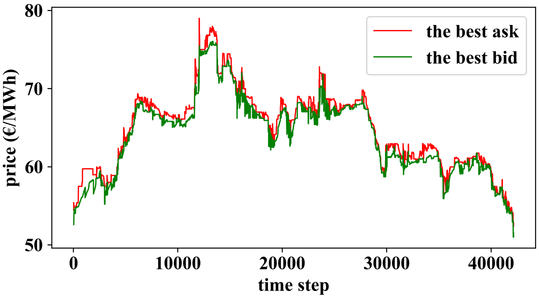


Figure 11.1: The best ask/bid prices for h_9 on 01/09/2020.

11.2.2 Actions

The SAT strategy aims to maximise profit while minimising risk, i.e. the volatility of profit. The challenge of this strategy is rooted in the uncertainty of future CID prices, balancing prices and liquidity. To limit these uncertainties and risk, a risk-constrained arbitrage trading strategy is imposed by constraining bought/-sold quantity at each time step, position and total bought/sold quantity.

An agent can either avoid placing a trade (hold, H) or execute an existing order (buy, B) $q_t^a \leq q^{max}$ or (sell, S) $q_t^b \leq q^{max}$ MW of electricity at time step t. Note that q_t^a is quantity at time step t for the best ask order, q_t^b is quantity at time step t for the best bid order, and q^{max} is the maximum allowed quantity to trade at each time step t. The agent with a maximum allowed position can buy/sell until its long/short position reaches v^{max}/v^{min}. Additionally, the agent with a maximum allowed total trading quantity can buy/sell until its total bought/sold quantity, $\sum q^a / \sum q^b$, reaches q^{high}. To implement the above conditions, the agent's action space is constrained, as shown in (11.1).

$$a_t \in \begin{cases} \{H\}, & \text{if } \sum q^a = q^{high} \text{ and } \sum q^b = q^{high} \\ \{B, H\}, & \text{if } v_t = v^{min} \text{ or } \sum q^b = q^{high} \\ \{S, H\}, & \text{if } v_t = v^{max} \text{ or } \sum q^a = q^{high} \\ \{B, S, H\}, & \text{otherwise} \end{cases} \tag{11.1}$$

where a_t and v_t are the action and volume positions respectively, at time step t.

11.2.3 Rewards

A reward function that motivates the agent to learn to avoid punishment is implemented. For a buy action, the best/worst ask order to buy is the lowest-/highest ask across the entire trading session l^a/h^a and offers a buy reward of $0/-2$. For the remaining ask orders $l^a < p_t^a < h^a$, the agent receives negative buy rewards $r_t^b \in (-2,0)$. The more expensive the asking price the lower the buy reward. Two buy thresholds, the highest bid price across the entire trading session $b^{th} = h^b$ and BAL feed price $b^{th} = p^{feed}$, are considered to separate possible profit and losses within the CID and between the CID and BAL. The agent receives a buy reward of -1, if it buys at $p_t^a = b^{th}$. Above/below the threshold, the agent receives a buy reward that takes on values in $[-2,-1)/(-1,0]$. The immediate buy reward $r_t^b \in [-2,0]$ at time step t is calculated as (11.2).

$$r_t^b = 1/2\ \left(f^b(h^b, p_t^a) + f^b(p^{feed}, p_t^a)\right), \tag{11.2}$$

where p_t^a is the best ask price at time step t and

$$f^b(b^{th}, p_t^a) = \begin{cases} -1 - \left((p_t^a - b^{th})/(h^a - b^{th})\right), & \text{if } p_t^a > b^{th} \\ -\left((p_t^a - l^a)/(b^{th} - l^a)\right), & \text{otherwise} \end{cases}$$

For a sell action, the best/worst bid order to sell is the highest/lowest bid across the entire trading session h^b/l^b and offers a reward of $0/-2$. For the remaining bid orders, the more expensive the bid price the higher the sell reward $r_t^s \in (-2,0)$. Two sell thresholds, the lowest ask price across the entire trading session $s^{th} = l^a$ and BAL take price $s^{th} = p^{take}$, are considered. At time step t, the immediate sell reward $r_t^s \in [-2,0]$ is calculated as (11.3).

$$r_t^s = 1/2\ \left(f^s(l^a, p_t^b) + f^s(p^{take}, p_t^b)\right), \tag{11.3}$$

where p_t^b is the best bid price at time step t and

$$f^s(s^{th}, p_t^b) = \begin{cases} -2 + \left((p_t^b - l^b)/(s^{th} - l^b)\right), & \text{if } p_t^b < s^{th} \\ -1 + \left((p_t^b - s^{th})/(h^b - s^{th})\right), & \text{otherwise} \end{cases}$$

Hold actions are evaluated using the opportunity cost of not buying/selling. If buy and sell decisions may lead to losses, the agent should hold and receive the best possible hold reward of 0. If a buy or sell decision may lead to profit, the agent should not hold and shall receive a hold reward $r_t^h \in [-1, 0)$. The more profitable ask or bid orders are the lower the hold reward is as shown in (11.4).

$$r_t^h = \begin{cases} 0, & \text{if } r_t^b \ \& \ r_t^s < -1 \\ -1 - max\{r_t^b,\ r_t^s\}, & \text{otherwise} \end{cases} \tag{11.4}$$

Additionally, at the end of the trading session for each contract, the end of contract reward $r_t^e = PnL/100$ is provided to motivate pursuing profitable actions. Note that PnL is calculated according to (10.1) in Section 10.3.1. This reward can also be used as a stop-loss threshold SLT. If losing €300 per contract is not desired, for instance, a condition $r_t < -3$ can be incorporated into training as an episode terminator, as further explained in Section 11.2.5.

11.2.4 States

The state vector $\mathbf{s}$ should contain the information required to predict the policy $\pi(\mathbf{s})$ or the state-value under the policy $V_\pi(\mathbf{s})$. Each component of the state vector is a state variable. Together, all variables should be a compact and sufficient representation of the environment. To predict π or V_π accurately, only the most important state variables should be used. A feature engineering process is thus conducted using domain knowledge, and subsequently a feature selection process is conducted using automated feature elimination methods. Below, the types of state variables are first described, subsequently, possible engineered variables for each type of variable are provided and finally our state selection process is elaborated upon.

States are split into internal and external, $\mathbf{s}_t = \mathbf{s}_t^i \cup \mathbf{s}_t^e$. While internal states reflect the agent's portfolio, external states come from the market environment. Example internal states for a trading agent are given below.

- *Agent:* volume position, cash, total traded volume, etc.

External states provide necessary information about the market and the market price drivers. The limit order book, which contains all submitted orders with their type, price, volume, and time information, offers the best insight for continuous market trading. The order book, however, cannot be used as it is for learning. Its inconsistent dimension size across time steps $t \in [t_1, T]$ is incompatible with A3C's function approximators which require consistent input size. A technique adopted in this thesis to achieve consistent dimension size for each time step is to extract some statistical information about the order book. Additionally, to reduce partial observability of the reinforcement learning (RL) environment, lags of the best ask and bid prices, are considered. Example order book features calculated from the order book are given below.

- *Order book:* the best ask price, the best bid price, bid-ask spread, the first quantile of ask prices, the third quantile of bid prices, lags of best ask prices, lags of best bid prices, total ask quantities, total bid quantities, total quantity spread, the first quantile of cumulative ask quantities, the third quantile of cumulative bid quantities, the second quantile of cumulative bid quantities, etc.

Other important price drivers offering executed trade information, such as the highest traded price for a contract (high price), the lowest traded price (low price), the volume-weighted average price of trades (vwap), and the latest traded price (last price), are extracted from the trade book.

- *Trade book:* total traded volume, high price, low price, vwap, last price, etc.

Widely considered seasonal and time-based features are also created.

- *Time-based:* minutes to end of trading session, day, hour, holiday, etc.

Finally, some forecasts assisting agents to make better decisions are considered.

- *Forecasts:* low price forecast, high price forecast, vwap forecast, average mid-price forecast, average bid price forecast, BAL feed price forecast, BAL take price forecast, wind forecast, etc.[1]

Among all the aforementioned engineered external states, the most important features are selected. This can be done with a recursive feature elimination method using, for example, a logistic regression model. Since this method is

[1] Methods implemented to predict CID prices, such as low price, high price, vwap, average mid-price, average bid price and average ask price, are described in Chapter 8. Similarly, methods implemented to predict BAL prices, such as take and feed, are described in Chapter 9. Each chapter uses several methods to predict prices. Forecasts of best performing models are used in this chapter. For instance, the ENSEMBLE models are used to predict CID prices, and deep neural networks are used to predict BAL prices.

based on supervised learning $f(\mathbf{s}^e) = a$, the target $[a_1, a_T]$ needs to be created. Intuitively, this target is the sequence of desired trading actions. To make a profit on the CID, the agent should buy when p_t^a is lower than h^b and sell when p_t^b is higher than l^a. To make a profit between the CID and the BAL, the agent should similarly buy when p_t^a is lower than p^{feed} and sell when p_t^b is higher than p^{take}. Our aforementioned conditions are given in (11.5).

$$a_t = \begin{cases} B, & \text{if } p_t^a < h^b \text{ or } p_t^a < p^{feed} \\ S, & \text{if } p_t^b > l^a \text{ or } p_t^b > p^{take} \\ H, & \text{otherwise} \end{cases} \tag{11.5}$$

Creating our multi-class target $[a_1, a_T]$, the feature selection process starts by removing correlated variables using the Pearson correlation method. The process continues by recursively pruning the least important variables until the performance of the logistic regression model worsens. The remaining variables are considered the most important external state variables.

11.2.5 Asynchronous Advantage Actor-Critic (A3C)

As Figure 11.2 shows, the A3C works with multiple agents which are connected through the global network. Agents interact with their environments and collect different experiences to increase overall exploration. For each thread, the *critic* updates the value and the *actor* updates the policy based on the feedback from the critic [135]. Differentiable function approximators, namely deep neural networks (DNN), are used to estimate the value and the policy, e.g. $V_\pi(\mathbf{s}_t) \approx V_\pi(\mathbf{s}_t;\boldsymbol{\theta}_v)$ and $\pi(a_t|\mathbf{s}_t) \approx \pi(a_t|\mathbf{s}_t;\boldsymbol{\theta}_\pi)$ respectively. Note that $\boldsymbol{\theta}_v$ and $\boldsymbol{\theta}_\pi$ are reserved for the global network parameters, and $\boldsymbol{\theta}_v^{'}$ and $\boldsymbol{\theta}_\pi^{'}$ for the local network parameters.

Figure 11.3 presents a detailed individual agent network. The input-output relationship is shown. Our policy is stochastic, i.e. $\pi(a|\mathbf{s}) = \mathbb{P}[a_t = a \mid \mathbf{s}_t = \mathbf{s}]$. The agent predicts a probability for each action based on the state inputs. If the agent predicts $\{P(B) = 0.6, P(S) = 0.1, P(H) = 0.3\}$, for example, it may with a 0.6 probability *exploit* by choosing its best action B, or with a 0.4 probability *explore* by choosing a random action of either S or H.

In the figure, the DNN - a shared two-headed two-layer neural network - is shown. Note that the network type or the number of layers can vary. Hyperparameter tuning is implemented to select 1) the type of network for π and V_π,

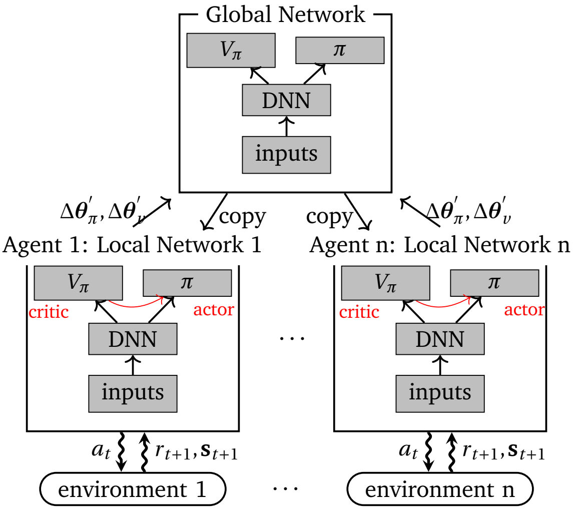


Figure 11.2: The global network keeps the best value and policy of the most successful local network. The global network cannot receive all agents' gradients at once. Agents get in the queue. Deep neural networks (DNN) are used as function approximators. The red arrow represents the critic's feedback.

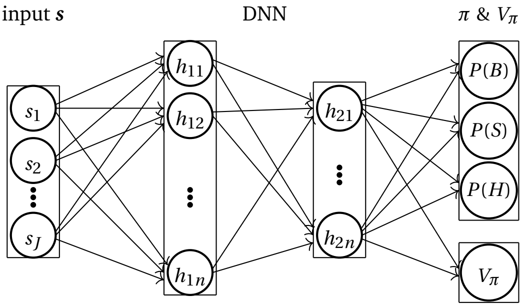


Figure 11.3: Individual network architecture showing the input-output relationship. DNN has two hidden layers. The value V_π has a linear activation function and the policy π has a softmax function converting logits to probabilities.

(i.e. whether to employ a two-headed shared network or separate networks), 2) the number of layers, 3) the number of neurons, 4) the type of hidden layer activation function, 5) the number of agents, 6) the type of optimiser, and 7) the hyper-parameters defined in Algorithm 11.1.

Algorithm 11.1 details the proposed A3C training procedure for each agent. At the start of an episode $e \in [1, e^{max}]$ gradients are reset (line 2), thread parameters are synchronised (line 3), an episode index multiplier is calculated (line 3), a discount rate is computed (line 5), an epsilon factor is calculated (line 6), the time step is initialised (line 4), and the first state is extracted (line 5). Experience is subsequently collected during an episode (lines 6 to 23). The action space is constrained accordingly to Section 11.2.2 (line 7).

