Universiteit Leiden

ICT in Business and the Public Sector

Arbitrage in Electricity Markets using AI and Photo-Voltaic Battery Energy Storage Systems

Name: Fabian G.B. Schut
Student-No: s1835262

Date: 28/09/2021

1st supervisor: Dr. W. Heijstek
2nd supervisor: Dr. J.N. van Rijn
External supervisor: M. Maarse

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Currently, a shift in electricity production to more sustainable solutions is taking place. This leads to challenges such as a mismatch between supply and demand, and an excess of electricity on peak times in the power grid. A battery could help solving these issues, however, their high prices make them less attractive. A trading strategy, which buys and sells electricity in favorable times, makes batteries more profitable and appealing. One aspect of the trading strategy is forecasting electricity prices to determine buy and sell signals. In this study, four artificial intelligence principles are compared in two Dutch electricity markets. Most electricity is settled on the day-ahead market, where all forecasts of electricity usage and generation are submitted for every hour of the following day and at least 12 hours before transmission. Surpluses and shortages in electricity are settled on the imbalance market, a real-time market system that does not require any submission in advance. Contrary to the day-ahead market where prices are settled every hour, prices in the imbalance market are settled every fifteen minutes. While using an extensive search for the best performing hyperparameters, gradient boosting appeared to be the best performing principle in the day-ahead market, achieving a mean absolute error of 8.85 EUR/MWh. However, a simple baseline strategy consisting of day-ahead prices 168 hours prior to forecasted prices already outperformed the artificial intelligence model. That is explained by the drop in electricity demand due to COVID-19. Due to real-time market information for the imbalance market, several minutes of input data collection are investigated. The best performing model with seven minutes of input data collection realizes a mean absolute error of 15.72 EUR/MWh. The best performing model over all input collection minutes in the imbalance market is the artificial neural network. This models outperforms all baseline methods in terms of mean squared error. With a 15 kWh battery and a inverter power capacity of 10 kW, the best performing models achieve 16.67 EUR, 521 EUR, and 529 EUR in profits during 2020 with trading strategies in the day-ahead market, imbalance market or a combination, respectively. These profits take into account the lifespan of the battery, which is set to a fixed number of charge and discharge cycles depending on the battery type. Without the lifespan restriction, a maximum profit of 650 EUR is achieved. This is already 65% of the maximum achievable profit with perfect forecasts. The best performing trading strategy with lifespan restriction from the imbalance market is applied in a real-world setting with a 15 kWh battery and 10 kW inverter. Charge and discharge instructions were sent to an external battery using modbus through a platform. This setting achieved a profit of 23.5 EUR in less than two weeks, which is an improvement of the static models. A real-world setup leads to more challenges, such as the difference in theoretically achievable charging power and the actual charging power, an architecture to steer the battery that depends on multiple providers, and errors in the software application. An even better performance is possible if these challenges are mitigated. From these results, it is concluded that two artificial intelligence models have much potential in forecasting the electricity markets and that the imbalance market is most profitable. Improvement of the models could increase profits with 50% and may be achieved with a change in input data. Moreover, profits could also increase by adding electricity markets that allow value stacking. For a live implementation it is important to find the best architecture to steer the battery, while maintaining a fast, well tested software application that finds trading signals from the market. A recommendation for an organizational structure is given, which includes an information communication department and an agile mindset.

Table of contents

List of Figures

List of Tables

List of terms

artificial neural network A function of artificial intelligence that imitates the functioning of the human brain used for non-linear computation. With a self-learning mechanism it keeps optimizing with more data. 3

artificial intelligence Computational technique that is inspired by nature-wise thinking to excel in non-linear problem solving. 2

day-ahead market An electricity market that settles demand and supply of electricity for every hour, 12 to 24 hours before transmission. This is the market in the Netherlands where most electricity is exchanged. 2

gradient boosting A function of artificial intelligence that ensembles weak learning predictors into one strong predictor by iterating through the set of weak learners. 3

imbalance market An electricity market that settles the real-time deviation from the forecasted supply and demand in electricity that was submitted for the day-ahead market. The price is settled every fifteen minutes. 2

random forests A function of artificial intelligence that constructs a decision by combining multiple individual predictions into one prediction. 3

support vector regression A function of artificial intelligence that uses regression to forecast continuous values, by increasing the dimension of the data points. 3

List of abbreviations

AI artificial intelligence. 2, 3, 5, 7, 8, 9, 12, 13, 14, 15, 19, 21, 24, 25, 26, 27, 28, 31, 32, 33, 34, 35, 36, 37, 38, 40, 42, 43, 45, 46, 47, 50, 51, 59

ANN artificial neural network. 7, 8, 9, 10, 11, 12, 14, 16, 17, 32, 34, 35, 36, 38, 42, 50, 51, 57

API application programming interface. 25

APX Amsterdam power exchange. 2

AWS Amazon's web services. 41

BESS battery energy storage systems. 2, 3, 5, 9, 12, 13, 21, 33, 50, 51

CD critical distance. 21, 32

CEO chief execution officer. 47

CIO chief information officer. 47, 48

CTO chief technology officer. 47

DoC depth of charge. 13

DoD depth of discharge. 13

DSO distribution system operator. 5

EPEX European power exchange. 2, 6, 14

FCR frequency containment reserves. 51, 52

GB gradient boosting. 7, 12, 14, 20, 21, 32, 34, 35, 36, 38, 43, 50, 51, 58

IoT internet of things. 24, 45, 46

IT information technology. 46, 47, 48

MAE mean absolute error. 19, 20, 33, 34, 35, 41, 42, 44, 45, 46, 50, 51

MLP multi-layer perceptron. 16

MQTT message queue telemetry transport. 24

MSE mean squared error. 19, 20, 34, 35, 41, 42, 45, 46, 50, 51

PRP program responsible party. 5, 6, 12

PTU program time unit. 6, 14, 15, 16, 22, 23, 24, 25, 28, 29, 30, 31, 34, 35, 43, 45

REST representational state transfer. 25

RF random forests. 7, 11, 12, 14, 19, 20, 32, 34, 35, 36, 42, 43, 50, 58

SVR support vector regression. 7, 9, 10, 11, 12, 14, 17, 18, 19, 34, 35, 36, 42, 43, 50, 57

TSO transmission system operator. 5, 9, 11, 28

1. Introduction

This master research starts with an introduction to the research topic. First, the importance of this research topic is given in the background subsection. Then the introduction continues with the research objectives, research questions and gives an overview for the remaining sections in this study.

1.1. Background

Energy consumption has been increasing in the world over the last few decades [28], which has led to the global warming issues we currently face [52]. These issues include, among others, loss of agriculture and species, air pollution, and natural disasters [17]. That is why countries were forced to take actions to reduce climate change. The first global approach to tackle climate change was made in 2015: the Paris agreement. 196 Parties agreed that the annual average temperature cannot increase with more than 2 °C compared to pre-industrial times [48]. However, the world struggles achieving the goals mentioned in the agreement. Only half of the agreed decarbonization is realized and parties are falling behind with other commitments as well [39]. To achieve the goals mentioned in the Paris agreement, renewable energy sources should be almost fully responsible for the production of electricity. In 2019, renewable energy sources only contributed with 18.8% of the total electricity production in the Netherlands [10]. That is why a large transition in the electricity production is currently taking place. Note that not only the electricity production should be transitioned, the total energy production has to absorb as much CO_2-equivalents as it produces in 2050. CO_2-equivalents consist of carbon dioxide, methane, nitrous oxide, and fluorine. However, this study will only focus on the electricity production in the Netherlands. The share of electricity production in the total energy production was 16.8% in 2019 for the Netherlands. The forecast is that this segment will grow in the following years [21]. This growth can be explained by the increasing usage of electric cars and electrical heating.

However, increasing usage of electricity and renewable production will lead to complications. First of all, renewable electricity sources are not producing a constant amount of electricity during a day or season. On a windy day, there will be more production of electricity by wind farms. Furthermore, solar panels will not produce electricity at night. Second, an increasing number of electrical devices will consume electricity at the same time. For example, electrical heaters will use more energy during cold winters, and electric cars will mainly be charged at 6 PM when people get home from work. Both events lead to peaks in the power grid and an imbalance between supply and demand. Three solutions are possible, where one only handles the increasing peaks, while the others handle the increasing peaks and imbalance.

The first solution would be extension of the power grid. Renewable electricity that is produced and not directly consumed locally, is transferred back into the power grid. The power grid needs to be extended to handle the maximum power flow. This could for example be on an extremely windy and sunny day. Disadvantages of a power grid extension are the high investment costs, the long time to realize it, and the amount of space that is needed for the electricity network [8]. Furthermore, the extension of the power grid only handles the increasing peaks. The imbalance between consumption and production remains unsolved. Another candidate is the demand responsiveness. A reduction in demand peaks also ensures that the power grid needs less extension. Instead of consuming much electricity in a short time frame, the same amount of electricity is used over a longer time period. Moreover, high peaks in electricity production lead to high consumption. Therefore, less electricity is sent back to the power grid. The disadvantage of demand responsiveness is the loss of comfort, since the consumer has to change its behaviour [8]. The last solution is the storage of unused electricity in batteries. When a peak in electricity production is realized, the electricity can be stored at the producer. This ensures that the power grid has fewer high peaks, and the imbalance is solved locally. Battery disadvantages are the space and weight required for storage, duration to charge and discharge, and the high costs [8]. However, the price of batteries has dropped at a fast pace over the last ten years, and is expected to drop even further [25].

While a combination of these three solutions will probably be required to achieve the goals for 2050, this study focuses on the storage of electricity in batteries. Several batteries are available on the market. In general, batteries can be classified in lead acid, nickel-cadmium, nickel–metal hydride, lithium-ion, and lithium polymer [40]. A battery is selected on several characteristics such as costs, life time, self-discharge and density. Due to the high density, low self-discharge, and a connection to solar panels, this study looks at lithium-ion batteries in Dutch companies with solar energy production. The lithium-ion battery in this case is also known as the battery energy storage systems (BESS). Local production and consumption of solar energy are often not in balance. Since there is no possibility to consume solar electricity at night or on cloudy days, electricity that is not directly consumed can be stored in a battery. In times when there is a consumption surplus, the battery can be discharged. The battery ensures that the electricity peak load can be reduced, which in turn leads to a decrease in costs [54].

Another cost reduction can be achieved by selling and buying electricity from electricity markets when prices are high and low, respectively. Several markets are available in the Netherlands. Most electricity is bought on the European power exchange (EPEX) day-ahead market (previously known as Amsterdam power exchange (APX)) [57]. Market participants are able to buy and sell electricity one day before electricity is exchanged. Supply and demand settle to a single price per hour in the next day. Due to a forecasting error in production and consumption of electricity, a real-time market is introduced. This market is known as the imbalance market. Here, the price is settled every 15 minutes. Both markets are researched and forecasted in this study. The reason for those markets is the ease of implementation at small organizations. Both the day-ahead and the imbalance market do not have a minimum requirement of volume to trade on the market, while other markets have. Furthermore, much research is done in the day-ahead market, which could be used to build even better performing forecasters [4, 5, 36].