Goal-based exploration is implemented to ensure that an agent learns effectively during the initial training stage. With goal-based exploration, i.e. behaviour cloning, the first $b < n$ number of agents are provided with exemplary behaviours during their initial episodes (line 9). One agent follows (11.6) to learn how to exploit arbitrage opportunities between the CID and BAL.

$$a_t = \begin{cases} B, & \text{if } p_t^a < p^{feed} \ \& \ v_t < v^{max} \ \& \sum q^a < q^{high} \\ S, & \text{if } p_t^b > p^{take} \ \& \ v_t > v^{min} \ \& \sum q^b < q^{high} \\ H, & \text{otherwise} \end{cases} \tag{11.6}$$

Another agent follows (11.7) to learn arbitrage opportunities within the CID.

$$a_t = \begin{cases} B, & \text{if } p_t^a < h^b \ \& \ v_t < v^{max} \ \& \sum q^a < q^{high} \\ S, & \text{if } p_t^b > l^a \ \& \ v_t > v^{min} \ \& \sum q^b < q^{high} \\ H, & \text{otherwise} \end{cases} \tag{11.7}$$

Finally, another agent follows (11.8) pursuing high rewards based on a reward threshold $r^{th} \in [-1, 0)$.

$$a_t = \begin{cases} B, & \text{if } r_t^b > r^{th} \ \& \ v_t < v^{max} \ \& \sum q^a < q^{high} \\ S, & \text{if } r_t^s > r^{th} \ \& \ v_t > v^{min} \ \& \sum q^b < q^{high} \\ H, & \text{otherwise} \end{cases} \tag{11.8}$$

If agents show unsuccessful repetitive behaviours, exemplary behaviours using (11.8) with $r^{th} \in [-0.3, 0)$ can be further provided to agents during training to improve their exploration.

To ensure that agents efficiently explore, decayed/decaying epsilon greedy exploration [135] is additionally implemented. With decayed epsilon greedy exploration, agents are forced to explore with probability ϵ and exploit with probability $1-\epsilon$ (lines 15 and 18). The ϵ is decreased slowly starting from the first episode to the last episode to ensure a higher exploration rate in the early episodes and a higher exploitation rate in the final episodes.

Algorithm 11.1 A3C: Pseudocode for each thread

```
Require: hyper-parameters, $\gamma_{start}$, $\gamma_{end}$, $\epsilon_{start}$, $\epsilon_{end}$, $t^{max}$, $e^{max}$, $\beta$, $SLT$, $b$; global parameters, $\boldsymbol{\theta}_\pi, \boldsymbol{\theta}_v$; thread
   parameters, $\boldsymbol{\theta}'_\pi, \boldsymbol{\theta}'_v$; global counter, $e \leftarrow 1$; and $t_{start} \leftarrow 1$
 1: while $e < e^{max}$ do
 2:   reset gradients $d\boldsymbol{\theta}_\pi \leftarrow 0$, $d\boldsymbol{\theta}_v \leftarrow 0$
 3:   synchronise thread parameters $\boldsymbol{\theta}'_\pi = \boldsymbol{\theta}_\pi$, $\boldsymbol{\theta}'_v = \boldsymbol{\theta}_v$
 4:   calculate episode index $e_i = 1 - (e/e^{max})$
 5:   calculate discount $\gamma = (\gamma_{start} - \gamma_{end}) * e_i + \gamma_{end}$
 6:   calculate epsilon $\epsilon = (\epsilon_{start} - \epsilon_{end}) * e_i + \epsilon_{end}$
 7:   $t = t_{start}$
 8:   get state $\mathbf{s}_t$
 9:   while $\mathbf{s}_t \neq \mathbf{s}_T$ and $t - t_{start} < t^{max}$ do
10:     constrain the action space
11:     if $e < b$ then
12:       clone $a_t$ according to (11.6), (11.7) or (11.8)
13:     else
14:       if $\epsilon_{random} < \epsilon$ then
15:         explore: $random\ a_t$ according to $\pi(a_t|\mathbf{s}_t;\boldsymbol{\theta}'_\pi)$
16:       else
17:         exploit: $max\ a_t$ according to $\pi(a_t|\mathbf{s}_t;\boldsymbol{\theta}'_\pi)$
18:       end if
19:     end if
20:     receive reward $r_{t+1}$ and new state $\mathbf{s}_{t+1}$
21:     $t \leftarrow t+1$
22:     $\mathbf{s}_t = \mathbf{s}_T$, if $r_t < SLT$
23:   end while
24:   $R_\pi = \begin{cases} 0, & \text{for terminal } \mathbf{s}_T \\ V_\pi(\mathbf{s}_t, \boldsymbol{\theta}'_v), & \text{for non-terminal } \mathbf{s}_t \end{cases}$
25:   for $i \in \{t-1, \dots t_{start}\}$ do
26:     $R_\pi \leftarrow r_i + \gamma R_\pi$
27:     $A_\pi \leftarrow R_\pi - V_\pi(\mathbf{s}_i,;\boldsymbol{\theta}'_v)$
28:     $d\boldsymbol{\theta}_\pi \leftarrow d\boldsymbol{\theta}_\pi + \beta \ \nabla_{\boldsymbol{\theta}'_\pi} \log\pi(a_i|\mathbf{s}_i;\boldsymbol{\theta}'_\pi)(A_\pi)$
29:     $d\boldsymbol{\theta}_v \leftarrow d\boldsymbol{\theta}_v + \beta\ \partial(A_\pi)^2/\partial\boldsymbol{\theta}'_v$
30:   end for
31:   $e \leftarrow e+1$
32:   update asynchronously $\boldsymbol{\theta}_\pi$ using $d\boldsymbol{\theta}_\pi$, and $\boldsymbol{\theta}_v$ using $d\boldsymbol{\theta}_v$
33: end while
```

Selecting an action a_t, the agent receives its reward and next state (line 21). The agent collects experiences until either the terminal state, $\mathbf{s}_T$ or t^{max} are reached (line 22). The $\mathbf{s}_T$ is defined to be the last state in the training set. Early episode termination is employed when the agent, following its policy, reaches a predefined stop-loss threshold SLT, linked to the end of contract reward r^e. In this way, the number of contracts in each episode varies. As a result, unsuccessful policies are promptly updated for the better, and training time is reduced.

To update the actor, critic and global network, the critic first estimates the value of a state V_π, line 24. Then, for each gradient step the value of the state-action R_π is calculated (line 26). Intuitively, this value is an improvement compared to V_π. The difference is called the *advantage* function A_π [139]. It can be estimated by the temporal difference (TD) error and offers information about the extra reward attainable when following the policy's actions (line 27). If A_π is positive, gradients of the actor and critic are pushed in that direction by the learning rate β. If A_π is negative, gradients are pushed in the opposite direction (lines 26 and 27). Using updated actor and critic gradients, the global network is also updated (line 30). The network weights are updated until a predefined maximum number of episodes e^{max} is reached.

11.3 Case Study

11.3.1 Data

The primary data sources are the limit order book and trade book of the hourly CID contracts available for the Dutch market area. The order and trade books of CID are not publicly available. Data spanning from 01/01/2020 to 10/12/2020 were provided by the energy supplier Scholt Energy [109]. Forecasts, such as wind speed and temperature, were also provided by Scholt Energy. Publicly available data, namely BAL prices of The Netherlands and day-ahead market prices of The Netherlands and Belgium, are collected from the ENTSO-E Transparency Platform [22].

For the period under study, data for 8280 hourly contracts are available. The data is split into training (4148 contracts), validation (1238 contracts) and test sets (1760 contracts). Contracts not used in these sets are excluded because of data quality issues. For example, numerous contracts, especially in January and October, are found to have a significant number of missing observations.

To continuously train and evaluate our agent, a rolling window method illustrated in Figure 11.4 is employed. First, feature elimination is implemented to select the most important features and A3C is shortly trained to tune hyperparameters. These are evaluated on the validation set. Defining the hyper-

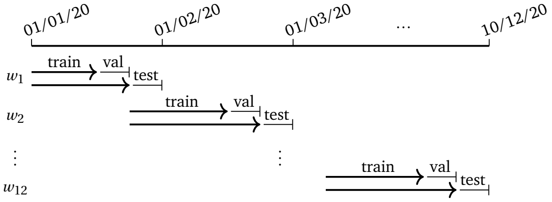


Figure 11.4: Rolling windows for training, validation and test sets.

parameters and features, A3C is trained on combined training and validation set and evaluated on the test set.

11.3.2 Data Preprocessing

Order book and trade book data are processed by creating state variables indexed by important revision numbers. Processing each contract, all contracts are vertically stacked by time separately for training, validation and test sets. Min-max scaling parameters are extracted from the training set for each variable and scale all the data.

11.3.3 External State Selection

Feature selection is used to find the most important external states as mentioned in Section 11.2.4. Initially, collinearity between features is measured using the Pearson correlation. Correlated features with a correlation larger than 0.8 are removed. Subsequently, logistic regression is used to eliminate the least important features. The regularisation penalty λ is set to 0.001 and the mixing parameter between lasso and ridge regressions α is set to 0.9. Class balancing is employed to increase f1 validation scores of buy and sell classes.

Using weights of the logistic regression as feature importance scores, 50% of the least important features are eliminated in the first round and, 10% of the least important features are recursively eliminated in the following rounds. The recursive feature elimination is stop when a stopping criterion is reached; the f1 validation score decreases by a total of 0.1. Using a variable step size ensures that fewer features are eliminated after each round. Features with a high importance are thus less likely to be removed. Features selected by our method are presented in Table 11.1.

Table 11.1: External states.

minutes to end of trading session
spread between p_t^b and p_t^a
spread between $(p_t^a + p_t^b)/2$ and its average forecast
spread between p_t^b and Dutch day-ahead market price
spread between p_t^b and $\bar{p}^b$ forecast
spread between p_t^b and p^{feed} forecast
spread between p_t^b and $1/6\sum_{d-3}^{d-1}\sum_{h-2}^{h} p^{take}$
spreads between p_t^b and its lags $[p_{t-8}^b : p_{t-1}^b]$
the best bid quantity q_t^b
the number of bid orders
the third (upper) quantile of cumulative bid quantities
the best ask price p_t^a
the first (lower) quantile of ask prices
the second quantile (median) of ask prices
the third (upper) quantile of ask prices
spread between p_t^a and $1/6\sum_{d-3}^{d-1}\sum_{h-2}^{h} p^{take}$
the best ask quantity q_t^a
total ask quantities
the number of ask orders

11.3.4 A3C Modelling

In our RL environment, states, consisting of external and internal states, are provided in Tables 11.1 and 11.2. In Table 11.2, q^{high}, PnL^{low} and PnL^{high} are set per contract to 40, −600 and 10000 respectively. These are given to scale agents' cash and volume portfolios, similarly to the min-max scaling applied for the external states. The categorical trade rule variable is added to guide agents using forecast decision thresholds. This variable is updated for every rolling window, based on the best validation set performance, resulting from one of the following forecasts: the average traded price; vwap; high price and low price; and p^{take} and p^{feed}. For instance, the buy category is assigned if p_t^a is lower than a low price forecast, sell is assigned if p_t^b is higher than a high price forecast, and hold is assigned otherwise.

DRL training is implemented with A3C per Section 11.2.5. The probability of buy/sell is assigned to 0 if the volume position reaches $v^{max} = 3/v^{min} = -3$ or the total bought/sold quantity reaches to $q^{high} = 40$. The subtracted probability is added to another action using the categorical trade rule variable. For example, when $P(S) = 0.1$ and $v_t = v^{min}$, $P(S) = 0$ and $P(H) = P(H) + 0.1$ are assigned,

Table 11.2: Internal states.

categorical trade rule based on forecasts
scaled total bought quantity $\sum_t q^a_t / q^{high}$
scaled total sold quantity $\sum_t q^b_t / q^{high}$
scaled position $(v_t + v^{max})/(2 \times v^{max})$
scaled cash $(PnL - PnL^{low})/(PnL^{high} - PnL^{low})$

if the categorical trade rule variable advises to hold. At the end of each episode, a callback function checks whether the total reward and PnL of the training set increased and saves improved models. The last saved model is evaluated.

As mentioned in Section 11.2.5, hyper-parameter tuning is implemented. Specifically, trials are set on the number of agents $n \sim [4, 8]$, discount rate $\gamma_{start} \sim [0.001, 0.9]$, learning rate $\beta \sim [0.00001, 0.1]$, the number of time steps to update the global network $t^{max} \sim [300, 5000]$, optimiser $\sim \{\text{Adam}, \text{SGD}, \text{RMSProp}\}$, and the type of network $\sim \{\text{one-headed}, \text{two-headed}\}$, activation function for hidden layers $\sim \{\text{relu}, \text{tanh}\}$, the number of hidden layers $k \sim [1, 3]$, the number of neurons for the first, second and third hidden layers $n_1 \sim [4, 512]$, $n_2 \sim [4, 256]$ and $n_3 \sim [4, 128]$ respectively. A3C is trained until $e^{max} = 100$ and evaluated on the validation set by PnL for each trial. The process is performed by an automated Python tuning library, `optuna`, which prunes unsuccessful hyper-parameters and makes the tuning process more efficient [145]. The resulting hyper-parameters are $n = 8$, $\gamma_{start} = 0.29$, $t^{max} = 2906$, $\beta = 0.0003$, $k = 2$, $n_1 = 216$, $n_2 = 193$, the activation function of hidden layers is tanh, the optimiser is Adam, and the type of network is a two-headed shared network.

Other hyper-parameters are defined by user preferences: discount rate $\gamma_{end} = 0.9999$, stop loss threshold $SLT = -300$ (i.e. the end of contract reward $r^e < -3$), epsilon $\epsilon_{start} = 0.9$ and $\epsilon_{end} = 0.01$, maximum buy/sell quantity $q^{max} = 1$, maximum allowed position $[v^{min}, v^{max}]$ = [-3, 3], maximum allowed total traded quantity for both buy and sell sides $q^{high} = 40$, and the number of agents who clone behaviour $b = 5$. Note that the first agent clones (11.6), the second agent clones (11.7), and the next three agents clone (11.8) with different reward thresholds for their first episodes. Additionally, during training agents clone (11.8) for the last t^{max} states if the last state $s_t = s_T$ is reached by an early episode ending and is the same for the last four episodes.

A3C is trained for each month using the rolling window method. If a policy manages to profit from most of the training contracts in an episode, this policy is defined as an optimal policy. If an optimal policy is reached, training is stop as early as $e^{max} = 1000$. If not, training continuous until $e^{max} = 20000$. After training, A3C can execute a decision instantly.

11.3.5 Benchmark Strategies

Similarly to A3C, explored benchmarks execute an existing order $q_t^a \leq q^{max} = 1$ and $q_t^b \leq q^{max} = 1$ for buy and sell actions at time step t. Their positions are constrained between $[v^{min}, v^{max}] = [-3, 3]$. And the total bought and sold quantities, $\sum q^a$ and $\sum q^b$, are limited to $q^{high} = 40$ MW. The first benchmark is BENCHVWAP using the volume weighted average price of trades until time step t, i.e. $p_t^{vwap} = \sum_{i=1}^{t}(\text{price}_i \times \text{volume}_i) / \sum_{i=1}^{t}(\text{volume}_i)$. Knowing the past traded prices, BENCHVWAP tries to maximise its revenue by following (11.9).

$$a_t = \begin{cases} B, \text{ if } p_t^a < p_t^{vwap} \\ S, \text{ if } p_t^b > p_t^{vwap} \\ H, \text{ otherwise} \end{cases} \tag{11.9}$$

The remaining benchmarks that are programmed, BENCHPLUS and BENCH, are expected to perform well because all the true values provided to these benchmarks are valuable and not known ex-ante. If these two benchmarks indeed result in success, a simple trading rule using forecasts of the provided values could be a promising trading strategy. If, on the other hand, these two benchmarks result in poorer performance than A3C, an intelligent agent who can consider various outcomes is needed. BENCHPLUS sets its trading rules knowing l^a, h^b, p^{take} and p^{feed} as given in (11.10).

$$a_t = \begin{cases} B, \text{ if } p_t^a < h^b \ \& \begin{cases} p_t^b > l^a \ \& \ p_t^a < p^{feed} \\ \text{or } p_t^b < l^a \end{cases} \\ S, \text{ if } p_t^b > l^a \ \& \begin{cases} p_t^a < h^b \ \& \ p_t^b > p^{take} \\ \text{or } p_t^a > h^b \end{cases} \\ H, \text{ otherwise} \end{cases} \tag{11.10}$$

BENCH sets its trading rules knowing l^a, h^b, and the averages of 30 future time steps of the best ask and bid prices. Its trading rule is given in (11.11).

$$a_t = \begin{cases} B, \text{ if } p_t^a < h^b \; \& \begin{cases} p_t^a < 1/30 \sum_{i=t+1}^{t+30} p_i^b \\ \text{or } p_t^b < l^a \end{cases} \\ S, \text{ if } p_t^b > l^a \; \& \begin{cases} p_t^b > 1/30 \sum_{i=t+1}^{t+30} p_i^a \\ \text{or } p_t^a > h^b \end{cases} \\ H, \text{ otherwise} \end{cases} \tag{11.11}$$

11.3.6 Evaluation

Our A3C trading algorithm is evaluated on the test set using both pure profit and reward-risk metrics. For the profit metric, PnL_c made for each contract $c \in [c_1, C]$, where C is the total number of contracts in the test set, is calculated. A policy is defined as being successful if more than half of the contracts return positive PnL_c and $\sum_{c=1}^{C} PnL_c$ is positive. For the reward-risk metrics, the profit to deviation ratio $PD = \sum_{c=1}^{C} PnL_c / \sigma$, where σ is the standard deviation of the PnL distribution, is calculated. To assess the overall quality of actions, the profit per trade (PT) ratio is also checked. PT is calculated as the cumulative PnL divided by the total traded quantity, $PT = \sum_{c=1}^{C} PnL_c / \sum_{c=1}^{C} \sum_{t=1}^{T} q_t$. Considering our pre-defined trading cost, a successful policy should have a $PT > 0.232$. The higher the PD and PT the better the trading strategy. All the above metrics together inform us about the trade-off between the profit and risk, i.e. the volatility of profits.

11.3.7 Results and Discussion

Figures 11.5 and 11.6 present the cumulative traded quantity and the cumulative PnL for the three evaluated benchmarks and our A3C trading algorithm across the test set. Evaluating the performances of BENCHVWAP and BENCH, relative to our A3C algorithm the two benchmarks place three times as many trades. Despite a high traded quantity, however, they generate lower revenues. Given high trade quantities but low revenues, one may expect trading costs, i.e. €0.232 per 1 MW, to be responsible for the two benchmarks' poor performances. An inability to manage balancing risk is, however, also responsible. BENCHVWAP and BENCH perform reasonably well across the CID. Their performance, however, is broadly negatively impacted when the requirements of

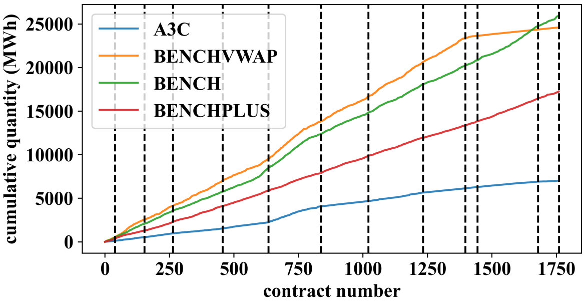


Figure 11.5: $\sum q$ across test set contracts. The dashed black lines represent monthly rolling window splits.

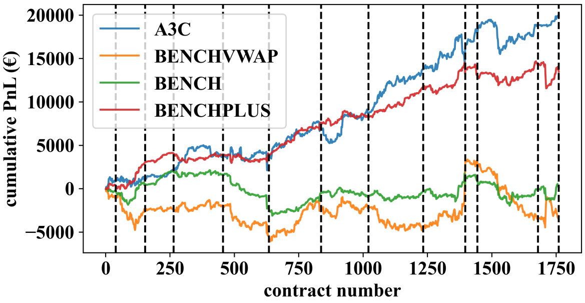


Figure 11.6: $\sum PnL$ on the test set contracts. The dashed black lines represent monthly rolling window splits.

re-balancing are factored-in. BENCHVWAP and BENCH fail to profit from arbitrage opportunities between the CID and the BAL. Moreover, they fail to efficiently plan for necessary position closures. This restricts the final attainable PnL of the strategies.

Evaluating the performance of the third benchmark, BENCHPLUS places significantly fewer trades than BENCHVWAP or BENCH. Despite placing fewer trades, BENCHPLUS generates significantly higher profits. Knowledge of BAL prices allows BENCHPLUS to outperform BENCHVWAP and BENCH. BENCHPLUS, however, cannot surpass A3C's $\sum PnL$. The granularity of time-steps may partially explain the lower $\sum PnL$. Across our case study, an agent is required

Table 11.3: Statistical description of traded quantity (in MW).

	A3C	BENCHVWAP	BENCH	BENCHPLUS
$\sum q$	7017.20	24614.60	25943.20	17245.80
mean	3.99	13.99	14.74	9.80

Table 11.4: Statistical description of PnL (in €).

	A3C	BENCHVWAP	BENCH	BENCHPLUS
$\sum$ ***PnL***	19927.22	-1631.73	2578.04	14862.57
mean	11.32	-0.92	1.46	8.44
std	139.54	152.26	96.50	92.91
PnL > 0	56.08%	50.23%	59.77%	64.21%
PnL < 0	38.64%	49.77%	40.22%	35.80%
PD	142.80	-10.72	26.72	159.96
PT	2.84	-0.07	0.10	0.86

to make thousands of decisions per contract (see Figure 11.1). Implementing an efficient rule-based trading strategy is challenging and complex when an agent is required to make thousands of decisions. An intelligent agent who can autonomously learn to consider and plan for future outcomes is needed. A3C manages this, steadily increasing its revenue from 0 to €19927.22 and surpassing all trading rule-based benchmarks.

Expanding the analysis, Tables 11.3 and 11.4 present additional metrics for evaluating the performances of the benchmark and A3C trading agents. Table 11.3 highlights that A3C on average trades 3.99 MW. This is marginally higher than the maximum position size (3 MW) that the agent is permitted to have open at any moment. The relatively low average traded quantity of the A3C agent can be explained by our choice of hyperparameters. Our A3C agent is set up to learn a risk-constrained trading strategy.

Analysing the percentage of positive $PnLs$, Table 11.4 shows that A3C and the benchmarks successfully return a positive $PnL > 0$ at least 50% of the test contracts. BENCHVWAP, with knowledge of only past prices, is found to be the worst-performing agent. It fails to beat trading rules with knowledge of future prices as would be expected.

Analysing risk-reward ratios, A3C has a high PD and PT indicating a favourable reward-to-risk ratio. The PD of BENCHPLUS is higher, however. This result is expected since BENCHPLUS uses true values in its decision-making process. In practice, were forecasts used instead of actual prices in BENCHPLUS, the risk (standard deviation of profits) would be higher and the PD would be lower due to the direct impacts of forecast errors. The PT of A3C is the highest out of all evaluated trading strategies. Based on its PnL, PD and PT performances, A3C is considered to be the best trading agent.

11.4 Conclusion

This chapter focused on the SAT strategy which aims to profit by trading continuously on the intraday market with the opportunity to close the remaining position on the balancing market. The objective of the chapter was to maximise profit while minimising any associated risk. To hedge against the risk, a constrained trading strategy was implemented with a maximum allowed position for both short and long trades, and a maximum total allowed trading quantity for total bought and sold quantities.

This chapter demonstrated that asynchronous advantage actor-critic (A3C), a deep reinforcement learning (DRL) method, can be successfully utilised to develop an autonomous trading agent capable of exploiting arbitrage opportunities. In this chapter, states were selected from a large number of engineered features using a recursive feature elimination method. Exploration was enhanced by using decayed epsilon greedy exploration and behaviour cloning, i.e. goal-based exploration. A3C function approximators were designed with two-headed shared deep neural networks.

Our methodology was evaluated on a case study using the limit order book of the single intraday coupled market (CID) available for the Dutch market area. A3C was trained on 4148 hourly contracts and tested on 1760 contracts using a rolling window method. On test contracts, A3C traded a total of 7017.20 MW of electricity and generated profits of €19927.22 with an average profit of €2.84 per trade. The majority of contracts, 56.08%, returned a profit.

Chapter 12

Trading Across the Day-Ahead, Intraday and Balancing Markets

In this chapter, statistical arbitrage trading strategies that exploit price differences arising across short-term electricity markets, namely day-ahead (DAM), continuous intraday (CID) and balancing (BAL) markets, are developed and evaluated. To open initial DAM positions, a rule-based trading policy using DAM and CID price forecasts is proposed. Using the DAM forecasts from Chapter 7 and CID forecasts from Chapter 8, the direction of price movements is correctly predicted the majority of the time. To manage open DAM positions while optimising the risk-reward ratio, deep reinforcement learning agents are employed utilising the advantage actor-critic algorithm (A2C). Note that BAL price forecasts from Chapter 9 are used as a state variable by A2C. Evaluated across Dutch short-term markets, A2C yields profits surpassing those obtained using A3C and other benchmarks.

12.1 Introduction

Arbitrage trading offers financial incentives to traders and benefits to markets, such as increasing market liquidity and efficiency [1, 2]. In this chapter, we evaluate the profitability of statistical arbitrage trading (SAT) strategies across short-term electricity markets, introduced in Section 10.3.2. SAT strategies employing deep neural networks, technical indicators, data augmentation and the synchronous advantage actor-critic (A2C) algorithm are developed and analysed for short-term electricity markets, namely the day-ahead market (DAM), continuous intraday market (CID) and real-time balancing market (BAL). Our autonomous agents first open a position on the auction-based DAM, before closing this position on the continuous-based CID or the BAL.

We employ deep reinforcement learning (DRL) to optimise the decision-making process of placing trades across the CID. DRL is selected because of its numerous documented successes, such as [136, 137], in solving sequential decision problems. Among DRL algorithms, advantage actor-critic algorithms have proven particularly adept at solving sequential decision problems because of their ability to reduce the volatility of gradient updates [138, 139]. Yang et al. [146] and [147] employed A2C in optimising trading decisions across equity markets. Meanwhile, [148] applied A2C to optimise the decision making of a retail electricity trader. To the best of our knowledge, we are the first to employ A2C in the context of SAT across the CID.

While SAT studies have thus far focused on the DAM-BAL and CID-BAL, as shown in the literature review in Chapter 10, to the best of our knowledge no study has developed or analysed a trading strategy capable of simultaneously exploiting arbitrage opportunities arising across all short-term electricity markets yet. Given SAT's benefits to markets and traders, we investigate the profitability of SAT across the DAM-CID-BAL trading floors. We propose employing a rule-based trading agent, which uses forecasts of DAM and CID prices to open a position on the DAM. These prices are already predicted in Chapters 7 and 8 using data augmentation methods, technical indicator features and novel engineered features.

For the CID and BAL, we propose developing an agent trained utilising A2C. The asynchronous version of A2C, A3C, has already been employed in Chapter 11. Using A2C, however, is more cost-effective. Note that to improve training stability, unlike Chapter 11, in this chapter we use gradient clippings, different reward thresholds, additional behaviour cloning methods, more goal-based explorations, more flexible constraints and no early episode terminations.

Summarising the contributions of the chapter, to the best of our knowledge, we are the first to:

- investigate the profitability of SAT across the DAM, CID, and BAL for a purely financial trader,
- evaluate a rule-based trading agent that uses DAM and CID price forecasts for the DAM,
- utilise A2C to develop a risk-constrained arbitrage trading algorithm for the CID and BAL.

Outlining the structure of this chapter, in Section 12.2, our rule-based trading method for the DAM is outlined. In Section 12.3, our advantage actor-critic trading methods for the CID are introduced. The case study for the CID is presented in Section 12.4. The results are shown in Section 12.5. Finally, in Section 12.6 the chapter is concluded.