According to Mulder & Sholtens [41], the electricity price is dependent on many variables. A few examples are wind, sun, daylight and gas prices. Furthermore, the required balance between production and consumption of electricity leads to large price differences in a short time frame, making it a unique market [65]. That is not all, electricity prices show rare characteristics that can only be explained with their physical features. That is why the price is influenced by seasonality (e.g., annual, weekly or daily), has a long memory and price spikes, and is extremely volatile [11]. Due to the high price fluctuations, price forecasting has become a need for many participants in the electricity market [46]. Many approaches have been studied. These can be classified in multi-agent, fundamental, reduced-form, statistical, and artificial intelligence models [64]. Most of these approaches, however, have trouble forecasting highly volatile and non-linear price processes such as the electricity market [64]. Except for artificial intelligence (AI) models, as they excel in non-linear models. However, Weron [64] states that an optimal AI model is hard to find, since there are many models that are difficult to compare due to differences in errors. These errors contain, among others, the rare characteristics from the in-sample dataset that are not applicable on other datasets, and the initial parameters from the models. Only with identical in-sample, out-of-sample, and accuracy performance metrics, different AI models can be compared.

1.2. Research Objectives

Due to the problems that arise with the comparison of AI models in different studies, this study makes a contribution by comparing and improving AI models that forecast electricity prices with the same in-sample, out-of-sample and accuracy performance metrics. State-of-the-art models in electricity price forecasting, such as artificial neural network and support vector regression, are compared to less popular electricity price forecasters, e.g., random forests and gradient boosting. By looking at the less popular models too, a trade-off between using less popular forecasters or continue with popular forecasters can easily be made by other researchers. Both the popular and less popular AI models have their unique structure, leading to different behaviour and other forecasts. Therefore, the models will be further referred to as principles or approaches. On top of that, the AI models are applied and compared in a BESS trading strategy by charging and discharging a battery. This could lead to a lower average electricity price and stimulates companies to invest in solar energy sources, which in turn reduces the global warming issues. That is why this study benefits both researchers and society.

1.3. Research Questions

To achieve the research objectives, the following research question needs to be answered.

> *Which AI principles are applicable to create arbitrage on Dutch electricity markets with price forecasting by trading energy from a BESS?*

To answer the main research questions, several sub-questions are composed. This study starts with developing an understanding of several Dutch electricity markets to inventorize and define the features and restrictions of the forecasting models. This defines the first sub question.

1) What are the characteristics of different Dutch electricity markets and which markets are suitable for a trading strategy?

When the characteristics of the Dutch electricity market are defined, a collection and comparison of established forecasting models is made. Several previous work and suppliers of electricity price forecasting models are included. This sub-question is therefore defined as follows.

2) Which AI principles in electricity price forecasting are available and how do they differ?

Once an overview and comparison is made between several established forecasting principles, the models are improved further. This is covered in the following sub question.

3) How can AI principles in electricity price forecasting be improved to reduce forecasting errors?

After an AI model is built to forecast electricity prices on Dutch markets, it needs to be implemented in a trading strategy. Therefore, the fourth sub-question is defined as follows.

4) How can an electricity price forecasting model be used in a trading strategy?

The final step of this study is to implement the trading strategy into a real-world setting, that controls the flow of a battery energy storage system real-time. A comparison can then be made between static models and the real-world setting. This is covered in the final sub question.

5) How do real-time AI electricity trading strategies with a BESS compare to static models?

1.4. Study overview

The remainder of this study is structured as follows. Section 2 discusses several studies that have analysed the electricity markets and performed similar forecasts. Here, the background information for Dutch electricity markets is given as well. Section 3 explains what methods are used and the reasons for those methods. Section 4 continues with the experimental setup and the results that are collected during this study. Section 5 discusses the results and explains what they mean. Finally, Section 6 summarizes the study, answers the research question, shows the relation between the results and the objectives and draws conclusions.

2. Literature review

The literature section gives an overview of previous work that has been done in the field of electricity markets and electricity price forecasting. First, a thorough analysis of the Dutch electricity market is performed to provide insights that could help in the selection of features and the kind of trading model. Then, previous work of several AI forecasting principles in the electricity market is compared and analyzed.

2.1. Dutch electricity markets

Tanrisever et al. [57] have performed an in-depth analysis on the Dutch electricity market. This subsection provides a summary of the most important parts to understand the market for a forecasting and trading model.

The Dutch electricity market is a liberalized market where participants can freely trade electricity in a competitive environment. However, not all participants of the Dutch electricity supply chain need to fear competition. System operators have a natural monopoly since they are responsible for the maintenance of the grid, and only one grid is present per region. There are two types of system operators in the Netherlands, namely, transmission system operator (TSO) and distribution system operator (DSO). The primary function of the TenneT, the Dutch TSO, is maintaining grid balance. This is achieved by the inspection of forecasted transmissions and energy flows provided by market participants to ensure grid stability and balance, respectively. While TenneT is responsible for the high voltage grid, DSOs are responsible for the medium voltage grids. Nine DSOs transport and distribute electricity from the high voltage grid to the customer, all in their own region.

In addition to the system operators, there are four more parties involved in the supply chain. This includes the production company, program responsible party (PRP), metering company, and supplier. Production companies contain all participants that generate electricity that is sent to the grid. This could also be a non-energy company with many solar panels that sends electricity back to the grid. PRPs have a physical connection with the grid and need to forecast their difference between supply and demand. They send these forecasts to TenneT. If the actual usage deviates from the forecast, the imbalance is settled in the imbalance market, explained in Section 2.1.2. PRPs could also be seen as the traders of electricity. However, anyone is able to trade on the electricity market, but they should have a contract with a PRP.

After a settlement on the markets, electricity is sent through the grid to the customer. The customer has a contract with the supplier, who charges the customer based on their energy usage. In the Netherlands, there are 47 suppliers of energy. They are competing with each other in services and prices of electricity. The amount of energy that is passed through an individual connection is monitored by the metering company. They collect and send metering data to TenneT. An overview of the Dutch electricity supply chain can be found in Figure 1. This study focuses on the trading aspect of the supply chain, in combination with the generation of electricity with solar panels and BESSs.

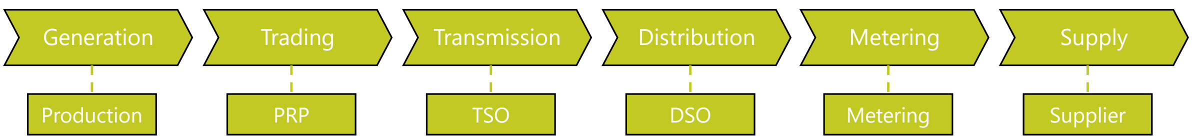


Figure 1: Electricity supply chain of the decentralized market in the Netherlands [56].

2.1.1. EPEX day-ahead market

One of two markets investigated in this research is the EPEX market. The EPEX consists of two different markets: The day-ahead market and the intraday market. Due to the significant larger volume on the day-ahead market [53] and the importance of volume in trading [6], the day-ahead market is selected for forecasting and trading.

On the day-ahead market, buyers and sellers place bids and offers for physical delivery of electricity during the next day. There are 24 supply and demand settlements, one for each hour of the delivery day. The market opens at midnight the day before, and closes 12 hours later at noon. This means that the first delivery starts 12 hours after closure, while the latest delivery is 36 hours after closure. One hour before closure, the available transmission capacity is published, which indicates the total capacity that is available during each hour of the day. Within one hour after closure, the results of the auction are published on the website of EPEX.

2.1.2. Imbalance market

Since the grid should always be in balance, and forecasts of the PRPs have small deviations with the actual consumption and delivery, a real-time balancing market is introduced: the imbalance market. When a PRP consumes more electricity than forecasted, TenneT sells the difference to that PRP at an unknown price. This electricity comes from another PRP that consumes less than forecasted, or from a large producer that is obligated to have spare electricity ready for the imbalance market. The imbalance price is determined after the settlement of supply and demand and is calculated over the total imbalance in 15 minutes, the so-called program time unit (PTU). Every PRP can make bids and offers to buy and sell their electricity on the imbalance market, respectively. These bids and offers make up the bid price ladder that is used as a reference for the actual imbalance price. An example of such a bid price ladder is shown in Figure 2, and can be interpreted as follows. At PTU 20, when the total market adjustment is +300 MW because of unforeseen extra electricity usage, the reference price upwards P_{up} for one MWh is 112.7 EUR. If there is an excess of electricity and the total market adjustment at PTU 20 is -600 MW, the reference price downwards P_{down} is equal to -118.51 EUR/MWh. Note that a PRP has to pay money to deliver unforeseen electricity in the imbalance market in the latter example. The bid ladder is updated every minute by TenneT, such that market participants can update their bids and offers based on real-time market movement. This means that the bid ladder changes during one PTU. However, the prices shown in Figure 2 are fixed, because the data is constructed after the delivery day has ended. TenneT buys or sells electricity according the bid ladder on behalf of the PRP that had a deviation in their consumption forecast.

The actual imbalance price that is used for settlement depends on the regulation states. Four different regulation states are possible, which are summarized in Table 1. In regulation state 0, small errors are settled with a single average price of the bid ladder. This imbalance price does not differ much from the EPEX day-ahead price. In regulation state +1 or -1 however, the imbalance price is equal to the highest marginal bid to buy and the lowest marginal bid to sell from the bid price ladder, respectively. These states are accountable for the high volatility in the imbalance market. The final regulation state 2 is settled with two prices from the bid price ladder. One price for the additionally bought electricity, and one for the additionally sold electricity. The prices are formally summarized in Table 2.

Due to the real-time market, near real-time data could be used to forecast the imbalance price. That arouses interest in the comparison between the day-ahead market, where a forecast for 12 to 36 hours ahead is made, and the real-time imbalance market, where a forecast for at maximum 15 minutes is made.