12.2 Rule-Based Approach to Day-Ahead Market Trading

To open a position on the DAM, we implement a rule-based trading agent using forecasts of day-ahead market prices p^{dam}, and continuous intraday market prices, more specifically the volume-weighted average price of trades p^{vwap}. If the forecast of p^{dam} is lower than the forecast of p^{vwap}, we open a long position, i.e buy v^{max} MWh. Otherwise, we open a short position, i.e. sell v^{min} MWh. Note that v^{max} is the maximum allowed long position, and v^{min} is the minimum allowed position for risk-constrained SAT strategies. Note also that the methods employed to predict p^{dam} and p^{vwap} are described in Chapters 7 and 8 respectively.[1]

[1] Chapters 7 and 8 evaluate several models to predict p^{dam} and p^{vwap}. Forecasts of best-performing models are used by the rule-based trading agent. Specifically, the 2NN model utilising the AE+VAE+WGAN-GP+TI method is used to forecast p^{dam}, and the ENSEMBLE model is used to predict p^{vwap}.

12.3 Advantage Actor-Critic Approach to Intraday Market Trading

To manage open DAM positions, we utilise a deep reinforcement learning trading agent trained using the advantage actor-critic algorithm (A2C). This section details this procedure.

12.3.1 Actions

Actions a_t are defined over a discrete set. The action set is constrained to minimise risk. Firstly, the volume position v_t is restricted by the maximum allowed positions for short and long sides, v^{min} and v^{max} respectively. Following this strategy, if B is chosen, the agent buys $q_t^a \leq v^{max} - v_t$ MWh of electricity at time step $t \in [1, T]$. If S is chosen, the agent sells $q_t^b \leq -v^{min} + v_t$ MWh of electricity at time step t. Additionally, similar to Chapter 11, the total bought quantity until time step t, i.e. $\sum_{i=1}^{t} q_i^a$, and the total sold quantity until time step t, i.e. $\sum_{i=1}^{t} q_i^b$, are restricted by the maximum allowed total quantity q^{high}.

12.3.2 Rewards

Similarly to Chapter 11, we employ negative reward functions to ensure the stability and efficiency of the learning algorithm. Note, however, that different reward thresholds are used. Additionally, we avoid using the end of contract reward as in Chapter 11. Only three different reward functions are needed to span the action space of our agent.

The immediate buy reward function is calculated by measuring the distance between a buy action - buying at p_t^a - and both the best and worst possible buy actions across a trading session - buying at the lowest and highest prices $p^{a_{low}}/p^{a_{high}}$. The buy reward function is bounded: $r_t^B \in [-2, 0]$. Equation (12.1) formalises the buy reward function employed:

$$r_t^B = 1/2\left(f^B(p^{dam}, p_t^a) + f^B(p^{b_{high}}, p_t^a)\right), \tag{12.1}$$

where p_t^a is the best bid price at time step t and

$$f^B(\tau^B, p_t^a) = \begin{cases} -1 - \left((p_t^a - \tau^B)/(p^{a_{high}} - \tau^B)\right), & \text{if } p_t^a > \tau^B \\ -\left((p_t^a - p^{a_{low}})/(\tau^B - p^{a_{low}})\right), & \text{otherwise} \end{cases}$$

and τ^B is the buy threshold separating gains and losses. A reward of $r_t^B = -1$, attained at τ^B, marks where profit can at best equal 0 over a trading session. When an agent purchases electricity for less than τ^B positive profits are attainable and $r_t^B > -1$. Equally, when the agent purchases electricity for more than τ^B a loss is guaranteed and $r_t^B < -1$. To separate the gains and losses of trading between the DAM and CID, we firstly consider $\tau^B = p^{dam}$. To separate the gains and losses of trading within the CID, we secondly consider $\tau^B = p^{b_{high}}$.

Similarly to the above, the immediate sell reward function is calculated by measuring the distance between a sell action - selling at p_t^b - and both the best and worst possible sell actions across a trading session - selling at the most expensive and least expensive prices $p^{b_{high}}/p^{b_{low}}$. The sell reward function is intrinsically bounded: $r_t^S \in [-2, 0]$. The more expensive the bid price at which the agent sells electricity the higher the sell reward. Equation (12.2) formalises the sell reward function employed:

$$r_t^S = 1/2\ \left(f^S(p^{dam}, p_t^b) + f^S(p^{a_{low}}, p_t^b)\right), \tag{12.2}$$

where p_t^b is the best bid price at time step t and

$$f^S(\tau^S, p_t^b) = \begin{cases} -2 + \left((p_t^b - p^{b_{low}})/(\tau^S - p^{b_{low}})\right), & \text{if } p_t^b < \tau^S \\ -1 + \left((p_t^b - \tau^S)/(p^{b_{high}} - \tau^S)\right), & \text{otherwise} \end{cases}$$

Sell thresholds of $\tau^S = p^{dam}$ and $\tau^S = p^{a_{low}}$ are considered to separate the gains and losses of trading between the DAM and CID, and within the CID.

Finally, the hold reward function is determined by quantifying the opportunity cost of a buy/sell action. Equation (12.3) formalises the hold reward function.

$$r_t^H = \begin{cases} 0, & \text{if } r_t^B \ \&\ r_t^S < -1 \\ -1 - max\{r_t^B,\ r_t^S\}, & \text{otherwise} \end{cases} \tag{12.3}$$

Observe that the agent receives a hold reward of $r_t^H = 0$ when buy and sell actions lead to losses. The agent receives a hold reward $r_t^H \in [-1, 0)$ if either a buy action or a sell action is profitable. The more profitable the buy or sell action is, the lower the hold reward is.

12.3.3 States

The features presented in Tables 11.1 and 11.2 are used to encode the state space. The categorical trade rule feature, in contrast with Chapter 11, uses

p^{dam} in place of CID and BAL price forecasts. Formally, when $p_t^a < p^{dam}$ the feature takes the value buy. When $p_t^b > p^{dam}$ the feature takes the value sell. Otherwise, the feature takes the value hold.

12.3.4 Advantage Actor-Critic Algorithms

Algorithm 12.1 Pseudocode for each AC worker

```
Require: hyper-parameters, $\gamma_{start}$, $\gamma_{end}$, $\epsilon_{start}$, $\epsilon_{end}$, $t^{max}$, $e^{max}$,$\beta$; global parameters, $\boldsymbol{\theta}_\pi$,$\boldsymbol{\theta}_v$; thread parameters, $\boldsymbol{\theta}_\pi^{'}$,$\boldsymbol{\theta}_v^{'}$; global counter, $e \leftarrow 1$; and $t_1 \leftarrow 1$
Ensure: $d\boldsymbol{\theta}_\pi$, $d\boldsymbol{\theta}_v$
while $e < e^{max}$ do
    reset gradients $d\boldsymbol{\theta}_\pi \leftarrow 0$, $d\boldsymbol{\theta}_v \leftarrow 0$ and synchronise thread parameters $\boldsymbol{\theta}_\pi^{'} = \boldsymbol{\theta}_\pi$, $\boldsymbol{\theta}_v^{'} = \boldsymbol{\theta}_v$
    calculate episode index $e_i = 1 - (e/e^{max})$, discount $\gamma = (\gamma_{start} - \gamma_{end}) * e_i + \gamma_{end}$, and epsilon $\epsilon = (\epsilon_{start} - \epsilon_{end}) * e_i + \epsilon_{end}$
    $t = t_1$
    get state $\mathbf{s}_t$
    while $\mathbf{s}_t \neq \mathbf{s}_T$ and $t - t_1 < t^{max}$ do
        constrain the action space
        if $e = 1$ then
            clone behaviour $a_t$
        else
            if $\epsilon_{random} < \epsilon$ then
                if $t_{random} < t < t_{random} + t^{max}$ then
                    clone behaviour $a_t$
                else
                    explore: $random\ a_t$ according to $\pi(a_t|\mathbf{s}_t;\boldsymbol{\theta}_\pi^{'})$
                end if
            else
                exploit: $max\ a_t$ according to $\pi(a_t|\mathbf{s}_t;\boldsymbol{\theta}_\pi^{'})$
            end if
        end if
        receive reward $r_{t+1}$ and new state $\mathbf{s}_{t+1}$
        $t \leftarrow t+1$
    end while
    $R_\pi = \begin{cases} 0, & \text{for terminal}\,\mathbf{s}_T \\ V_\pi(\mathbf{s}_t,\boldsymbol{\theta}_v^{'}), & \text{for non-terminal}\,\mathbf{s}_t \end{cases}$
    for $i \in \{t-1, \ldots t_{start}\}$ do
        $d\boldsymbol{\theta}_\pi \leftarrow d\boldsymbol{\theta}_\pi + \beta\ \nabla_{\boldsymbol{\theta}_\pi^{'}} \log\pi(a_i\,|\,\mathbf{s}_i\,;\boldsymbol{\theta}_\pi^{'})(r_i + \gamma R_\pi - V_\pi(\mathbf{s}_i,;\boldsymbol{\theta}_v^{'}))$
        $d\boldsymbol{\theta}_v \leftarrow d\boldsymbol{\theta}_v + \beta\ \partial(A_\pi)^2/\partial\boldsymbol{\theta}_v^{'}$
    end for
    $e \leftarrow e+1$
    update $\boldsymbol{\theta}_\pi$ using $d\boldsymbol{\theta}_\pi$, and $\boldsymbol{\theta}_v$ using $d\boldsymbol{\theta}_v$
end while
```

In the context of actor-critic (AC) algorithms, a DRL agent is an AC worker, the value is updated by the critic and the policy is updated by the actor using the critic's feedback [135]. The AC algorithm uses a single global network but multiple AC workers, i.e. $w \in [1, W]$, where W is the number of local networks. AC workers update the global network asynchronously in A3C, whereas synchronously in A2C. Each AC worker collects different experiences by independently interacting with the environment. Using multiple workers results in a greater exploration of the state space.

Each actor is stochastic and approximated by deep neural networks that use softmax functions in the output layer. Hyper-parameters, such as the number of hidden layers L and the number of neurons n_l for each hidden layer $l \in [1, L]$, are optimised.

Algorithm 12.1 formalises the update procedure of each AC worker. Describing the algorithm, firstly, the AC worker resets gradients and synchronises thread parameters (line 2). The worker then calculates the episode index, ascending discount rate and descending epsilon (line 3). Subsequently, it receives its first state (line 5). Next, the worker interacts with the environment and collects experiences (lines 6 to 23). The action space is constrained following Section 12.3.1 (line 7). Each worker clones behaviours during the first episode. It also clones behaviours t^{max}/T of the time over the rest of the episodes (lines 9 and 13). Note that to spur exploration, each worker clones different behaviours. Elaborating, the first worker, w_1, clones (12.4), to learn to effectively arbitrage between the DAM and the CID.

$$a_t = \begin{cases} B, & \text{if } p_t^a < p^{dam} \ \& \ v_t < v^{max} \ \& \ \sum_{i=1}^t q_i^a < q^{high} \\ S, & \text{if } p_t^b > p^{dam} \ \& \ v_t > v^{min} \ \& \ \sum_{i=1}^t q_i^b < q^{high} \\ H, & \text{otherwise} \end{cases} \tag{12.4}$$

The second worker, w_2, meanwhile clones (11.7) to learn to arbitrage well within the CID. The third worker, w_3, clones (11.6) to arbitrage well between the CID and the BAL. Finally, the rest of the workers, $w \in [4, W]$, clone (11.8) with different reward thresholds $\in [-1, 0)$ to learn to reach higher rewards.

Following the decayed epsilon greedy exploration method, the worker explores $\epsilon - (t^{max}/T)$ of the time and exploits $1 - \epsilon$ of the time (lines 15 and 18). The worker collects experiences until the number of time steps to update the global network t^{max} or the last state in the training set $\mathbf{s}_T$ is reached. Using collected experiences, the critic estimates the value of a state. The advantage function [139] A_π is calculated by the difference between the estimated value of this state and the value of the state-action. The actor and critic are updated using A_π (lines 26 and 27). Consequently, the global network is updated asynchronously for A3C and synchronously for A2C (line 30). Note that the smooth L1 loss and gradient clipping are implemented to avoid exploding gradients.

Updating process continues until a predefined maximum number of episodes e^{max} is reached.

12.4 Case Study

In this section, the data gathering and processing steps employed in developing agents for CID trading are outlined. Further, our actor-critic hyperparameters are specified. And finally, the evaluation procedure is described.

12.4.1 Data and Data Processing

Data spanning 2020 - from 01/01/2020 to 11/12/2020 - are collected from Scholt Energy [109]. In total the order-books of 8280 contracts are obtained. 4148 contracts are used for training and 1760 contracts for testing. The remaining contracts are excluded from the study for failing data quality tests. Note that because optimised parameters from Chapter 11 are used, no contract is ascribed to a validation set. A rolling window is used to continuously train and evaluate AC workers every month. Finally, min-max scaling is utilised to scale states.

12.4.2 Advantage Actor-Critic Algorithms

The reinforcement learning environment is configured following Section 12.3. Using hyper-parameters from Chapter 11, hyperparameters are set: $W = 8$, $\epsilon^{start} = 0.9$, $\epsilon^{end} = 0.01$, $\gamma^{start} = 0.29$, $\gamma^{end} = 0.9999$, $t^{max} = 2906$, $\beta = 0.003$, $L = 2$, $n_1 = 216$ and $n_2 = 193$. The chosen neural network architecture is a two-headed shared network. A tanh activation function is used in the hidden layers. Adam is selected as an optimiser.

To scale our states, PnL^{low} and PnL^{high} are set to -5000 and 10000 respectively. To constrain our action space, a more flexible approach than Chapter 11 is used. While Chapter 11 sets $q^{high} = 40$, $v^{max} = 3$, and $v^{min} = -3$, we set $q^{high} = 50$, $v^{max} = 10$, and $v^{min} = -10$. While Chapter 11 additionally constrains the maximum allowed buying/selling quantity at each time step t to 1 MW, such a fixed constraint is avoided in this chapter.

For every month of the training set, AC workers are trained until $e^{max} = 100$. Eight workers are thus trained across 800 episodes. At the end of each episode e, the performance, i.e. the total reward accumulated across the training window, is calculated. Model weights are saved whenever performance improves. The last saved model is used on the test set.

Training times vary across the rolling windows due to varying monthly contract counts. For a single month, 1-2 days are required to train AC agents using a GeForce GTX 1080. Once trained, the trading agent can however execute a decision immediately.

12.4.3 Evaluation

Both profit (PnL and PT) and risk-reward (PD) metrics are used to evaluate agents' test performances. Note that PnL is calculated according to (10.2) in Section 10.3.2. The PT is calculated by dividing the PnL by the total traded quantity. Formally, $PT = \sum_{n=1}^{N} PnL_n / \sum_{n=1}^{N} \sum_{t=1}^{T} q_t$, where N is the total number of contracts in the test set. The PD, measuring the profit per unit of risk, is calculated as $\sum_{n=1}^{N} PnL_n / \sigma$, where σ is the standard deviation of the PnL. The higher PnL, PT, and PD the better the trading algorithm.

To contextualise the performances of our intelligent agents, two rule-based benchmarks are evaluated as well. The first, intended to gauge the minimal attainable profit from arbitrage trading on short-term markets, is the HOLD benchmark. The HOLD closes out all open DAM positions on the BAL; no trade is executed on the CID. The second benchmark (PRE-BA) follows the rules specified in (12.5):

$$a_t = \begin{cases} B, \text{if } p_t^a < 1/30 \times \sum_{i=t-30}^{t} p_i^a \ \& \ v_t < v^{max} \ \& \ \sum q^a < q^{high} \\ S, \text{if } p_t^b > 1/30 \times \sum_{i=t-30}^{t} p_i^b \ \& \ v_t > v^{min} \ \& \ \sum q^b < q^{high} \\ H, \text{otherwise} \end{cases} \tag{12.5}$$

PRE-BA uses the previous best bid and best ask prices to place trades. It is developed to highlight the risk associated with CID trading.

12.5 Results and Discussion

12.5.1 Opening Positions on the Day-Ahead Market

DAM price forecasts, $\hat{p}^{dam}$, are utilised along with vwap forecasts, $\hat{p}^{vwap}$, by our ruled-based DAM trading agent. Defining the agent's strategy, a long position is opened on the Dutch DAM, $v^{max} = 10$, if $\hat{p}^{dam} < \hat{p}^{vwap}$. Otherwise, a

Table 12.1: Confusion matrix across 2020 for the ruled-based DAM agent.

		Targets	
		True	**False**
Predictions	**True**	1781	665
	False	737	2203

Table 12.2: Traded quantity, profit and risk results on the test set.

	quantity (in MWh)		***PnL* (in €)**				
	sum	**mean**	**sum**	**mean**	***PnL*>0**	***PD***	***PT***
A2C	33805.10	19.21	97853.69	55.60	61%	190.66	2.90
A3C	29571.70	16.80	89248.52	50.71	62%	174.06	3.02
PRE-BA	52070.50	29.59	1586.07	0.90	51%	3.21	0.03
HOLD	17600.00	10.00	1395.20	0.79	52%	2.83	0.08

short position is opened: $v^{min} = -10$. Dutch and Belgian DAM price forecasts are averaged to obtain $\hat{p}^{dam}$. Averaging DAM forecasts of neighbouring countries improves the performance of the rule-based trading agent. We postulate that this occurs because the CID market provides quotes of pan-European electricity prices. Taking an average of DAM prices allows us to identify profitable opportunities arising between neighbouring market areas.

Table 12.1 presents a confusion matrix summarising the performance results of the DAM trading strategy. Following the rule-based trading strategy, the direction of price movements is correctly predicted 73.97% of the time. As opened DAM positions are used as the initial CID positions, this accuracy is important for the performance of the upcoming CID trading agent and the final profit. The lower the accuracy the harder for the CID agent to yield a profit. For example, assuming no trade is made on the CID and all opened DAM positions are closed out at the BAL, an agent with an accuracy of 0% would yield a loss of €406419.42, while an agent with an accuracy of 100% would return a profit of €231842.07. Our ruled-based DAM trading agent with an accuracy of 73.97% would return a profit of €678.07. Note that this agent will be evaluated as a benchmark, called HOLD, in the next section.

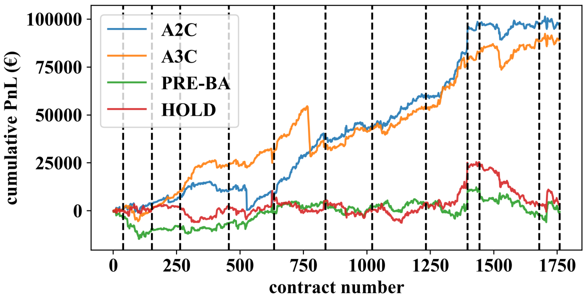


Figure 12.1: Cumulative PnL across test contracts. Monthly rolling windows are shown with dashed black lines.

12.5.2 Closing Positions on the Continuous Intraday Market or the Balancing Market

Analysing the summary results presented in Table 12.2, HOLD is observed to yield the lowest PnL and PD, and the second-lowest PT. Only 52% of traded test contracts return positive profits following HOLD. Evaluated across test set contracts, Figure 12.1 shows the fluctuations in HOLD revenues. The cumulative PnL is almost flat at around 0. By not placing trades on the CID, HOLD exposes the importance of the CID in arbitrage trading.

Analysing the overall performance of PRE-BA, the second benchmark spurs the highest execution of trades. 52070.5 MWh of electricity is traded by PRE-BA. Despite this, PRE-BA yields the second-lowest PnL and PD, and the lowest PT. Evaluated across test set contracts, Figure 12.1 exposes persistently low revenues. Similarly to HOLD, the cumulative PnL fluctuates around 0. PRE-BA highlights the risks of trading too frequently on the CID.

Turning to AC strategies, from the results in Table 12.2 it can be observed that 62% of traded contracts yield positive profits following A3C: 10% more contracts than HOLD and 11% more than PRE-BA. Figure 12.1 further shows that A3C manages to steadily increase the cumulative PnL. A3C generates 6319% more profit per contract than HOLD and 5535% more than PRE-BA. Note, however, that extra profit is not generated by accepting disproportionately more risk. This is highlighted by A3C's higher PD.

Comparing the results relative to Chapter 11, A3C also yields greater returns and a higher PD than the A3C implemented in Chapter 11. Across the same test set, the A3C in Chapter 11 yields: a total PnL of €19927.22 with an average PnL of €11.32 per contract, a PD of 142.80, and a PT of 2.84. Relative to these results, A3C in this chapter generates 348% more profit, a 22% higher PD, and a 6% higher PT. Relaxing trading quantity constraints and using the DAM in arbitrage trading thus appears to increase the profit and reward-risk ratios.

Finally analysing A2C results, Table 12.2 shows that A2C trades: 92% more MWh per contract than HOLD, 35% less than PRE-BA, and 14% more than A3C. A2C yields the highest profit per contract: 6938% greater than HOLD, 6078% greater than PRE-BA, and 10% greater than A3C. Evaluated across test set contracts, Figure 12.1 shows that A2C gradually and consistently increases revenues; surpassing A3C total revenues by the final test contract. Despite A3C having a marginally higher PT, evaluated using the PnL and PD, A2C is found to be the best arbitrage trading agent across short-term markets.

12.6 Conclusion

In this chapter, arbitrage trading agents capable of trading across the day-ahead (DAM), continuous intraday (CID), and balancing (BAL) markets were developed and evaluated. A rule-based trading method, using forecasts of DAM prices and CID volume-weighted average price of trades (vwap), was developed to open positions on the DAM. Using these forecasts, 74% of positions were accurately opened across the DAM following our rule-based trading method.

Focusing on the CID and BAL, a deep reinforcement learning (DRL) agent, employing the synchronous advantage actor-critic algorithm (A2C), was trained. Behaviour cloning, i.e. goal-based exploration, was employed to increase the performance of the agent. A two-headed shared deep neural networks was used to determine the agent's policy. The performance of the agent was compared against three benchmark policies: HOLD, PRE-BA and A3C. A2C surpassed A3C and significantly outperformed HOLD and PRE-BA.

Overall, using TIs, data augmentation, ensemble model and A2C, our best agent was found to trade 33805.10 MWh of electricity across 1760 hourly test contracts; yielding significantly positive profits of €97853.69.

Chapter 13

Conclusion and Recommendations

In implementing arbitrage trading strategies, traders can help in the transition away from fossil fuels, combating climate change and helping attain energy independence. Strategies for arbitrage trading provide the chance to boost market liquidity, improve market efficiency, eliminate price discrepancies across markets, and generate a profit. Three of these opportunities counteract the challenges renewable energy creation creates. Helping arbitragers, thus, presents an opportunity to help the climate.

This PhD thesis has aimed to automate and optimise the decision-making of arbitrage trading agents for short-term electricity markets. Multiple contributions have been made to the existing literature by optimising the decision-making of intelligent agents. The thesis has developed new methods for forecasting electricity prices and algorithms for optimising trading decisions. All developed methods can be deployed immediately into production. This chapter summarises the thesis' main contributions and conclusions, potential impacts on society, and avenues for future research.

13.1 Conclusion

Throughout the thesis, we have analysed and devised state-of-the-art algorithms for exploiting statistical arbitrage trading (SAT) opportunities emerging across short-term electricity markets. Proposed novel forecasting methods significantly increased the accuracy of short-term electricity price forecasts. After predicting prices, a rule-based trading agent was developed for the day-ahead market (DAM), and a deep reinforcement learning trading agent was developed for the continuous intraday market (CID) and balancing market (BAL). Proposed novel trading methods increased the profitability of arbitrage trading strategies across short-term electricity markets. The main contributions of the thesis, shown in Figure 1.2, are summarised below. They help advance the fields of intraday market analytics, electricity price forecasting and electricity trading.

Intraday Market Analytics

This thesis presented market insights obtained using novel exploratory visualisation techniques for the CID. Our contribution, described below, helps to broaden existing domain knowledge about the CID, exposing CID price drivers and revealing CID trading strategies.

1. **Exploratory visual analytics tools for the CID:** Motivated by a need to explore trends and behaviours of the CID, exploratory visual analytics tools were developed for the CID. In Chapter 3, analytics tools, such as CID transactions, executed trades, the volume-at-price, and distributions of prices and volumes, were developed using data from the limit order book and trade book. Using these tools, we showed that traders could continuously monitor the market trend, behaviour, depth, price consensus and liquidity. An additional analytics tool was also developed using CID volumes and short-term market prices. This tool identified potential risks and opportunities associated with electricity trading.

Electricity Price Forecasting

This thesis identified novel forecasting methods capable of increasing the accuracy of electricity price forecasts. Accurate forecasts are required when training profitable SAT agents. Our contributions ensure that price uncertainty is decreased, increasing potential trading profits.

2. **Technical indicators for forecasting DAM prices:** Motivated by a desire to capture the effects of technical DAM price drivers, technical indicators (TIs) were proposed as features in Chapter 5. Through a case study, which involved forecasting Belgian DAM prices, it was demonstrated that TIs can capture the residual impacts from traders' behavioural biases, resulting in statistically significant reductions in forecast errors of machine learning models; reducing the root mean squared errors of linear, ensemble, and deep model forecasts by up to 4.50%, 5.42%, and 4.09% respectively. Four TIs, namely the Exponential Moving Average indicator (EMA), Bollinger Bands (%B), the Momentum indicator (MOM) and the Rate of Change indicator (ROC), were identified as being well suited to forecasting DAM prices. While TI performance is model dependent, the ROC and the EMA succeeded in reducing forecast errors of most machine learning models.

3. **Data augmentation methods for forecasting DAM prices:** Motivated by a desire to boost out-of-sample prediction accuracies, in Chapter 6, data augmentation methods were developed to artificially expand the training set size. Our augmentation methods utilised autoencoders (AEs), variational autoencoders (VAEs) and Wasserstein generative adversarial networks with a gradient penalty (WGAN-GPs). Through two case studies, which focused on forecasting Belgian and Dutch DAM prices, it was demonstrated that augmentation methods can significantly boost the regression accuracies of both autoregressive models with exogenous inputs and artificial neural networks. AEs, VAEs, and WGAN-GPs were found to significantly reduce a majority of forecast errors; on average decreasing mean absolute errors by 2.23%, 2.73% and 2.97% respectively. Taking every result into consideration, VAEs and WGAN-GPs were found to be the best and most stable data augmentors.

4. **Ensemble methods for forecasting DAM prices:** Motivated by a desire to improve forecasting accuracies of DAM prices, in Chapter 7, ensemble methods utilising different combinations of forecasts from TIs, AEs, VAEs and WGAN-GPs were proposed. Across ensemble methods for forecasting Belgian and Dutch DAM prices, the best-performing method was the method averaging AE, VAE, WGAN-GP and TI. This method outperformed the naive benchmark model by 27.63%.

This contribution is critical to the performance of SAT strategies because the rule-based trading agent needs accurate DAM price forecasts to open DAM positions correctly.

5. **A feature engineering method for forecasting CID prices:** Motivated by a desire to capture CID price drivers and an urge to improve forecasting accuracies of CID prices, in Chapter 8, a feature engineering method was proposed. Novel features were created from the limit order book and trade book and used as inputs to machine learning models. CID prices, such as the volume-weighted average price of trades and average best ask price, were predicted using our proposed feature pool. Forecasting models using the most significant features, such as the total number of submitted bid orders and the total number of revisions, were shown to boost the naive benchmark forecasting accuracies of Dutch CID prices on average by 15.27%. Note that as an additional tangential contribution from the above, artificial neural networks were shown to, on average, outperform machine learning models in forecasting CID prices by 6.62%.

   CID price forecasts have been used as inputs in our rule-based trading agents for the DAM and deep reinforcement learning agents for the CID and BAL. This contribution is thus vital in correctly opening DAM positions and making better decisions when trading across the CID and BAL.

6. **Machine learning methods for forecasting BAL prices:** Motivated by an urge to improve forecasting accuracies of BAL prices, in Chapter 9, machine learning methods were proposed. Artificial neural networks were shown to outperform the naive benchmark model by 19% and all other machine learning models on average by 8.43% when forecasting Dutch BAL prices.

   BAL price forecasts have been used as state variables in deep reinforcement learning trading agents. This contribution thus aids in ensuring that trading agents make profitable decisions across the CID and BAL.