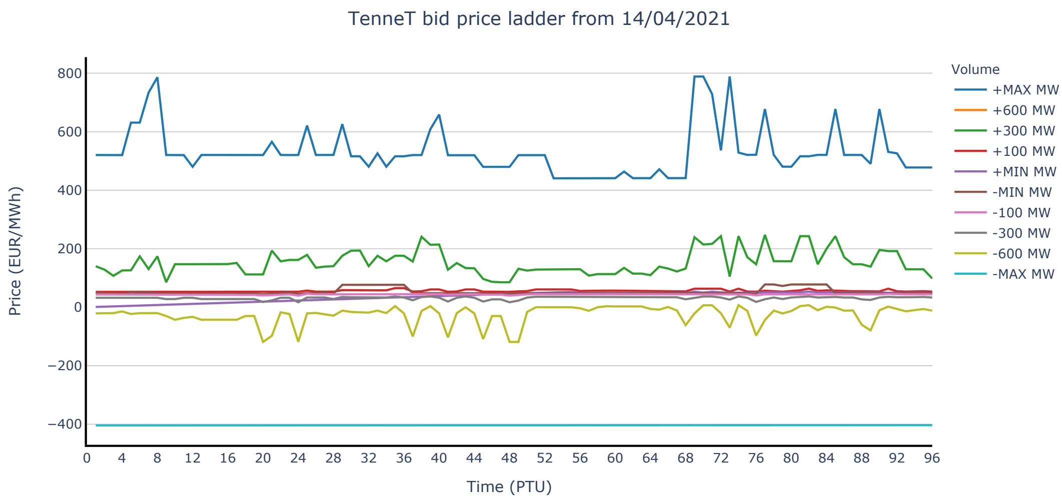


Figure 2: Dutch bid price ladder for 14/04/2021 published by TenneT [60]. The volume indicates the adjustment upwards (+) or downwards (-) in the imbalance market. One PTU equals 15 minutes.

Table 1: Overview of regulation states in the Dutch imbalance market [57].

Regulation state	Description
0	The consumption of electricity is equal to the forecast.
+1	More electricity is consumed than forecasted exclusively; TenneT mainly buys electricity.
-1	Less electricity is consumed than forecasted exclusively; TenneT mainly sells electricity.
2	Both more and less electricity is consumed than forecasted; TenneT buys and sells electricity.

2.2. AI forecasting principles

Much research has been done in the field of electricity price forecasting, and an overview of the state-of-the-art approaches is given by Weron [64] in 2014. According to that study, electricity price forecasters can be classified into multi-agent, fundamental, reduced-form, statistical, and artificial intelligence. This study will mainly focus on the AI forecasters. AI approaches could be further classified into well-known classifiers in electricity price forecasting such as artificial neural network (ANN) and support vector regression (SVR), and less popular classifiers such as random forests (RF) and gradient boosting (GB).

The difficulties that arise with a comparison of different AI approaches from different research

Table 2: Overview of the settlement prices in the Dutch imbalance market by TenneT [61]. P_{mid} is the average price of the lowest price upwards and the highest price downward from the bid ladder, where an example is given in Figure 2. P_{up} and P_{down} reflect the prices from the bid ladder given the upward or downward volume, respectively.

Regulation state	Settlement prices from bid ladder
0	P_{mid}
+1	P_{up}
-1	P_{down}
2	$MAX(P_{up}, P_{mid})$; if consumption is larger than forecast $MIN(P_{down}, P_{mid})$; if consumption is smaller than forecast

papers are the differences in evaluation robustness error, different datasets to train and test the approaches, and the absence of a statistical test for significance [64]. That is why several models from previous work will be analysed, reconstructed, trained and tested with the same dataset, and compared with each other, including a statistical test for significance. This section focuses on the analyses of previous work, such that a selection can be made for reconstruction.

All AI approaches will be used to forecast both the day-ahead market and the imbalance market. Most researchers try to forecast prices from the day-ahead market. Examples are one hour ahead forecasts [11, 46, 22], h hours ahead forecasts [46, 4], and one day-ahead complete price profiles [67]. Even though forecasts for prices in the longer term ($h > 24$) are less popular, the economic value is of great importance. That is because a forecast of 25 to 36 hours ahead is required to buy electricity for the last 12 hours of the delivery day in the Dutch day-ahead market, as stated in Section 2.1.1.

In contrast to the day-ahead forecasting models, there has been limited research into the imbalance market forecasting, especially with AI. However, some researchers have forecasted or described the imbalance market thoroughly. Interestingly, researchers do not agree on the long-term predictability of electricity balance markets. On the one hand, Klæbo et al. [32] claim that balance markets can not be accurately forecasted one day ahead, since all available market information before market closure is already reflected in the day-ahead market. While Peters [45] describes a model to forecast the long-term Dutch balance market. However, the latter study does not forecast individual imbalance prices. Therefore, this study attempts to forecast the imbalance market on short term (≤ 15 minutes).

2.2.1. Artificial neural network

ANN is the most widely used AI approach in forecasting electricity prices and electricity load [64, 42, 4, 11]. The architecture of the ANN defines the category that a model belongs to. One of these architecture options is the number of output nodes. Most ANN models only have one output and predict one value, e.g., a single price one hour ahead [22] or a single price 24 hours ahead [4]. However, some researchers try to predict 24 consecutive hours, and provide an ANN with 24 output nodes [67]. Another architecture option is the addition of a feedback loop. ANNs with a feedback loop are called recurrent networks. Without a feedback loop, they are called feed-forward networks. Feed-forward networks have three main advantages over recurrent networks. First, a feed-forward network always produces the same output with the same input, while a recurrent network is dependent on preceding inputs. For a comparison between several AI models, the same output is preferred. Second, a feed-forward network is able to find the best performing hyperparameters faster than recurrent networks, since recurrent networks go through to the input data more than once. Thirdly, recurrent networks could oscillate too much between values, such that no accurate forecast can be given [15]. Therefore, feed-forward networks are used in this study.

Feed-forward networks can be further classified into several groups. However, all feed-forward networks consist of layers. The most simple network, a single-layer perceptron, is similar to linear regression models. Here, only one input layer and one output layer are present. It can be used for forecasting by combining linear inputs. When a layer is added between the input nodes and the output nodes, the so-called hidden layer, the non-linear multi-layer perceptron is created. Multi-layer perceptrons solve a forecasting problem by using the output of one layer as input in the following layer. The number of hidden layers in a multi-layer perceptron is variable, and can be optimized. Every layer consist of a number of neurons. These neurons determine how the input values affect the outcome. Several types in activation could be used in an ANN, e.g., radial basis activation. A specific multi-layer perceptron with one hidden layer and a radial basis activation is called the radial basis function. Two other activations are the sigmoid and piecewise linear. A previous study shows that the latter two are more favorable by researchers [64]. The activation function determines if the value from one node is used as input for another node in the next layer. Therefore, the sum of weights multiplied with node values from the previous layer should exceed a certain threshold. The weights, numerical values that belong to the connection between nodes from one layer to another, can be optimized using several methods. The most popular training method is back-propagation, where continuous valued functions and supervised learning is applied.

Cruz et al. [14] have used two different activation functions in their multi-layer perceptron with one hidden layer. While the nodes in the hidden layer were activated using the hyperbolic tangent, the output layer used linear function as its activation function. The ANN was used to forecast all 24 hourly spot prices for day $D + 1$ on a given time in day D. In contrast with Chaâbane [11], Cruz et al. [14] forecast all prices for the following day before market closure. Therefore, this setting could be used for a trading strategy in real-world situations. Interestingly, the ANN does not outperform dynamic regression models. However, Keynia [30] and Amjady et al. [5] provide evidence that neural networks can outperform dynamic regression models if a hybrid setup is used. This hybrid setup consist of three neural networks that execute consecutively. The downside of the hybrid method is that the execution time triples, since three neural networks are run. Furthermore, both hybrid models use input data from electricity markets that can not be used in a forecast for real-world trading models, since that data is only available after the electricity markets are closed.

The number of hidden layers is one of the architecture settings that defines an ANN. Besides the number of hidden layers, many settings define the ANN and AI models in general. These settings are called the hyperparameters. The hyperparameters could be optimized in an AI model to get the most accurate forecast. Another hyperparameter for an ANN is the number of nodes in the hidden layers. The example in Figure 3 has one hidden layer with three nodes in the hidden layer. Not all related work provides this hyperparameter value. Cruz et al. [14] indicate that they have tested several configurations regarding the number of nodes. However, the final number of nodes where the results are based upon is not given. Neupane et al. [42] on the other hand state that they have used 10 nodes in the hidden layer with a 3 layer network. For the sake of reproducibility, it is important to know this hyperparameter configuration. That also holds for the feature extraction and feature selection steps to come up with a list of selected features. For example, Neupane et al. [42] have used many past price indicators from the last 24 hours and previous year, and used the wrapper method [34] to select a pool of features. Others also used forecasted wind production, provided by the TSO [14, 63], forecasted solar energy production [63], and electric load [63].

The ANN in this study will have one output node. While starting with a fixed configuration of hyperparameters, the final ANN model will be optimized in their hyperparameters. Furthermore, the best performing AI approach will be applied in a real-world trading setting, where energy from a BESS is used.

2.2.2. Support vector regression

Another popular AI approach is the SVR. SVR is also a non linear regression model that adds new dimensions to the data points, such that the similar data points can be forecasted. Although it is used less often than ANN in the electricity price forecasting [64], many applications are researched [47, 68, 23]. An SVR works as follows. A regression function in high dimensional space, the so-called hyperplane, tries to include as many data points as possible. The data points are included if they fall within the margin from the hyperplane. This margin is variable and can be optimized. Note that in SVR with classification, often referred to as support vector machine, the goal is to maximize the margin of the hyperplane while separating the different

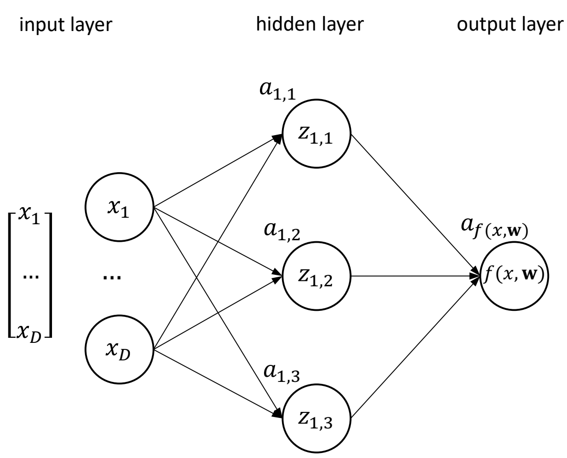


Figure 3: Example of a multi-layer perceptron with one hidden layer and one output node. Given the input vector $\mathbf{x}$, three nodes in the hidden layer and the single output layer are activated using an activation function a, which can differ per layer.

Table 3: Overview of SVR hyperparameter configuration from previous studies. The numbers in brackets indicate the best performing value for that hyperparameter found in that study.

Literature	kernel function	C [best]	γ
Guo et al. [23]	radial basis function	1-500 [-]	0.01-50 [-]
Sansom et al. [47]	-	0.1-5000 [0.5]	- [-]

classes. Not all data points can be included using a single linear hyperplane. That is where the kernel function is designed for. The kernel function maps non-linear data into a higher dimensional space such that a simple linear function could include as many data points as possible [23]. Different kernel functions are possible in an SVR. Guo et al. [23] have used one of the most popular kernel functions, namely, radial basis function.