Electricity Trading

This thesis developed novel trading methods that increase the profits of autonomous electricity trading agents. Each contribution, described below, helps to yield better decision-making systems for SAT strategies.

7. **Asynchronous advantage actor-critic (A3C) agent for statistical arbitrage trading across the CID and BAL:** Motivated by a desire to develop an autonomous trading agent capable of exploiting arbitrage trading opportunities across the CID and BAL, an algorithm employing a deep reinforcement learning method, namely asynchronous advantage actor-critic (A3C), was proposed in Chapter 11. Function approximators were designed with two-headed shared deep neural networks. A risk-constrained trading strategy was enforced by limiting the maximum allowed position. State engineering and selection processes were conducted to increase the performance of the function approximators. A novel reward function and behaviour cloning were introduced to motivate trading agents to explore. Our proposed A3C implementation successfully optimised the risk-reward ratio and yielded profits of €19927 by arbitraging 7017 MWh of electricity across the Dutch CID and BAL.

8. **Rule-based (RB) and advantage actor-critic (A2C) agents for statistical arbitrage trading across the DAM, CID and BAL:** Motivated by an urge to develop autonomous trading agents capable of exploiting arbitrage opportunities across the DAM, CID and BAL, a rule-based (RB) trading agent was proposed for the DAM, and the advantage actor-critic (A2C) trading agent was proposed for the CID and BAL in Chapter 12. A risk-constrained trading strategy was implemented by limiting the maximum exposure for each market layer. The RB agent, which takes forecasts of DAM and CID prices as inputs, was found to open accurate positions on the Dutch DAM 74% of the time. The A2C agent, which was designed with function approximators of two-headed shared deep neural networks, closed the DAM positions opened by the RB agent. Behaviour cloning, i.e. goal-based exploration, was integrated to increase the performance of A2C. A2C was compared against several benchmark policies, including the A3C. A2C was found to surpass A3C and significantly outperform other benchmarks. The RB and A2C agents successfully optimised the risk-reward ratio and yielded high returns of €97853 by arbitraging 33805 MWh of electricity across the Dutch DAM, CID and BAL.

13.2 Impact of the Research

This thesis aimed to optimise and automate SAT agents' decision-making for short-term electricity markets. In optimising the decision-making of SAT agents, this thesis has generated multiple contributions which have the potential to positively impact society. As we outline below, the thesis has identified strategies for aiding to improve the efficiency of electricity markets, which in turn should quicken the transition from non-renewable energy sources to renewable energy sources, helping fight climate change and secure Europe's energy independence.

Electricity Markets and Participants

Transitioning from non-renewable to renewable sources of energy has increased short-term market volatility. This thesis has proposed SAT strategies which, when enacted, have the potential to reduce price volatility across short-term electricity markets, as well as increase market liquidity and efficiency.

More accurate forecasts, secured using novel techniques proposed in this thesis, such as TIs and data augmentation, should decrease market risk. Decreased market risk should, in turn, increase both market participants' willingness to spread trades across short-term markets and implement SAT strategies. More intermarket trading and enactment of SAT strategies - particularly automated SAT strategies proposed in this thesis - should, in turn, spur short-term market liquidity and efficiency, facilitating continued renewable energy uptake.

Overall, this PhD thesis has thus outlined methods to reduce market participants' risks associated with price uncertainty, market liquidity, and intermarket price volatility.

Renewable Energy Transition

As part of their strategy to combat climate change, policymakers have subsidised and supported consumers and suppliers of clean and fully renewable energy. The energy supply from renewable sources is, at present, hard to predict for longer forecasting horizons. Renewable producers prefer to trade on the CID using their short-term forecasts because it allows them to hedge their volatile positions continuously.

To aid the sector in transitioning from non-renewable energy sources to renewable sources of energy, this thesis has developed exploratory analytics tools, forecasting methods and automated trading strategies for the CID. These methods guide and incentivise renewable energy producers to better use the CID for their trading decisions.

13.3 Suggestions for Future Research

While this thesis develops novel electricity price forecasting and SAT methods, more research is needed to achieve continued advances. Below, potential avenues of future research are described.

Exploratory Visual Tools for the CID

During tumultuous times, when market conditions are changing, visualisation tools for analysing market trends can yield new insights. In this thesis, visual analytics tools, such as graphs of transaction flows and ask/bid prices, were shown to be able to identify novel trading opportunities and risks. Given that our implemented tools successfully analysed CID trading activity, future research could focus on novel tools for exploring CID trading activity further. Indicators, similar to Bollinger Bands, for example, could be visualised.

Technical Indicators for Forecasting DAM Prices

Since this thesis is the first to successfully use technical indicator (TI) features in DAM price forecasting, it is reasonable to argue that the potential of technical feature engineering should be investigated further. While this thesis has examined the explanatory power of some of the simplest and widely used TIs in finance, further research could explore the predictive power of more complex TIs. Moreover, given the unique qualities of the DAM, future research could focus on engineering new TIs, which incorporate DAM volumes, order book data, etc. to capture electricity traders' behavioural biases.

Data Augmentation for Forecasting DAM Prices

Changing market conditions, epitomised by, for instance, increasing renewable energy production, have impacted DAM prices drastically in recent years. This shift in energy production has created breakpoints, making the inclusion of historic data in the learning process potentially detrimental. When data is scarce it is crucial to explore techniques capable of augmenting the training dataset size.

Given the successes of data augmentation methods in increasing DAM price forecast accuracies, data augmentation methods should be investigated further. While this thesis has examined autoencoders (AEs), variational autoencoders (VAEs) and Wasserstein generative adversarial networks with a gradient penalty

(WGAN-GPs), further research could examine variants of these successful augmentors, such as adversarial VAEs and Info-GANs. Given the successes of combined and ensemble augmentors shown in this thesis, further research could also explore more sophisticated bootstrapping algorithms and novel averaging methods for ensemble augmentation.

Moreover, generative adversarial networks (GANs) could be used for scenario generation. A myriad of GANs exists, including TransGANs, that could be integrated into the adversarial augmentation algorithm shown in this thesis to generate correlated time series. Researchers may consider using context encoders, context-conditional GANs, or conditional WGAN-GPs in future research.

Feature Engineering for Forecasting CID Prices

The quality and quantity of inputs that are fed into forecasting models are generally more important than the model chosen to forecast targets. Given the significance features play in ensuring high forecasting accuracies, the investigation of all possible price drivers for CID price forecasting is imperative.

Manually engineered features, which extract statistical information from the limit order book and trade book, were selected as important price predictors for forecasting CID prices in this thesis. Given that these features managed to capture CID price drivers, further research could be directed towards examining novel techniques for manually engineering features.

Alternatively, further research could focus on investigating automated feature engineering methods. Deep neural network encoding, for instance, could be utilised to extract latent features from the limit order book. These latent features could be used as input features in forecasting models, potentially boosting forecasting accuracies. Similarly, principal component analysis (PCA) could be utilised to extract principal components that could be used as input features in forecasting models.

Methods to Forecast CID Prices

Accurate CID price forecasts are vital in the identification of profitable trading strategies. Despite this, note that an overwhelming proportion of electricity price forecasting research has thus far focused on the DAM. The rising importance of the CID, however, enhances the need for CID price forecasting studies.

This thesis has investigated machine learning forecasting models and engineered features extracted from the limit order book and trade book for CID forecasting. While a successful approach was found for improving CID forecasting accuracies, more research is needed to improve upon existing CID forecasting methods. ResNet or SqueezeNet, for instance, could be evaluated as CID forecasting models. Moreover, lagged BAL prices or TIs could be evaluated as feature inputs, assessing their explanatory power in forecasting CID prices. Finally, data augmentation methods could be evaluated in the context of CID forecasting.

Methods to Forecast BAL Prices

Similarly to the CID, the BAL has not received as much attention as the DAM. Accurately forecasting BAL prices, however, is vital to optimising the risk-reward ratio of any short-term electricity trading strategy.

While the existing literature and this thesis have explored a few models and features, more research is needed to increase the forecasting accuracies of BAL prices further. CID prices, for instance, can be evaluated as additional features in BAL forecasting models. Moreover, gated recurrent units (GRUs) or deep long short-term memory (LSTM) models could be evaluated. Finally, data augmentation methods could again be assessed.

Methods to Forecast Price Spreads

Trading agents use price forecasts to make decisions. A trading agent that used both DAM and CID price forecasts to open positions on the DAM was defined in this thesis. Instead of forecasting prices separately for each market floor, price spreads between markets could be predicted directly. The price spread between the DAM and CID, for instance, could be predicted using regression or classification models to open positions on the DAM. Studies focusing on forecasting price spreads could identify critical relationships between markets.

Rule-Based Approaches for Statistical Arbitrage Trading

Even though rule-based (RB) approaches are considered the simplest among trading methods, they can be very profitable with a good strategy. As RB trading agents used in this thesis were found to be successful for the DAM, RB approaches should be explored further for the DAM as well as the CID. Future

RB studies, for example, can explore additional inputs than just price forecasts, such as load and generation forecasts.

Moreover, an RB approach should be defined as a benchmark for all SAT studies. As this thesis offers a benchmark RB using the volume-weighted average price of trades for the CID, the same or similar RB approach should be tested as a benchmark RB approach to make easy comparisons between studies.

Supervised Learning Approaches for Statistical Arbitrage Trading

Supervised learning approaches are easier to implement than reinforcement learning (RL) approaches. The target, i.e. the sequence of desired trading actions, can be obtained by cloning the decisions of expert traders. This target can be continuous, e.g. price of a limit order, or discrete, e.g. buy, sell and hold. If the discrete target is chosen, classification models would have an unbalanced class problem resulting from a significantly higher number of hold decisions. Future research can tackle this by weighing classes during training or augmenting data for buy and sell classes. Future studies should analyse the performance of models, from statistical models to deep neural networks, for both continuous and discrete targets. Additionally, transformers using attention can be utilised to improve the forecasting accuracies of artificial neural networks.

The above-supervised learning approach is used solely as a trading algorithm. It, however, can also be integrated into RL approaches as a policy function to decrease training time while increasing performance. Future research should explore this to see whether performance improvement is attainable for RL trading algorithms.

Reinforcement Learning Approaches for Statistical Arbitrage Trading

RL approaches evaluated in this thesis were found to be profitable. Consequently, RL-based SAT strategies should be investigated further. Due to the complexity of RL systems, however, each RL component should be studied separately. Future research, for example, should evaluate more sophisticated: reward functions, action constraints, and function approximators. Instead of focusing on single artificial neural networks, the use of ensembles could be investigated. Moreover, future research could focus on exploration methods for increasing sample efficiency. Given that this thesis identified goal-based exploration as a useful training method, further goal-based exploration methods, which use novel expert knowledge, could be investigated further. Trading agents could also be incentivised to explore unknown states. States can be defined by the uncertainty of the state, future, model or value function. Alternatively, they can be defined by entropy maximisation methods. K-neighbors, for instance, could be used to separate unknown states from known states.

Finally, different RL methods, such as value-based or policy-based methods, should be further evaluated and compared. Future research, for example, could explore the use of the soft actor-critic (SAC) algorithm, proximal policy optimisation (PPO), Rainbow and Impala methods for SAT strategy determination. It would also be interesting to study whether adversarial attacks can increase the robustness of these RL trading algorithms.

Multi-Agent Reinforcement Learning Approaches for Statistical Arbitrage Trading

In multi-agent RL approaches, multiple RL agents collaborate to achieve a common goal. Considering RL approaches were found to be successful across the CID and BAL in this thesis, multi-agent RL approaches that have proved apt at solving SAT problems in other domains should also be investigated across electricity markets to find whether they are able to extract further performance improvements. Future research, for instance, could explore a multi-agent RL team of buy-and-sell agents for the CID and BAL.

Note that the adoption of multi-agent approaches in the determination of SAT strategies could yield solutions capable of covering several market layers. Future research could investigate, for instance, a team of DAM, CID and BAL trading agents trained together using RL algorithms.

Bibliography

[1] C. Saravia, Speculative trading and market performance: the effect of arbitrageurs on efficiency and market power in the New York electricity market, Center for the Study of Energy Markets (2003).

[2] S. Baltaoglu, L. Tong, Q. Zhao, Algorithmic bidding for virtual trading in electricity markets, IEEE Transactions on Power Systems 34 (1) (2019) 535–543. doi:10.1109/TPWRS.2018.2862246.

[3] R. Li, A. J. Svoboda, S. S. Oren, Efficiency impact of convergence bidding in the California electricity market, Journal of Regulatory Economics 48 (3) (2015) 245–284.

[4] S. Borenstein, J. Bushnell, C. R. Knittel, C. Wolfram, Inefficiencies and market power in financial arbitrage: a study of California's electricity markets, The Journal of Industrial Economics 56 (2) (2008) 347–378.

[5] J. Mather, E. Bitar, K. Poolla, Virtual bidding: Equilibrium, learning, and the wisdom of crowds, IFAC-PapersOnLine 50 (1) (2017) 225–232.

[6] W. W. Hogan, Virtual bidding and electricity market design, The Electricity Journal 29 (5) (2016) 33–47.

[7] L. Hadsell, The impact of virtual bidding on price volatility in New York's wholesale electricity market, Economics Letters 95 (1) (2007) 66–72.

[8] M. K. AlAshery, D. Xiao, W. Qiao, Second-order stochastic dominance constraints for risk management of a wind power producer's optimal bidding strategy, IEEE Transactions on Sustainable Energy 11 (3) (2019) 1404–1413.

[9] S. Demir, K. Kok, N. G. Paterakis, Exploratory visual analytics for the European single intra-day coupled electricity market, in: 2020 International Conference on Smart Energy Systems and Technologies (SEST), 2020, pp. 1–6. doi:10.1109/SEST48500.2020.9203043.

[10] S. Demir, K. Mincev, K. Kok, N. G. Paterakis, Introducing technical indicators to electricity price forecasting: A feature engineering study for linear, ensemble, and deep machine learning models, Applied Sciences 10 (1) (2019). doi:10.3390/app10010255.