Next to the kernel function, another hyperparameter needs optimization. This hyperparameter is expressed as C, and indicates the penalty that is given to regression errors. An infinite C allows no errors in the forecasting model. This results in a high dimensional SVR where no generalization is possible. A small C, on the contrary, generalizes much and creates a less complex SVRs than an infinite C. However, a smaller C also indicates that more errors are allowed, which could be seen in the accuracy. Sansom et al. [47] use a larger range of values, from 0.1 to 5000, for the optimization of C than Guo et al. [23], where the range is from 1 to 500. Notably, Sansom et al. [47] find an optimum of hyperparameter C at 0.5, which could not be found by Guo et al. [23] due to their range. Together with the hyperparameter γ, that configures the differences in feature vectors, the overfitting of the SVR is determined. γ is not available for all kernel functions, and that may explain why Sansom et al. [47] did not describe γ. However, Guo et al. [23] set their γ range from 0.01 to 50. An overview of these hyperparameter choices is given in Table 3.

An SVR differs much in the forecasting approach compared to ANNs. The consistent accuracy of an SVR makes it faster and simpler to optimize the hyperparameters than to optimize the architecture and hyperparameters of an ANN [47]. Since an ANN gets randomly assigned initial weights at the beginning of the optimization, the consistency is missing. Furthermore, SVRs are deterministic, thus create a single output with no option to create probability distributions. Finally, the discussed literature with SVR do not include the feature selection step. This could mean that feature selection decreases the performance, or it increases time with no significant increase in performance. However, ANNs and SVRs do not differ in every aspect. According to Sansom et al. [47] the performance of an ANN and an SVR are comparable. Moreover, the same features that are extracted for a forecast with an ANN could be used for SVRs too.

Instead of seven days ahead electricity price forecasting [47] or one year ahead load forecasting [23], this study will predict the day-ahead prices before, and very close to, market closure. This will be between 12 and 48 hours ahead. In the imbalance market, however, electricity can be bought and sold in real-time, resulting in very short term price predictions.

2.2.3. Random forests

As mentioned before, RF are less popular in forecasting electricity prices. However, a few examples are given. Lahouar & Slama [36] forecast the electricity load for one day ahead with RF, and Ludwig et al. [37] use RF in electricity price forecasting and investigate the influence of external variables. Before these researches are described, a general definition of RF is given. RF are described as the collection of (weak) decision trees consisting of independent random vectors, where each decision tree has the same number of vectors to label the output [9]. A final regression forecast is generated by averaging the outputs of the individual decision trees. Note that in RF classification, the final forecast is given by the output that is selected most in the individual decision trees. The advantage of RF over ANN and SVR is that they are able to produce the same accuracy, without the need to tune hyperparameters [36]. That does not mean that there is no optimization possible for RF. Two important hyperparameters that could be optimized are the number of individual decision trees ($n\_estimators$), and the number of features that are considered in splitting one node ($max\_features$) [36].

In the case of load forecasting [36], 24 different RF are constructed, one for every hour of the next day. All forecasts are constructed 24 hours before the given hour on delivery day. Unfortunately, this approach is only applicable on the first 12 hours of the Dutch day-ahead market, since the last 12 hours need predictions 25 to 36 hours ahead. Furthermore, many input variables are generated according electricity load, such as morning load peak at day $D-1$ or load of day $D-1$ at hour h. That makes sense in the case of load forecasting, however, less in the case of electricity price forecasting. Moreover, load forecasting is given by the TSO in the Netherlands and could be used as input for the forecasting model [63]. One noticeable model input is the month number, which has not been used in the papers described above. Due to the ease of implementation, this input will also be used in this study. Another interesting finding is that Lahouar & Slama [36] provide results that the RF outperform the ANN and SVR in terms of prediction error. All models had fixed hyperparameters, where RF was set with $n\_estimators = 100$ and $max\_features = 2.5$.

On the contrary of Lahouar & Slama, Ludwig et al. [37] have forecasted electricity prices. Two RF methods are compared in this case. The first method uses RF without external input variables. The second method, however, includes external input variables and performs feature selection as a byproduct of RF. As expected, the RF with more input variables than only price indicators outperform the RF model without external input variables. The disadvantage of many variables, however, is that the computation times increases. Therefore, a trade-off between computation time and forecast accuracy should be made. That also holds for the optimization of hyperparameters. In the described article, the hyperparameters are not optimized and the $n\_estimators$ is set to 500, for both RF. Finally, Ludwig et al. suggest RF for feature selection in other forecast approaches such as ANN and SVR.

This study distinguishes from related work in electricity price forecasting with RF by optimizing hyperparameters and by forecasting all prices from the Dutch electricity markets before closure. The extensive hyperparameter optimization should lead to better performing models than previous work. Moreover, the larger forecast horizon is required to use the forecast in a real-world trading strategy.

2.2.4. Gradient boosting

In addition to RF, GB is also a forecasting model that consists of multiple individual trees that are combined into one final forecast. However, the process of constructing the individual trees is different. While RF construct independent individual trees, the trees from GB are constructed sequentially and depend on its preceding trees. Every tree that is added to the GB should minimize the error of a fitting criterion [20]. The error of a fitting criterion for combining individual trees, also called the loss function, can be measured with several algorithms. For example, Taieb & Hyndman [55] have used the quadratic loss function, also known as the L2 or least squares. Besides the loss function, there is another algorithm that selects the right objects using a criteria function. This algorithm splits the nodes in individual trees according to a certain criteria. One of these splitting criteria is the mean squared error, which is comparable to L2. However, other algorithms are possible and could be optimized.

Taieb & Hyndman show that two other hyperparameters are also important to optimize: *learning_rate* and *n_estimators*. *learning_rate* defines the contribution of every individual tree, whereas *n_estimators* controls the number of individual trees. Therefore, these hyperparameters are correlated and control the degree of fit together. These values are set to 0.5 and 500 for the *learning_rate* and *n_estimators*, respectively, in the paper of Taieb & Hyndman.

In contrast to Taieb & Hyndman, Agrawal et al. [2] have used a hybrid method with GB and SVR to forecast day-ahead electricity prices, instead of a single GB approach. With hyperparameters *learning_rate* set to 0.09 and *n_estimators* set to 800, their proposed model outperforms SVR, recurrent ANN and RF. Another difference between the two papers is that Agrawal et al. forecast the electricity price, while Taieb & Hyndman forecast the electricity load.

Both papers have different forecast horizons than this study. The electricity load forecasting [55], for example, has a forecasting horizon of one hour ahead. Electricity price forecasting [2], on the other hand, derives 168 forecast models to predict every hour of the following week. In a comparison of used features, there are also notable differences. Taieb & Hyndman use a holiday indicator, previous demand data and generated forecasts of previous hours. Notably, Agrawal et al. use crude oil prices in addition to price and calendar indicators. Unfortunately, both papers do not apply the constructed AI approaches in a real-world market setting, making it difficult to assess their economic value. Therefore, this study makes a contribution by, among others, applying the AI models into real-world electricity markets.

2.3. Battery Energy Storage Systems

The forecasts of the electricity prices from different markets will be used in a trading model. Generally, it is not possible to buy electricity at one point in time and sell it a later moment in time, in contrast to other commodities such as gold. This is due to the fact that the physical transfer of electricity is obligatory when a PRP trades on the market [57]. Therefore, some kind of electricity storage is required to have a trading mechanism that could sell the electricity that is bought earlier. Many applications of electricity storage are investigated and developed. A few examples are pumped hydro storage, thermal energy storage, compressed air energy storage, and chemical storage such as lithium-ion batteries [27]. Due to their high energy density, discharge/charge efficiency, and long lifetimes, lithium-ion batteries lend themselves to applications in the electricity grid [13].

An example of such an application is given by Jiang & Powell [29]. They apply a BESS in the near real-time electricity market from New York, where a bid is placed one hour before closure. By using an approximate dynamic programming algorithm, they achieved results that outperformed regularly used trading policy, such as peak and off-peak determination. With a battery of 1 MW, a yearly arbitrage of more than $75k is feasible. However, taking into account the investments costs of a 1 MW battery, the net profit is negligible. That is why Jiang & Powell suggest that an arbitrage should only be exploited in select, high revenue months. Fortunately, the BESSs used in this study are not only used for an arbitrage on electricity markets, which results in lower initial investment costs and a higher profit. Moreover, the price of batteries has dropped over the last five years and is expected to drop even further [25]. However, using electricity from the BESS for other purposes also comes along with other challenges, such as the amount of electricity available. Even though that is already a constraint in trading with BESSs, external usage and production makes it more complicated. Another difference between the research of Jiang & Powell and this study is that the forecasts for electricity prices is achieved with AI, instead of dynamic programming. On top of that, both the day-ahead and the imbalance market determine the buy or sell strategy, where the former one is forecasted 12-36 hours ahead and the latter real-time ($\leq$ 15 minutes).

On the other end, Nizami et al. [43] propose a novel trading algorithm that sends excess energy from a BESS to the grid, while external usage and supply reduce and increase the electricity level, respectively. An optimal trading strategy takes into account several constraints when a BESS is also used for consumption and production. A few examples are the depth of charge (DoC), depth of discharge (DoD), number of charge/discharge cycles, and the impossibility to charge and discharge at the same time. Their paper formulates the bidding strategy as a non-linear programming problem that minimizes the cost of electricity. They consider that the supply and demand rates are fixed and known at the time of bidding. As mentioned before, both the day-ahead price and the imbalance price are not known before the bidding takes place. Therefore, the actual cost reduction resulting from this study will likely be smaller due to the forecasting error.

3. Methodology

The methodology section describes what methods are used to retrieve the results and why those methods are suitable. First, the methods of four AI approaches are explained, which also include the input variables and target. The input variables differ per target, whereas the target has two cases: the day-ahead price and the imbalance price. Then, the methodology section continues with the explanation of several trading strategies.

3.1. AI forecasting principles

This subsection contains the methods of four AI principles, namely artificial neural network (ANN), support vector regression (SVR), random forests (RF) and gradient boosting (GB). All four methods are used to forecast the same targets with the same input variables. In general, an AI approach is trained such that the input vector $\mathbf{x}$ maps to a function $f(\mathbf{x})$, which closely represents the target value y. The input vector $\mathbf{x}$ is given by all D hyperparameters as $(x_1, ..., x_D)$.

The first target discussed is the EPEX day-ahead price at time t, which is formulated as P_{da}^{t}. To ensure that the AI approaches could be applied in a real-world setting, the input variables $\mathbf{x}$ should be available at the previous day $D-1$, before noon ($t \ mod\ 24 < 12$), to forecast all 24 hourly prices ($t \in \{0, 1, .., 22, 23\} + 24 * D$) of P_{da}^{t} at day D. Due to the market closure, the price of the previous day, same hour P_{da}^{t-24} is only known for the first 12 hours. Therefore, the last 12 hours get the value 0 assigned. Other input variables are price indicators, date characteristics, generation forecasts and weather forecasts. A complete overview of the input variables that are used in the day-ahead forecast is given in Table 4.

Table 4: Overview of the input variables that are used to forecast the EPEX day-ahead market.

Description	Definition	Condition	Unit	Data source
Price previous day same hour	P_{da}^{t-24}	$t \ mod\ 24 < 12$	EUR/MWh	-
Price two days ago same hour	P_{da}^{t-48}	-	EUR/MWh	-
Price previous week same hour	P_{da}^{t-168}	-	EUR/MWh	-
Day of week	t_{week}	-	$\{0, 1, \ldots, 5, 6\}$	-
Is workday	t_{work}	-	$\{0, 1\}$	-
Month number	t_{month}	-	$\{0, 1, \ldots, 11, 12\}$	-
Forecasted solar power	p_{solar}	-	MW	[18]
Forecasted wind power	p_{wind}	-	MW	[18]
Forecasted load	p_{load}	-	MW	[18]
Forecasted temperature	T_k	-	K	[66]
Forecasted u wind speed	$v_{u,wind}$	-	m/s	[66]
Forecasted v wind speed	$v_{v,wind}$	-	m/s	[66]
Forecasted humidity	H	-	%	[66]
Forecasted cloud cover	CC	-	%	[66]
Forecasted solar radiation	R	-	W/m^2	[66]

The second target is the imbalance price for every PTU. As stated in Table 2, the selling price could be different from the buying price in regulation state 2. However, SVR, RF and GB are not able to forecast two targets. Since this only happens in one of four regulation states, an average of these two prices will be the target of the forecasting models. The imbalance price target is defined as P_{imb}. Since researchers state that all information before day-ahead market closure is included in the day-ahead prices and not in the imbalance price [32], other

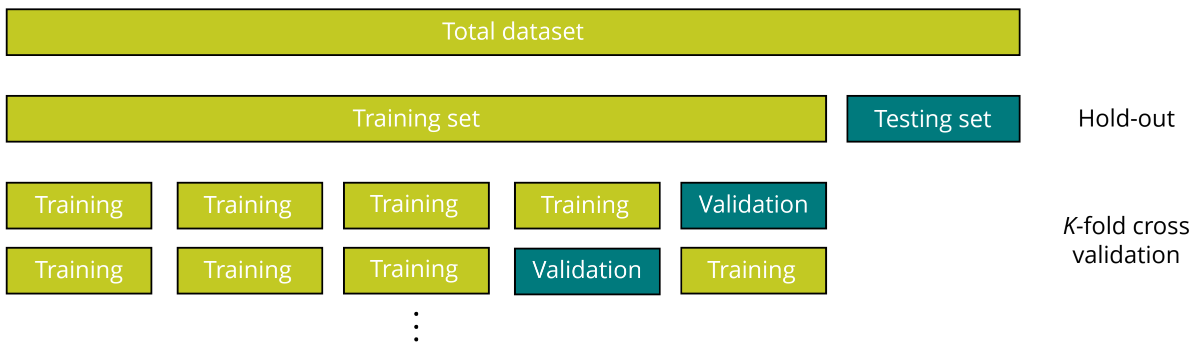


Figure 4: Splitting of the total dataset into training, testing and validation using hold-out and k-fold cross validation.

input variables and forecast horizon are required. These input variables are near real-time and updated every minute. Hence, the imbalance price P_{imb} will be forecasted during the PTU itself and accounts for the remaining minutes of the PTU. For example, when the first 5 minutes are used for data collection and forecasting, the target is forecasted for the remaining 10 minutes. The complete list of input variables is showed in Table 5. Eventually, the imbalance price is equal for the first 5 minutes and the last 10 minutes, however, the first 5 minutes can not be exploited with a trading strategy.

As discussed in Section 2, all AI principles have several hyperparameters that needs optimization. In the selection of hyperparameters, one should be careful for over-fitting. Over-fitting occurs when the selection of hyperparameters are performing too well on the training set, such that the generalisation is lost and the model performs bad for other datasets. Therefore, the total dataset is split into a training, validation, and a testing set. The results are based on the testing set, whereas the hyperparameters are optimized in the training set and validated with the validation set. The testing set is separated from the other data with the hold-out method, while retaining the date order. That means that the samples in the training and validation set take place before the samples in the testing set. The testing and validation set are split using k-fold cross validation. In this method, the dataset is split into k equal sizes. One set is selected for the validation, and the other $k-1$ are used for training the AI model. This step is repeated k times, where each set is used as the validation set once. The performance is then given by the average of k runs. An overview of the splitting process of the total dataset into training, validation and testing is given in Figure 4.

Since many hyperparameter configurations are possible for training, a selection should be made to make it computationally acceptable. This selection is made using the random search algorithm [7]. With random search, random configurations in the hyperparameter configuration space are selected during model training and validated using k-fold cross validation as described above. The advantage of random search above grid search is that it is more likely to find the best performing hyperparameters when the same amount of hyperparameter configurations are used [7]. However, the assumption that not all hyperparameters are equally important needs to hold.

Most AI approaches rely on the fact that the input variables are normalized and distributed around the mean. That is why all input variables are transformed using the following formula

$$z_{ti} = \frac{\mathbf{x}_{ti} - \mu_i}{\sigma_i} \tag{1}$$

Table 5: Overview of the input variables that are used to forecast the real-time imbalance market. T_s in this case is the number of minutes from one PTU that is used to derive the input variables.

Description	Definition	Unit	Data source
Upward IGCC volume	$\sum_{t=1}^{T_s} IGCC_{up}^t$	MW	[59]
Downward IGCC volume	$\sum_{t=1}^{T_s} IGCC_{down}^t$	MW	[59]
Upward regulating volume	$\sum_{t=1}^{T_s} REG_{up}^t$	MW	[59]
Downward regulating volume	$\sum_{t=1}^{T_s} REG_{down}^t$	MW	[59]
Upward reserve volume	$\sum_{t=1}^{T_s} RES_{up}^t$	MW	[59]
Downward reserve volume	$\sum_{t=1}^{T_s} RES_{down}^t$	MW	[59]
Emergency activation	$max(EMER^1, \dots, EMER^{T_s})$	$\{0, 1\}$	[59]
Highest price upward	$max(P_{up}^1, \dots, P_{up}^{T_s})$	EUR	[59]
Lowest price downward	$min(P_{down}^1, \dots, P_{down}^{T_s})$	EUR	[59]
Mid price	P_{mid}	EUR	[59]

where $\mathbf{x}_{ti}$ is the i-th input variable at sample t, μ_i is the mean, and calculated with

$$\mu_i = \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{ti} \tag{2}$$

with T, the number of samples. σ_i is the standard deviation, with

$$\sigma_i = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\mathbf{x}_{ti} - \mu_i)^2} \tag{3}$$

Furthermore, not all input variables are equally important, and too many variables could decrease the accuracy in the testing set due to overfitting. However, this study does not emphasize the input selection process. Therefore, only the inputs with the same values for every sample will be removed from the input selection.

3.1.1. Artificial neural network

Section 2.2.1 explained that an ANN can have multiple shapes, based on their architecture. This study focuses on the multi-layer perceptron (MLP), with N hidden layers and one target output. The total number of layers M is then described as the number of hidden layers and the target output. Formally, $M = N + 1$. An example of an ANN with one hidden layer and one output node is given in Figure 3.

For every layer in the MLP, except for the input layer, the nodes have an activation a. For the first layer, a in node j is formalised as

$$a_{1,j} = \sum_{i=1}^{D} w_{j,i}x_i + w_{j,0} \tag{4}$$

where $w_{j,i}$ is the weight between nodes j and i, x_i is the input value, and $w_{j,0}$ gives the bias. Given the activation of a node, the output of the node is provided by activation function h

$$z_{m,j} = h(a_{m,j}) \tag{5}$$

where m is the layer number. For every layer where $m > 1$, the output layer depends on the output of the nodes from the previous layer, instead of the input nodes. In the example of Figure 3, the activation of layer 2, which is also the activation of the output, is given by

$$a_{2,j} = \sum_{i=1}^{D} w_{j,i} z_{1,i} + w_{j,m-1} \tag{6}$$

Several activation functions are possible in an ANN. Two of the most popular are described and used in this study. The first activation function is the hyperbolic tangent function and is given by

$$tanh(a_{m,j}) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \tag{7}$$

The second activation function is the rectified linear unit function, which outputs zero if the activation is smaller than zero and the activation itself otherwise. More formally

$$ReLu(a_{m,j}) = \begin{cases} 0, & \text{if } a_{m,j} \leq 0 \\ a_{m,j}, & \text{if } a_{m,j} > 0 \end{cases} \tag{8}$$

The goal of an ANN is minimizing the forecasting error on a given validation set. This is achieved by assigning an optimal weight vector $\mathbf{w}$ to minimize the loss function. The loss function is based on a stochastic gradient descent and in this case given by

$$L(\mathbf{w}) = \sum_{k=1}^{K} \{f(\mathbf{x}_k, \mathbf{w}) - y_k\}^2 \tag{9}$$

where K is the number of samples in the validation set, $f(\mathbf{x}, \mathbf{w})$ is the activation function for the output node, and y is the actual price. Since the loss function is computationally expensive for large datasets, a stochastic gradient descent based optimizer is used, called Adam [31]. The pseudo-code of Adam is shown in Algorithm 1. Since the weights from the initial weights vector are assigned randomly, a small difference in forecasting error is possible between several runs.

3.1.2. Support vector regression

In addition to the ANN, SVR also construct its regression estimation by minimization. To formalize the SVR, the simple linear case is taken as a starting reference. The linear form is described as

$$f(\mathbf{x}) = \langle \mathbf{x} \bullet w \rangle + b, \text{ with } w \in \mathbf{x} \tag{10}$$

where $\langle \mathbf{x} \bullet w \rangle$ denotes the dot product of $\mathbf{x}$ and w. The optimization is achieved by minimizing $\langle w \bullet w \rangle$, while keeping the error below the allowed error ϵ

$$\begin{aligned} &\text{minimize } \langle w \bullet w \rangle \\ &\text{subject to } \begin{cases} y_t - \langle w \bullet \mathbf{x}_t \rangle - b & \leq \epsilon \\ \langle w \bullet \mathbf{x}_t \rangle + b - y_t & \leq \epsilon \end{cases} \end{aligned} \tag{11}$$

With a fixed allowed error ϵ, there is assumed that all $f(\mathbf{x}, \alpha)$ fall within this error margin. Since that assumption does not always hold, two additive error margins are introduced, namely,

Algorithm 1: *Adam* algorithm for stochastic gradient descent optimization, adapted from Kingma & Ba [31]. All operations on vectors are element-wise. g_t^2 denotes the element wise square $g_t \odot g_t$. β_1 and β_2 to the power of t is represented as β_1^t and β_2^t, respectively.