[11] S. Demir, K. Mincev, K. Kok, N. G. Paterakis, Data augmentation for time series regression: Applying transformations, autoencoders and adversarial networks to electricity price forecasting, Applied Energy 304 (2021) 117695. doi:https://doi.org/10.1016/j.apenergy.2021.117695.

[12] S. Demir, K. Kok, N. G. Paterakis, Statistical arbitrage trading across electricity markets using advantage actor–critic methods, Sustainable Energy, Grids and Networks 34 (2023) 101023. doi:https://doi.org/10.1016/j.segan.2023.101023.

[13] S. Demir, B. Stappers, K. Kok, N. G. Paterakis, Statistical arbitrage trading on the intraday market using the asynchronous advantage actor–critic method, Applied Energy 314 (2022) 118912. doi:https://doi.org/10.1016/j.apenergy.2022.118912.

[14] R. Scharff, M. Amelin, Trading behaviour on the continuous intraday market Elbas, Energy Policy 88 (C) (2016) 544–557.

[15] P. Shinde, M. Amelin, A literature review of intraday electricity markets and prices, in: 2019 IEEE Milan PowerTech, 2019. doi:10.1109/PTC.2019.8810752.

[16] P. N. Biskas, I. G. Marneris, D. I. Chatzigiannis, C. G. Roumkos, A. G. Bakirtzis, A. Papalexopoulos, High-level design for the compliance of the Greek wholesale electricity market with the target model provisions in Europe, Electric Power Systems Research 152 (2017) 323 – 341. doi:https://doi.org/10.1016/j.epsr.2017.06.024.

[17] K. Neuhoff, N. Ritter, A. Salah-Abou-El-Enien, P. Vassilopoulos, Intraday markets for power: Discretizing the continuous trading?, SSRN Electronic Journal (01 2016). doi:10.2139/ssrn.2723902.

[18] All Nemo Committee, SIDC: Cross border intra-day market, https://www.nemo-committee.eu/sidc, [accessed on 11 February 2023].

[19] TenneT, https://www.tennet.org/, [accessed on 12 December 2020].

[20] L. Hirth, I. Ziegenhagen, Balancing power and variable renewables: Three links, Renewable and Sustainable Energy Reviews 50 (2015) 1035 – 1051. doi:https://doi.org/10.1016/j.rser.2015.04.180.

[21] F. Ocker, K.-M. Ehrhart, The “German Paradox” in the balancing power markets, Renewable and Sustainable Energy Reviews 67 (2017) 892 – 898. doi:https://doi.org/10.1016/j.rser.2016.09.040.

[22] ENTSO-E Transparency Platform, https://transparency.entsoe.eu/, [accessed on 12 December 2020].

[23] M. Schmidt, The Sankey diagram in energy and material flow management, Journal of Industrial Ecology 12 (1) (2008) 82–94. doi:10.1111/j.1530-9290.2008.00004.x.

[24] G. Morris, Candlestick Charting Explained: Timeless Techniques for Trading Stocks and Futures, Irwin, 1995.

[25] J. Lago, G. Marcjasz, B. De Schutter, R. Weron, Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark, Applied Energy 293 (2021) 116983. doi:https://doi.org/10.1016/j.apenergy.2021.116983.

[26] 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)

(2014) 1030 – 1081. doi:https://doi.org/10.1016/j.ijforecast.2014.08.008.

[27] J. Lago, F. D. Ridder, P. Vrancx, B. D. Schutter, Forecasting day-ahead electricity prices in Europe: The importance of considering market integration, Applied Energy 211 (2018) 890 – 903. doi:https://doi.org/10.1016/j.apenergy.2017.11.098.

[28] J. Lago, F. D. Ridder, B. D. Schutter, Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms, Applied Energy 221 (2018) 386 – 405. doi:https://doi.org/10.1016/j.apenergy.2018.02.069.

[29] B. Uniejewski, J. Nowotarski, R. Weron, Automated variable selection and shrinkage for day-ahead electricity price forecasting, Energies 9 (8) (2016) 621. doi:https://doi.org/10.3390/en9080621.

[30] B. Uniejewski, R. Weron, Efficient forecasting of electricity spot prices with expert and LASSO models, Energies 11 (2018) 2039. doi:10.3390/en11082039.

[31] U. Ugurlu, I. Oksuz, O. Tas, Electricity price forecasting using recurrent neural networks, Energies 11 (5) (2018). doi:10.3390/en11051255.

[32] S. Gunduz, U. Ugurlu, I. Oksuz, Transfer learning for electricity price forecasting, pre-print (2020). arXiv:2007.03762.

[33] W. Shi, Y. Wang, Y. Chen, J. Ma, An effective two-stage electricity price forecasting scheme, Electric Power Systems Research 199 (2021) 107416. doi:https://doi.org/10.1016/j.epsr.2021.107416.

[34] Z. Yang, L. Ce, L. Lian, Electricity price forecasting by a hybrid model, combining wavelet transform, ARMA and kernel-based extreme learning machine methods, Applied Energy 190 (2017) 291–305. doi:https://doi.org/10.1016/j.apenergy.2016.12.130.

[35] J. Zhang, Z. Tan, Y. Wei, An adaptive hybrid model for short term electricity price forecasting, Applied Energy 258 (2020) 114087. doi:https://doi.org/10.1016/j.apenergy.2019.114087.

[36] E. E. Elattar, S. K. Elsayed, T. A. Farrag, Hybrid local general regression neural network and harmony search algorithm for electricity price forecasting, IEEE Access 9 (2021) 2044–2054. doi:10.1109/ACCESS.2020.3048519.

[37] B. Uniejewski, G. Marcjasz, R. Weron, Understanding intraday electricity markets: Variable selection and very short-term price forecasting using LASSO, International Journal of Forecasting 35 (4) (2019) 1533 – 1547. doi:https://doi.org/10.1016/j.ijforecast.2019.02.001.

[38] S. Hagemann, Price determinants in the German intraday market for electricity: an empirical analysis, The Journal of Energy Markets 8 (2015) 21–45. doi:10.21314/JEM.2015.128.

[39] T. Janke, F. Steinke, Forecasting the price distribution of continuous intraday electricity trading, Energies 12 (22) (2019). doi:10.3390/en12224262.

[40] M. Gürtler, T. Paulsen, The effect of wind and solar power forecasts on day-ahead and intraday electricity prices in Germany, Energy Economics 75 (2018) 150–162.

[41] F. Ziel, Modeling the impact of wind and solar power forecasting errors on intraday electricity prices, in: 2017 14th International Conference on the European Energy Market (EEM), 2017, pp. 1–5. doi:10.1109/EEM.2017.7981900.

[42] K. Maciejowska, B. Uniejewski, T. Serafin, PCA forecast averaging—predicting day-ahead and intraday electricity prices, Energies 13 (14) (2020). doi:10.3390/en13143530.

[43] C. Pape, S. Hagemann, C. Weber, Are fundamentals enough? Explaining price variations in the German day-ahead and intraday power market, Energy Economics 54 (C) (2016) 376–387.

[44] M. Narajewski, F. Ziel, Ensemble forecasting for intraday electricity prices: Simulating trajectories, Applied Energy 279 (2020) 115801. doi:https://doi.org/10.1016/j.apenergy.2020.115801.

[45] C. Scholz, M. Lehna, K. Brauns, A. Baier, Towards the prediction of electricity prices at the intraday market using shallow and deep-learning methods, in: V. Bitetta, I. Bordino, A. Ferretti, F. Gullo, G. Ponti, L. Severini (Eds.), Mining Data for Financial Applications, Springer International Publishing, Cham, 2021, pp. 101–118.

[46] I. Oksuz, U. Ugurlu, Neural network based model comparison for intraday electricity price forecasting, Energies 12 (23) (2019). doi:10.3390/en12234557.

[47] A. Lucas, K. Pegios, E. Kotsakis, D. Clarke, Price forecasting for the balancing energy market using machine-learning regression, Energies 13 (20) (2020). doi:10.3390/en13205420.

[48] G. Klæboe, A. L. Eriksrud, S.-E. Fleten, Benchmarking time series based forecasting models for electricity balancing market prices, Energy Systems 6 (1) (2015) 43–61.

[49] J. Dumas, I. Boukas, M. M. de Villena, S. Mathieu, B. Cornélusse, Probabilistic forecasting of imbalance prices in the Belgian context, in: 2019 16th International Conference on the European Energy Market (EEM), IEEE, 2019, pp. 1–7.

[50] T. Jónsson, P. Pinson, H. A. Nielsen, H. Madsen, Exponential smoothing approaches for prediction in real-time electricity markets, Energies 7 (6) (2014) 3710–3732. doi:10.3390/en7063710.

[51] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001.

[52] P. Huber, J. Wiley, W. InterScience, Robust statistics, Wiley New York, New York, USA, 1981.

[53] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32. doi:10.1023/A:1010933404324.

[54] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139. doi:10.1006/jcss.1997.1504.

[55] J. H. Friedman, Greedy function approximation: A gradient boosting machine, Annals of Statistics 29 (2000) 1189–1232.

[56] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[57] L. Fausett (Ed.), Fundamentals of Neural Networks: Architectures, Algorithms, and Applications, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994.

[58] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals and Systems 2 (4) (1989) 303–314. doi:10.1007/BF02551274.

[59] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, 2016.

[60] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, M. Hasan, B. C. V. Esesn, A. A. S. Awwal, V. K. Asari, The history began from AlexNet: A comprehensive survey on deep learning approaches, pre-print (2018).

[61] Y. LeCun, Y. Bengio, The handbook of brain theory and neural networks, MIT Press, Cambridge, MA, USA, 1998, Ch. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258.

[62] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[63] Directive 2009/28/EC (the renewable energy directive), European Parliament, Council of the European Union, available online https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32009L0028 [accessed on 01 November 2019] (2019).

[64] Proposal for a regulation of the European parliament and of the council on the internal market for electricity, European Commission, available online https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52016PC0861R(01)from=EN [accessed on 11 September 2019] (2017).

[65] Internal market for electricity, European Parliamentary Research Service, available online http://www.europarl.europa.eu/RegData/etudes/BRIE/2017/595925/EPRS_BRI(2017)595925_EN.pdf [accessed on 01 November 2019] (2019).

[66] J. Murphy, Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications, New York Institute of Finance Series, New York Institute of Finance, New York, USA, 1999.

[67] C. Kirkpatrick, Technical Analysis : The Complete Resource for Financial Market Technicians, FT Press, Upper Saddle River, N.J, 2011.

[68] P. K. D. Rajashree Dash, A hybrid stock trading framework integrating technical analysis with machine learning techniques, The Journal of Finance and Data Science 2 (1) (2016) 42–57.

[69] J. I. Larsen, Predicting stock prices using technical analysis and machine learning, Master's thesis, Norwegian University of Science and Technology, Norway (2010).

[70] J.-S. T. Chun-Teh Lee, Trend-oriented training for neural networks to forecast stock markets, Asia Pacific Management Review 18 (2) (2013) 181 – 195.

[71] A. Kulp, D. Djupsjöbacka, M. Estlander, Managed futures and long volatility, AIMA Journal (2005) 27–28.

[72] S. Nison, Japanese Candlestick Charting Techniques: A Contemporary Guide to the Ancient Investment Techniques of the Far East, New York Institute of Finance, 1991.

[73] M. Pring, Technical Analysis Explained: The Successful Investor's Guide to Spotting Investment Trends and Turning Points, McGraw-Hill Education, 2014.

[74] S. N. Neftci, Naive trading rules in financial markets and Wiener-Kolmogorov prediction theory: A study of "technical analysis", The Journal of Business 64 (4) (1991) 549–571.

[75] I. Gurrib, Optimization of the double crossover strategy for the S&P500 market index, Global Review of Accounting and Finance 7 (1) (2016) 92–107.

[76] S.-H. Poon, C. W. Granger, Forecasting Volatility in Financial Markets: A Review, Journal of Economic Literature 41 (2) (2003) 478–539.

[77] A. Merrill, M and W patterns, Market Technicians Association (MTA) Journal Issue 7 (1980) 43–54.

[78] J. Bollinger, Bollinger on Bollinger Bands, Professional Finance & Investment, McGraw-Hill Education, 2002.

[79] E. S. C. Coppock, Practical relative strength charting, Trendex Corp., San Antonio, Tex. (1960).

[80] W. Blau, Momentum, Direction, and Divergence, Wiley Trader's Exchange, Wiley, 1995.

[81] Epex spot, Belgium, available online: https://www.belpex.be [accessed on 29 May 2019].

[82] G. E. P. Box, G. Jenkins, Time Series Analysis, Forecasting and Control, Holden-Day, Inc., San Francisco, CA, USA, 1990.

[83] R. Weron, Electricity price forecasting: A review of the state-of-the-art with a look into the future, HSC Research Reports HSC/14/07, Hugo Steinhaus Center, Wroclaw University of Technology (2014).

[84] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[85] F. Chollet, et al., Keras, available online: https://keras.io [accessed on 29 May 2019] (2015).

[86] F. X. Diebold, R. S. Mariano, Comparing predictive accuracy, Journal of Business & Economic Statistics 20 (1) (2002) pp. 134–144. doi:10.1198/073500102753410444.

[87] Y. S. Abu-Mostafa, M. Magdon-Ismail, H.-T. Lin, Learning from Data, Vol. 4, AMLBook New York, NY, USA:, 2012.

[88] C. Sun, A. Shrivastava, S. Singh, A. Gupta, Revisiting unreasonable effectiveness of data in deep learning era, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 843–852. doi:10.1109/ICCV.2017.97.

[89] A. L. Guennec, S. Malinowski, R. Tavenard, Data augmentation for time series classification using convolutional neural networks, ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data, Sep 2016, Riva Del Garda, Italy.

[90] T. T. Um, F. M. J. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, D. Kulić, Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks, in: Proceedings of the 19th ACM International Conference on Multimodal Interaction, ICMI 2017, ACM, New York, NY, USA, 2017, pp. 216–220. doi:10.1145/3136755.3136817.

[91] T. DeVries, G. W. Taylor, Dataset augmentation in feature space, pre-print (Feb. 2017). arXiv:1702.05538.