Input : α: Penalty parameter
$\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
$L(\mathbf{w})$: Loss function with parameters $\mathbf{w}$
$\mathbf{w}_0$: Initial weights vector
$m_0 \leftarrow 0$: Initialize first moment vector
$v_0 \leftarrow 0$: Initialize second moment vector
$t \leftarrow 0$: Initialize timestamp

Output: optimized weight vector $\mathbf{w}$

while $\mathbf{w}_t$ *not converged* **do**
    $t \leftarrow t + 1$ (set updated timestamp)
    $g_t \leftarrow \nabla_0 L_t(\mathbf{w}_{t-1})$ (set updated slope of loss function)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (set updated bias first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (set updated bias second moment estimate)
    $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (set updated first moment estimate corrected for bias)
    $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (set updated second moment estimate corrected for bias)
    $\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (set updated parameters)
return $\mathbf{w}_t$ (return optimized weight vector)

ξ and ξ^*. These error margins are not fixed and are optimized with penalty parameter C by the SVR as follows

$$\begin{aligned} &\text{minimize } \langle w \bullet w \rangle + C \sum_{t=1}^{T} (\xi_t + \xi_t^*) \\ &\text{subject to } \begin{cases} y_t - \langle w \bullet \mathbf{x}_t \rangle - b & \leq \epsilon + \xi_t \\ \langle w \bullet \mathbf{x}_t \rangle + b - y_t & \leq \epsilon + \xi_t^* \\ \xi_t, \xi_t^* & \geq 0 \end{cases} \end{aligned} \tag{12}$$

This objective function with constraints can be rewritten as a Lagrange function, by introducing Lagrange multiplier variables $\alpha, \alpha^*, \eta, \eta^*$

$$\begin{aligned} L := &\langle w \bullet w \rangle + C \sum_{t=1}^{T} (\xi_t + \xi_t^*) + \sum_{t=1}^{T} (\eta_t \xi_t + \eta_t^* \xi_t^*) \\ &- \sum_{t=1}^{T} \alpha_t (\epsilon + \xi_t - y_t + \langle w \bullet \mathbf{x}_t \rangle + b) \\ &- \sum_{t=1}^{T} \alpha_t^* (\epsilon + \xi_t^* + y_t - \langle w \bullet \mathbf{x}_t \rangle - b) \end{aligned} \tag{13}$$

Due to the constraints, substitution of the partial variables while optimizing, and Equation 13, w can be rewritten as

$$w = \sum_{t=1}^{T} (a_t - a_t^*) \mathbf{x}_t \tag{14}$$

which could be substituted in Equation 10 as

$$f(x) = \sum_{t=1}^{T} (a_t - a_t^*) \langle \mathbf{x}_t \bullet \mathbf{x} \rangle + b \tag{15}$$

Note that the SVR started with a linear expression. Especially AI approaches are able to handle non-linearity. That is why the $\langle \mathbf{x}_t \bullet \mathbf{x} \rangle$ expression is transformed into a non-linear expression using the kernel function. For non-linear SVR

$$\langle \mathbf{x}_t \bullet \mathbf{x} \rangle = k(\mathbf{x}_t, \mathbf{x}) \tag{16}$$

Fore a more in-depth constructing of the formulas, examine the research of Smola & Schölkopf [49]. In this study, two kernel function are applied in the SVR and described by Chang & Ling [12]. The first kernel is the radial basis function, also known as rbf, which takes the form

$$k(\mathbf{x}_t, \mathbf{x}) = e^{-\gamma \|\mathbf{x}_t - \mathbf{x}\|^2} \tag{17}$$

where γ is the kernel coefficient that could be optimized in the hyperparameter setting. The second kernel function is the sigmoid kernel

$$k(\mathbf{x}_t, \mathbf{x}) = tanh(\gamma \langle \mathbf{x}_t \bullet \mathbf{x} \rangle + r) \tag{18}$$

where r is also a kernel coefficient that needs optimization.

3.1.3. Random forests

The following AI approach is a form of decision tree regression, namely RF regressor. First, the decision tree regression methodology is explained. Given the example in Figure 5, every node has a selection criteria based on input $\mathbf{x}$. The top node, also called the root node, contains all data points from the training set. Every decision node is split into two sub-nodes with a splitting criteria on the input $\mathbf{x}$. The splitting criteria in every decision node uses the impurity, which in this study is set to the mean absolute error (MAE) and mean squared error (MSE). More formally, the MAE in node n_1 from Figure 5 is defined as

$$\frac{1}{N_1} \sum_{i=i}^{N_1} |y_i - \mu| \tag{19}$$

and MSE as

$$\frac{1}{N_1} \sum_{i=i}^{N_1} (y_i - \mu)^2 \tag{20}$$

where N_1 is the number of instances in node n_1, y_i is the value of an instance, and μ is given by

$$\frac{1}{N_1} \sum_{i=i}^{N_1} y_i \tag{21}$$

The split is set to the variable and criteria that achieves the highest information gain, which is defined as the impurity of the parent node minus the impurity of the child nodes. For Figure 5, the information gain for the root node is stated as

$$I(D_0) - (\frac{N_1}{N_0} I(D_1) + \frac{N_2}{N_0} I(D_2)) \tag{22}$$

where $I(D_0)$ is the impurity function, given by MAE or MSE, for the dataset in the root node. Eventually, all nodes are split using the information gain and the forecast is given by the average of all instances in the leaf node.

RF on the other hand, is an extension of the decision tree regressor. Here, multiple decision trees are built that all have their own forecast. The final forecast is then given by the average of all individual decision trees. Therefore, the number of decision trees $n\_estimators$ is an important

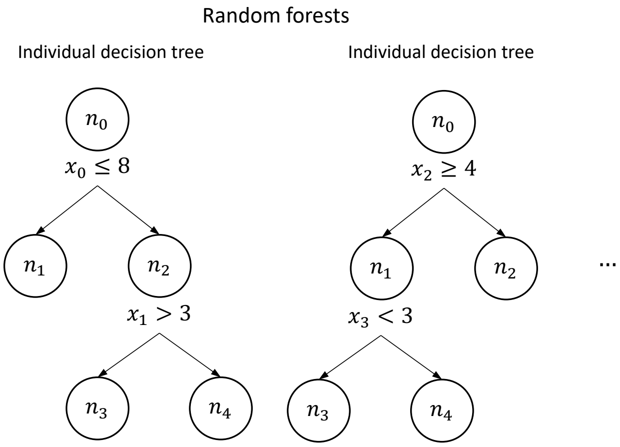


Figure 5: Example of a random forests regressor with 2 individual decision trees, 5 nodes, one root node, 2 decision nodes, and 3 leaf nodes per decision tree. The trees have a depth of 2.

hyperparameter. The difference in the individual tree construction is that each tree takes a subset of the training set, leading to different splitting criteria and different trees. However, the hyperparameter *bootstrap* could ensure that the whole dataset is used when constructing every single decision tree. Then, the only difference in tree construction could be made by randomly selecting features for a split when the maximum number of features $max\_features$ is smaller than the total number of features.

Normally, a decision tree adds new nodes until every leaf node only holds equal values. This could lead to a large tree that is time consuming to construct and sensitive to overfitting. Therefore, several stopping criteria are added in the RF regressor. Two example are the minimum number of samples that is required to split a decision node $min\_samples\_split$, and the minimum number of samples that is required to be at a leaf node $min\_samples\_leaf$. These hyperparameters are optimized in this study using the random search algorithm.

3.1.4. Gradient boosting

In addition to RF, GB also constructs its forecast using trees. However, instead of bagging, where trees are constructed individually, GB uses boosting, where trees are dependent on their predecessor. The approach starts with a weak individual tree that only contains a constant value, called the base learner. The base learner is defined as

$$F_0(\mathbf{x}) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma) \tag{23}$$

where n is the number of samples in the training set, and $L(y_i, \gamma)$ is the loss function with y_i as the actual value and γ as the base learner forecast value. In this study, two loss functions are used to get the best performing GB model. The loss function are the MAE and the MSE, given by Equations 19 and 20, respectively. After the first base learner is defined, the algorithm adds new decision trees until the maximum number of trees $n\_estimators$ is achieved. For every tree, Equations 24, 25 and 26 are executed. Given the forecast of predecessor trees, a new tree starts with calculating pseudo-residuals, which are constructed from the partial derivative of the loss function

$$r_{im} = -\left[\frac{\partial L(y_i, F_{m-1}(\mathbf{x_i}))}{\partial F_{m-1}(\mathbf{x_i})}\right] \text{ for } i = \{1, \dots, n\} \tag{24}$$

where m is the index for the three that is constructed. After that, an individual decision tree algorithm, such as the one given in Section 3.1.3, is trained to forecast the pseudo-residuals r.

Note that this setup is different, since decision trees generally forecast the y-values. The trained single tree contains J_m leaves, where each leaf node is addressed as R_{jm} with $j \in \{1, \dots, J_m\}$. Then for each node $j \in \{1, \dots, J_m\}$, the forecast value is computed with

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{\mathbf{x_i} \in R_{ij}}^{J_m} L(y_i, F_{m-1}(\mathbf{x_i}) + \gamma) \tag{25}$$

When all γ_{jm} forecasts are calculated, the forecast of the m-th tree is given by

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \sum_{j=1}^{J_m} \{\gamma_{jm} \text{ if } \mathbf{x} \in R_{jm}\} \tag{26}$$

with learning rate ν as the contribution of a single tree in the total forecast. Finally, the GB regression forecast is given by the $n\_estimators$-th tree following Equation 26.

3.1.5. Significance

Since the best performing AI approach will be used in the trading strategies, a significance test is run to select the best performing forecaster on the imbalance market. A significance test is only required for the imbalance markets due to the different T_s setups that are used. With a changing T_s, the input variables, and therefore the datasets, changes. The post-hoc Nemenyi test [16] is suitable for this comparison. First, the AI approaches get a rank assigned for each dataset, which is based on their performance. In case of a tie, an average rank is given. This procedure is based on the Friedman test [16]. Once the ranks are assigned, the critical distance (CD) is calculated with the following equation

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{27}$$

where k is the number of models to compare, N is the number of datasets, i.e., number of different T_s, and q_α is based on the Studentized range statistic divided by $\sqrt{2}$ [16], which depends on the significance level α. If two models differ at least the CD from each other, it is safe to say that the models are significantly different.

3.2. Trading strategies

There are many trading strategies possible with a BESS. In this subsection, several are discussed and formalized. The trading strategies are expressed in EUR, which refers to the energy bill over a period of time, without taxes and subsidies. A negative outcome means that the user should pay that amount to the energy supplier. It is assumed that the electricity consumption minus the electricity production through solar, further referred to as net consumption C_n in MWh, is known before the day-ahead market closure. Normally, there is a deviation between the forecast and the actual net consumption. However, all trading strategies are constructed with the same assumption, which makes the comparison adequate. Note that the assumption is only required when the electricity is traded on the day-ahead market. That is why the the trading strategies with a large share of imbalance trading are more accurate than the trading strategies with a small share of imbalance trading. Another assumption is that the battery has zero energy loss. Thus, the electricity that is sent that to battery can be fully reused in the future.

3.2.1. Baseline strategies

First, three baseline methods are given. The baseline method B_1 is the electricity bill if everything is bought from the day-ahead market, without the battery. More formally, the first baseline method for one day is

$$B_1 = \sum_{t=0}^{T} P_{da}^t \cdot -C_n^t \tag{28}$$

where T is the last cumulative hour of the final trading day. The second baseline B_2 is comparable to B_1, however, the net consumption is calculated with the imbalance price P_{imb}

$$B_2 = \sum_{t=1}^{T} P_{imb}^t \cdot -C_n^t \tag{29}$$

where T is 96, the number of PTUs in one day. Note that P_{imb}^t is different from the imbalance price that is forecasted, since P_{imb}^t is different for buying and selling at regulation state 2. P_{imb}^t is therefore dependent on the buy or sell signal.

While the two baseline methods do not include the battery, the third baseline method and the remaining trading strategies do include the battery. As such, the third baseline method includes the battery without a reaction to market movements. The battery flow is given by the optimization of self-consumption. If the net consumption is negative, i.e., the solar generation is larger than the consumption, the electricity is stored in the battery as long as the maximum capacity BT_{doc} is not achieved. The maximum electricity that could be transferred to the battery until the maximum capacity is achieved is defined as

$$BT_{max,c} = BT_{doc} - BT_{soc} \tag{30}$$

where BT_{soc} is the electricity that is stored in the battery. When the net consumption is positive, i.e., the consumption is larger than the solar generation, the electricity is used from the battery as long as the minimum capacity BT_{dod} is not achieved. The maximum electricity that could be transferred from the battery until the minimum capacity is achieved is then given by

$$BT_{max,d} = BT_{soc} - BT_{dod} \tag{31}$$

Note that the battery also has a maximum power flow per PTU $BT_{max,p}$, which means that the battery flow BT in B_3 is formalized as

$$BT = \begin{cases} -min(-C_n, BT_{max,c}, BT_{max,p}), & \text{if } C_n \leq 0 \\ min(C_n, BT_{max,d}, BT_{max,p}), & \text{if } C_n > 0 \end{cases} \tag{32}$$

where a negative BT means that the battery is charged, and conversely. Then the third baseline method is given by

$$B_3 = \sum_{t=0}^{T} P_{da}^t \cdot (-C_n^t + BT^t) \tag{33}$$

Since P_{da}^t is only defined for 24 values $\{0, \dots, 23\}$ per day, the price set is enlarged to 96 values $\{1, \dots, 96\}$ by adding 3 values per single price. Those added prices are equal to the last known day-ahead price, i.e., for $t_1 \in \{1, 2, 3\}$, $P_{da}^{t_1} = P_{da}^{t=0}$.

3.2.2. State of the art strategies

Generally, all other trading strategies are formalized as

$$G = \sum_{t_1 \in S_1} P_{da}^{t_1} \cdot -C_n^{t_1} + \sum_{t_2 \in S_2} P_{da}^{t_2} \cdot BT^{t_2} + \sum_{t_3 \in S_3} P_{imb}^{t_3} \cdot -C_n^{t_3} + \sum_{t_4 \in S_4} P_{imb}^{t_4} \cdot BT^{t_4} \tag{34}$$

where $S_1, S_2, S_3, S_4 \in S$ and S is the set of PTUs in one day $\{1, \ldots, 96\}$, S_1 is the set of PTUs to trade the net consumption on the day-ahead market, S_2 is the set of PTUs to trade the battery BT on the day-ahead market, S_3 is the set of PTUs to trade the net consumption on the imbalance market, S_4 is the set of PTUs to trade the battery BT on the imbalance market, with $S_1 + S_3 = S, S_1 \cap S_3 = \varnothing, S_2 \cap S_4 = \varnothing$. The trading strategies are then defined by selecting the PTU sets S_1, S_2, S_3, S_4 and the battery electricity flow BT.

The first strategy G_1 only considers the day-ahead market. The net consumption is settled on the day-ahead market such as Equation 28. However, the difference is made by adding the battery flow BT on the day-ahead market, where BT is dependent on the forecasted day-ahead price F_{da}. The battery is charged if the forecasted day-ahead price for a certain PTU is smaller or equal to X_{da}^c times the daily average of the forecasted day-ahead prices F_{da}^{μ}, where $0 < X_{da}^c < 1$. The battery is discharged when the forecasted day-ahead price is larger or equal to a threshold with $X_{da}^d > 1$. BT, S_1, S_2, S_3, S_4 are formalized for G_1 as

$$\begin{aligned} &BT = \begin{cases} -min(BT_{max,c}, BT_{max,p}), & \text{if } F_{da} \leq X_{da}^c \cdot F_{da}^{\mu} \\ min(BT_{max,d}, BT_{max,p}), & \text{if } F_{da} \geq X_{da}^d \cdot F_{da}^{\mu} \\ 0, & \text{if } X_{da}^c \cdot F_{da}^{\mu} < F_{da} < X_{da}^d \cdot F_{da}^{\mu} \end{cases} \\ &S_1 = S, S_2 = S, S_3 = \varnothing, S_4 = \varnothing \end{aligned} \tag{35}$$

The second state of the art strategy G_2, also settles its consumption on the day-ahead market such as $G1$. However, the battery flow BT is settled on the imbalance market instead. Since the actual day-ahead price is known before the imbalance forecasting, the forecasted imbalance price F_{imb} is compared to the day-ahead price. Another difference is that on the imbalance market, a setup time is required to collect the information for forecasting. Therefore, the $BT_{max,p}$ should be compensated for the setup time. This is achieved by multiplying $BT_{max,p}$ with the compensation factor, which depends on the setup time in minutes T_s and the total number of minutes in one PTU, 15. Furthermore, there is a three minute delay between the occurrence and the publication of the input data. Therefore, three minutes are added to T_s in order to calculate the battery flow for real-world scenarios. With the same constraints for X_{imb}^c and X_{imb}^d as X_{da}^c and X_{da}^d, respectively, G_2 is given by

$$\begin{aligned} &BT = \begin{cases} -min(BT_{max,c}, BT_{max,p} \cdot \frac{15-(T_s+3)}{15}), & \text{if } F_{imb} \leq X_{imb}^c \cdot P_{da} \\ min(BT_{max,d}, BT_{max,p} \cdot \frac{15-(T_s+3)}{15}), & \text{if } F_{imb} \geq X_{imb}^d \cdot P_{da} \\ 0, & \text{if } X_{imb}^c \cdot P_{da} < F_{imb} < X_{imb}^d \cdot P_{da} \end{cases} \\ &S_1 = S, S_2 = \varnothing, S_3 = \varnothing, S_4 = S \end{aligned} \tag{36}$$

The final strategy, G_3, is a combination of Equation 35 & 36. Here, the battery is charged in the day-ahead market for the X_{da}^{min} lowest forecasted day-ahead prices, and discharged for the X_{da}^{max} highest forecasted day-ahead prices. The set containing the PTUs with the X_{da}^{min} lowest prices will be further referred to as S_l, while the set with the X_{da}^{max} highest prices will be written as S_h. All residual hours are settled on the imbalance market. Formally, $G3$ is described by

$$\begin{aligned}
BT &= \begin{cases} BT \text{ from Eq. 35}, & \text{if } F_{da} \in S_l, S_h \\ BT \text{ from Eq. 36}, & \text{if } F_{da} \notin S_l, S_h \end{cases} \\
S_1 &= S, S_3 = \varnothing \\
S_2 &= \begin{cases} S, & \text{if } F_{da} \in S_l, S_h \\ \varnothing, & \text{if } F_{da} \notin S_l, S_h \end{cases}, S_4 = \begin{cases} \varnothing, & \text{if } F_{da} \in S_l, S_h \\ S, & \text{if } F_{da} \notin S_l, S_h \end{cases}
\end{aligned} \tag{37}$$

In addition to the profit in EUR that is used to measure the performance of a trading strategy, the number of cycles in a battery N_{BT} is also an important measure. This is due to the fact that a battery does not have an unlimited lifespan. In general, the lifespan of the battery is limited to a fixed number of cycles. The number of cycles is defined as follows

$$N_{BT} = \frac{\sum_{t_2 \in S_2} |BT^{t_2}| + \sum_{t_4 \in S_4} |BT^{t_4}|}{BT_{doc} - BT_{dod}} \tag{38}$$

Another interesting result is the profit of the trading machine divided by the number of cycles. More formally,

$$Y_{PN} = \frac{Profit}{N_{BT}} \tag{39}$$

where profit is defined as the electricity costs of the trading strategy, minus the electricity costs of B_1. In conclusion, the performance of the trading strategy performance is defined as the profit in EUR, the number of battery cycles and their ratio. The findings of the trading strategies and AI approaches are shown in the following section.

3.3. Implementation within organizations

Given the forecasts and trading strategies from the previous subsections, a live implementation within an organization is the next step. Logically, the best performing trading strategy is used for implementation. However, a preference for ease of use and costs within an organization could lead to another selection of strategy. For example, trading strategy G_3 requires both the input data of the day-ahead market and the imbalance market. This could lead to a connection with multiple data providers who may charge fees for their data. Trading strategy G_2 could then be less expensive within the live implementation.

Nonetheless is the selection of the trading strategy a small step towards the live implementation of the trading strategy within organizations. The main issue is to provide the battery a charge, discharge or hold command within every PTU, such that the electricity is bought and sold at the correct time. Commands are given to the battery by writing correct values in the registers of the inverter. The values are sent to the inverter via a communication standard called modbus serial. Modbus serial is based on the interaction between master and slave devices, where the master device is reading and writing values from and to the slave device, respectively [26]. In this study, the inverter is the slave device and the master device is a smart energy meter with an ability to connect to the internet. The master and slave devices are connected with RS-485.