[92] J. Jorge, J. Vieco, R. Paredes, J.-A. Sánchez, J.-M. Benedí, Empirical evaluation of variational autoencoders for data augmentation, in: VISIGRAPP, 2018.

[93] Y. Luo, B. Lu, Eeg data augmentation for emotion recognition using a conditional Wasserstein GAN, in: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2018, pp. 2535–2538. doi:10.1109/EMBC.2018.8512865.

[94] T. Tran, T. Pham, G. Carneiro, L. Palmer, I. Reid, A Bayesian data augmentation approach for learning deep models, in: Advances in neural information processing systems, 2017, pp. 2797–2806.

[95] C. Bergmeir, R. Hyndman, J. Benítez, Bagging exponential smoothing methods using STL decomposition and box–cox transformation, International Journal of Forecasting 32 (2016) 303–312. doi:10.1016/j.ijforecast.2015.07.002.

[96] S. Smyl, K. Kuber, Data preprocessing and augmentation for multiple short time series forecasting with recurrent neural networks, in: 36th International Symposium on Forecasting, 2016.

[97] G. E. Hinton, R. S. Zemel, Autoencoders, minimum description length and Helmholtz free energy, in: J. D. Cowan, G. Tesauro, J. Alspector (Eds.), Advances in Neural Information Processing Systems 6, Morgan-Kaufmann, 1994, pp. 3–10.

[98] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 2672–2680.

[99] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification, Neurocomputing 321 (2018) 321 – 331. doi:https://doi.org/10.1016/j.neucom.2018.09.013.

[100] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, ICML '08, ACM, New York, NY, USA, 2008, pp. 1096–1103. doi:10.1145/1390156.1390294.

[101] Y. Bengio, G. Mesnil, Y. Dauphin, S. Rifai, Better mixing via deep representations, in: International conference on machine learning, 2013, pp. 552–560.

[102] D. P. Kingma, M. Welling, Auto-encoding variational Bayes, pre-print (Dec. 2013). arXiv:1312.6114.

[103] Z. Hu, Z. Yang, R. Salakhutdinov, E. P. Xing, On unifying deep generative models, pre-print (2017). arXiv:1706.00550.

[104] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, MIT Press, Cambridge, MA, USA, 2014, pp. 2672–2680.

[105] M. Arjovsky, L. Bottou, Towards principled methods for training generative adversarial networks, pre-print (2017). arXiv:1701.04862.

[106] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, pre-print (Jan. 2017). arXiv:1701.07875.

[107] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, Improved training of Wasserstein GANs, in: Advances in neural information processing systems, 2017, pp. 5767–5777.

[108] ENTSO-E, available online https://www.entsoe.eu/data/transparency-platform/ [accessed on 16 January 2020].

[109] Scholt Energy, https://www.scholt.com, [accessed on 12 December 2020].

[110] A. Creswell, K. Arulkumaran, A. A. Bharath, On denoising autoencoders trained to minimise binary cross-entropy, pre-print (2017). arXiv:1708.08487.

[111] V. S. Cherkassky, F. Mulier, Learning from Data: Concepts, Theory, and Methods, 1st Edition, John Wiley Sons, Inc., USA, 1998.

[112] N. Ludwig, S. Feuerriegel, D. Neumann, Putting big data analytics to work: Feature selection for forecasting electricity prices using the LASSO and random forests, Journal of Decision Systems 24 (1) (2015) 19–36. doi:10.1080/12460125.2015.994290.

[113] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32. doi:10.1023/A:1010933404324.

[114] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, beta-VAE: Learning basic visual concepts with a constrained variational framework, in: ICLR, 2017.

[115] H. Petzka, A. Fischer, D. Lukovnicov, On the regularization of Wasserstein GANs, pre-print (2017). arXiv:1709.08894.

[116] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1125–1134.

[117] C. Wang, C. Xu, C. Wang, D. Tao, Perceptual adversarial networks for image-to-image transformation, IEEE Transactions on Image Processing 27 (8) (2018) 4066–4079. doi:10.1109/TIP.2018.2836316.

[118] J. J. Zhao, M. Mathieu, Y. LeCun, Energy-based generative adversarial network, pre-print (2016). arXiv:1609.03126.

[119] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (6) (1945) 80–83.

[120] H. B. Mann, D. R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, The annals of mathematical statistics (1947) 50–60.

[121] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B (Methodological) 58 (1) (1996) 267–288.

[122] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Physical review E 69 (6) (2004) 066138.

[123] D. Xiao, J. C. do Prado, W. Qiao, Optimal joint demand and virtual bidding for a strategic retailer in the short-term electricity market, Electric Power Systems Research 190 (2021) 106855. doi:https://doi.org/10.1016/j.epsr.2020.106855.

[124] D. Xiao, W. Qiao, L. Qu, Risk-constrained stochastic virtual bidding in two-settlement electricity markets, in: 2018 IEEE Power Energy Society

General Meeting (PESGM), 2018, pp. 1–5. doi:10.1109/PESGM.2018.8586115.

[125] H. Mehdipourpicha, S. Wang, R. Bo, Developing robust bidding strategy for virtual bidders in day-ahead electricity markets, IEEE Open Access Journal of Power and Energy 8 (2021) 329–340. doi:10.1109/OAJPE.2021.3105097.

[126] W. Tang, R. Rajagopal, K. Poolla, P. Varaiya, Model and data analysis of two-settlement electricity market with virtual bidding, in: 2016 IEEE 55th Conference on Decision and Control (CDC), 2016, pp. 6645–6650. doi:10.1109/CDC.2016.7799292.

[127] Y. Li, N. Yu, W. Wang, Machine learning-driven virtual bidding with electricity market efficiency analysis, IEEE Transactions on Power Systems 37 (1) (2022) 354–364. doi:10.1109/TPWRS.2021.3096469.

[128] L. Pozzetti, J. Cartlidge, Trading electricity markets using neural networks, Proceedings of the 32nd European Modeling Simulation Symposium (EMSS 2020) (2020) 311–318doi:https://doi.org/10.46354/i3m.2020.emss.045.

[129] E. Garnier, R. Madlener, Balancing forecast errors in continuous-trade intraday markets, Energy Syst 6 (2015) 361–388. doi:https://doi.org/10.1007/s12667-015-0143-y.

[130] R. Aïd, P. Gruet, H. Pham, An optimal trading problem in intraday electricity markets, Mathematics and Financial Economics 10 (2016) 49–85. doi:10.1007/s11579-015-0150-8.

[131] N. Čović, F. Braeuer, R. McKenna, H. Pandžić, Optimal PV and battery investment of market-participating industry facilities, IEEE Transactions on Power Systems 36 (4) (2021) 3441–3452. doi:10.1109/TPWRS.2020.3047260.

[132] G. Bertrand, A. Papavasiliou, Adaptive trading in continuous intraday electricity markets for a storage unit, IEEE Transactions on Power Systems 35 (3) (2020) 2339–2350. doi:10.1109/TPWRS.2019.2957246.

[133] I. Boukas, D. Ernst, T. Théate, A. Bolland, A. Huynen, M. Buchwald, C. Wynants, B. Cornélusse, A deep reinforcement learning framework for continuous intraday market bidding, Mach Learn (2021).

[134] T. Brijs, F. Geth, C. De Jonghe, R. Belmans, Quantifying electricity storage arbitrage opportunities in short-term electricity markets in the CWE region, Journal of Energy Storage 25 (2019) 100899. doi:https://doi.org/10.1016/j.est.2019.100899.

[135] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, 2nd Edition, The MIT Press, 2018.

[136] D. Qiu, Y. Ye, D. Papadaskalopoulos, G. Strbac, Scalable coordinated management of peer-to-peer energy trading: A multi-cluster deep reinforcement learning approach, Applied Energy 292 (2021) 116940. doi:https://doi.org/10.1016/j.apenergy.2021.116940.

[137] M. Dorokhova, Y. Martinson, C. Ballif, N. Wyrsch, Deep reinforcement learning control of electric vehicle charging in the presence of photovoltaic generation, Applied Energy 301 (2021) 117504. doi:https://doi.org/10.1016/j.apenergy.2021.117504.

[138] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, P. Abbeel, High-dimensional continuous control using generalized advantage estimation, pre-print (2015). arXiv:1506.02438.

[139] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: M. F. Balcan, K. Q. Weinberger (Eds.), Proceedings of The 33rd International Conference on Machine Learning, Vol. 48 of Proceedings of Machine Learning Research, PMLR, New York, New York, USA, 2016, pp. 1928–1937.

[140] Q. Kang, H. Zhou, Y. Kang, An asynchronous advantage actor-critic reinforcement learning method for stock selection and portfolio management, in: Proceedings of the 2nd International Conference on Big Data Research, ICBDR 2018, Association for Computing Machinery, New York, NY, USA, 2018, p. 141–145. doi:10.1145/3291801.3291831.

[141] Y. Li, W. Zheng, Z. Zheng, Deep robust reinforcement learning for practical algorithmic trading, IEEE Access 7 (2019) 108014–108022. doi:10.1109/ACCESS.2019.2932789.

[142] D. Cao, W. Hu, X. Xu, T. Dragičević, Q. Huang, Z. Liu, Z. Chen, F. Blaabjerg, Bidding strategy for trading wind energy and purchasing reserve of wind power producer – a DRL based approach, International Journal of Electrical Power Energy Systems 117 (2020) 105648. doi:https://doi.org/10.1016/j.ijepes.2019.105648.

[143] S. Glas, R. Kiesel, S. Kolkmann, M. Kremer, N. G. von Luckner, L. Ostmeier, et al., Intraday renewable electricity trading: advanced modeling and numerical optimal control, J. Math. Industry 10 (3) (2020). doi:https://doi.org/10.1186/s13362-020-0071-x.

[144] C. Kath, F. Ziel, Optimal order execution in intraday markets: minimizing costs in trade trajectories, pre-print (2020). arXiv:2009.07892.

[145] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.

[146] H. Yang, X.-Y. Liu, S. Zhong, A. Walid, Deep reinforcement learning for automated stock trading: An ensemble strategy, in: Proceedings of the First ACM International Conference on AI in Finance, ICAIF '20, Association for Computing Machinery, New York, NY, USA, 2020. doi:10.1145/3383455.3422540.

[147] Z. Zhang, S. Zohren, S. Roberts, Deep reinforcement learning for trading, The Journal of Financial Data Science 2 (2) (2020) 25–40. doi:10.3905/jfds.2020.1.030.

[148] Y. Liu, D. Zhang, H. B. Gooi, Data-driven decision-making strategies for electricity retailers: A deep reinforcement learning approach, CSEE Journal of Power and Energy Systems 7 (2) (2021) 358–367. doi:10.17775/CSEEJPES.2019.02510.

Curriculum Vitae

In 2008, S. Demir obtained her B.Sc. degree in mathematics from Ondokuz Mayis University, Turkey. She subsequently pursued an M.A. degree in philosophy from Istanbul University, graduating with a major in non-euclidean geometries. While pursuing the M.A. degree, she concurrently worked as a mathematics teacher; helping underprivileged children.

Following her years as a graduate/teacher, S. Demir was awarded the prestigious YLSY scholarship to complete an M.Sc. degree in applied statistics and data mining at the University of St Andrews, the UK. She graduated in 2017, earning a distinction in her thesis. During her M.Sc., she worked on sentiment analysis evaluating the credit risks of publicly listed companies; spearing collaboration efforts between the University of St Andrews and Moody's Analytics.

In 2018, S. Demir started her PhD at the Eindhoven University of Technology, The Netherlands. Her PhD research was funded by Scholt Energy and focused on forecasting electricity prices and developing algorithmic trading strategies using machine learning and reinforcement learning methods. During her PhD, she developed novel data augmentation and feature engineering methods to boost the forecasting accuracies of machine learning models. Moreover, she devised novel reinforcement learning trading environments to improve the performance of actor-critic trading agents. As well as optimising trading strategies, she has acted as a reviewer for distinguished journals and conferences, such as Applied Energy, Energies, Wind Energy, and International Conference on SEST. Furthermore, she has assisted B.Sc. and M.Sc. students with their AI-based projects. This culminated in one of her mentees being nominated for the best thesis award, and another mentee publishing his thesis at a conference.

List of Publications

S. Demir has the following publications:

Journal Articles

- S. Demir, K. Kok, N. G. Paterakis, “Statistical Arbitrage Trading Across Electricity Markets Using the Advantage Actor-Critic Methods,” *Sustainable Energy, Grids and Networks*, 34, 101023, 2023. DOI: 10.1016/j.segan-2023.101023

- S. Demir, B. Stappers, K. Kok, N. G. Paterakis, “Statistical Arbitrage Trading on the Intraday Market Using the Asynchronous Advantage Actor-Critic Method,” *Applied Energy*, 314, 118912, 2022. DOI: 10.1016/j.apenergy.2022.118912

- S. Demir, K. Mincev, K. Kok, N. G. Paterakis, “Data Augmentation for Time Series Regression: Applying Transformations, Autoencoders and Adversarial Networks to Electricity Price Forecasting,” *Applied Energy*, 304, 117695, 2021. DOI: 10.1016/j.apenergy.2021.117695

- S. Demir, K. Mincev, K. Kok, N. G. Paterakis, “Introducing Technical Indicators to Electricity Price Forecasting: A Feature Engineering Study for Linear, Ensemble, and Deep Machine Learning Models,” *Applied Sciences*, vol. 10, no. 1, 2020. DOI: 10.3390/app10010255

Conference Papers

- F. Verdaasdonk, S. Demir, N. G. Paterakis, “Intra-day Electricity Market Bidding for Storage Devices using Deep Reinforcement Learning,” in *2022 International Conference on Smart Energy Systems and Technologies (SEST)*, 2022, pp. 1-6. DOI: 10.1109/SEST53650.2022.9898405

- S. Demir, K. Kok, N. G. Paterakis, “Exploratory Visual Analytics for the European Single Intra-day Coupled Electricity Market,” in *2020 International Conference on Smart Energy Systems and Technologies (SEST)*, 2020, pp. 1-6. DOI: 10.1109/SEST48500.2020.9203043