Since the energy meter is in control of the inverter, the energy meter should be controlled remotely. This is achieved by custom software on the energy meter and a connection to an internet of things (IoT) platform. The values from the inverter are then sent via the energy meter to the platform using the message queue telemetry transport (MQTT) protocol. MQTT is a publish-subscribe protocol, where updates from the energy meter are received by the IoT platform instantaneously. The advantages of MQTT are low network overhead and low power

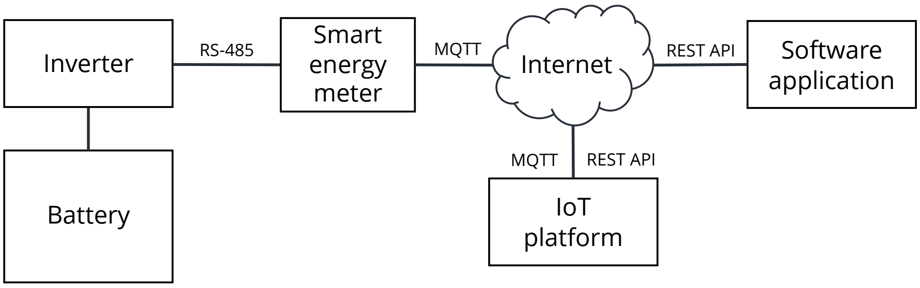


Figure 6: Architecture and communication protocols of battery and devices setup for the implementation of the trading strategy within an organization.

consumption [51]. Moreover, values that can be changed on the platform, i.e., actuators, are also sent to the energy meter and change the value in inverter registers. In turn, values from the platform can be controlled with any kind of application that allows an application programming interface (API) and obeys the representational state transfer (REST) constraints. An overview of these devices and their connections is shown in Figure 6.

With the physical architecture that allows the battery to be controlled remotely, another software application is required to send signals to the battery. This application contains the considerations from this study and converts it to a real-world application. Generally, the AI model receives real-time input data and transforms it into a forecast, which is used in an advice for the battery. Several steps are needed to apply this at a real-world battery. First, the input data for the AI models should be collected regularly. Then, a forecast is made with the AI model and transformed to an advice with the trading strategy. The frequency of an advice depends on the market. For the day-ahead market, 24 forecasts with an advice are made once every day. However, the advice is made every PTU for the imbalance market. An advice from the market could not always be transformed into an action for the battery, due to the capacity of the battery. Therefore, the application needs to check whether BT_{soc} is in range with BT_{doc} and BT_{dod}. If all constraints are met, the advice can be put into action. Organizations with multiple batteries could separate this final step from the forecasting step. The pseudo code of the written software application to steer the battery is given in Algorithm 2.

Algorithm 2: Pseudo code for the real-world implementation of the trading strategy. The results from the next section are based on this pseudo code.

Input : t: frequency in minutes to run the algorithm
Output: battery steering that keeps responding to market signals

```
train AI model on selected historical data
for every t minutes until application is terminated do
    get input data for imbalance market
    get BT_soc and charging mode from battery
    if BT_soc out of range or time of input data outside trading time then
        stop with charging and discharging
        continue to end of loop
    if time of input data inside trading time and not traded in current PTU yet then
        forecast imbalance price with AI model
        get P_da
        convert forecast to advice with trading model
        set battery charging mode equal to advice
```

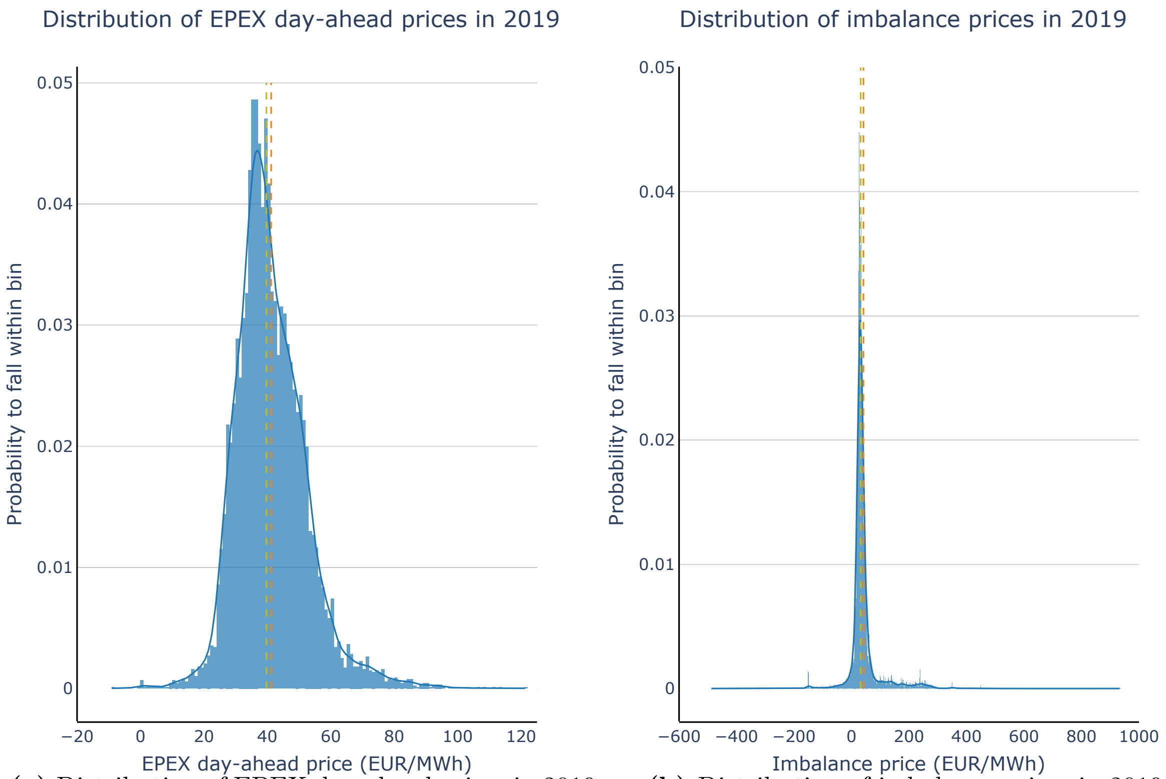
The first step in the pseudo code uses a selection of historical data. This selection is important for the performance of trading strategies within organizations. Moreover, the frequency in which AI models are updated with new training data could also improve the performance of trading strategies when the live implementation is used for a long time period. Selecting the frequency to update the AI model with new training is out of range in this study, and also less important due to the small window of live trading. Therefore, the training data is set for a fixed time period without an update. However, it is important to take this into account when the implementation is used for a longer time span. Two other notable rows in the pseudo code are the if statements. The first if statement in the pseudo code ensures that the battery stops with charging and discharging if the conditions for trading are not met. The second if statement in the pseudo code ensures that the most time consuming steps are only executed when necessary. This leads to a fast software application, even with high frequencies.

4. Findings

This section presents the used data, the experimental setup, and the achieved results. Even though Section 3 already discussed some data with their sources, this chapter expands the data understanding with graphs and figures. Furthermore, the data cleaning steps are also considered. The configurations for the AI approaches are shown in the experimental setup. This includes, among others, the hyperparameter configuration space and the number of iterations to search for the best performing hyperparameters. Finally, the results of several AI approaches, setup configurations and trading strategies are given.

4.1. Data

The first target, EPEX day-ahead, is provided by Entsoe [18]. Although the day-ahead price fluctuates over time, 2019 was a rather typical year for electricity prices. Therefore, the data visualization of 2019 also represent other years. The distribution is given in Figure 7a. As can be seen, the prices follow a right-skewed distribution. The imbalance prices, the second target, is retrieved from TenneT [62]. These prices are also right-skewed, as can be seen in Figure 7b. Table 6 shows the mean, median, minimum value and maximum value of both the day-ahead market and the imbalance market. Notably, the mean of the two markets are close to each other in 2019, while the minimum and maximum differ significantly. Furthermore, the median shows that more than half of the values from the imbalance market are smaller than the mean of the day-ahead market. However, the mean of the imbalance market is slightly higher than the mean of the day-ahead market. This can be explained by the significant amount of values above 100 EUR/MWh.



(a) Distribution of EPEX day-ahead prices in 2019 **(b)** Distribution of imbalance prices in 2019

Figure 7: Distribution of the EPEX day-ahead (a) and the imbalance market (b) for 2019. The y-axis shows the probability that a price falls within that bin. Both distributions are right-skewed, since the mean (orange line) is more to the right than the median (yellow line).

Table 6: An overview of the statistics for the EPEX day-ahead market and imbalance market in 2019. The AI approaches are trained on this dataset.

Statistical measure	EPEX day-ahead	Imbalance	Unit
Mean	41.19	41.47	EUR/MWh
Median	39.7	31.66	EUR/MWh
Minimum	-9.02	-487.65	EUR/MWh
Maximum	121.46	936.12	EUR/MWh

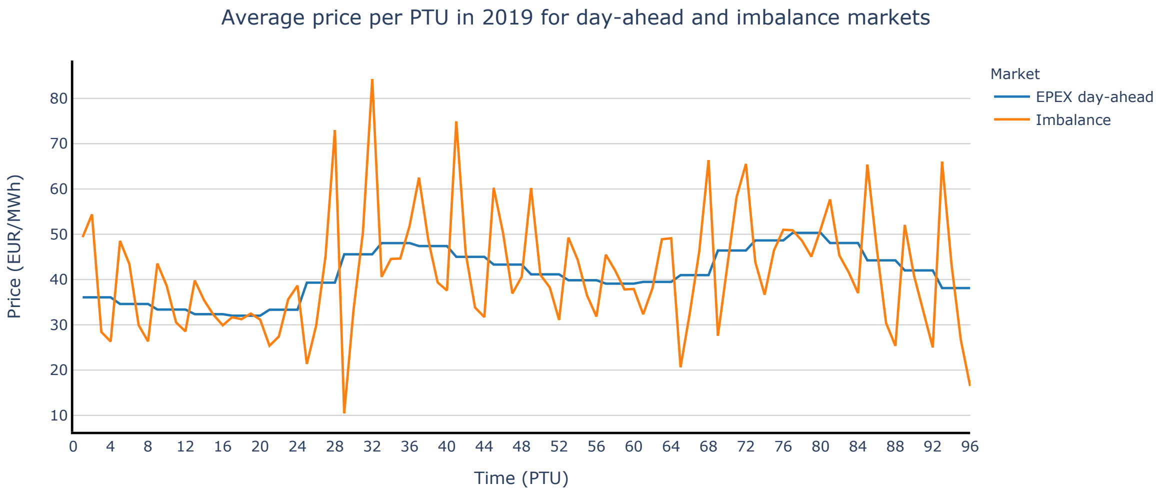


Figure 8: The average price per PTU for the EPEX day-ahead market and the imbalance market in 2019.

On average, the day-ahead and imbalance prices fluctuate during the day as shown in Figure 8. The highest prices in the day-ahead market are achieved in the morning hours, between 7 AM and 10 AM, and in the evening hours, between 6 PM and 9 PM. This could be explained by the average working schedule of people. The imbalance market on the other hand, shows a different pattern. While the imbalance prices fluctuate around the day-ahead prices, the price often jumps from a large value to a small value and reversed after one PTU. Prices from the previous PTU are therefore less valuable in the imbalance market than they would be in the day-ahead market. Furthermore, the prices from the previous PTU can not be used in the imbalance market, since they are not known by then.

Other pricing indicators that are available in the day-ahead forecasting are shown in Table 4. The forecasted load p_{load}, forecasted wind power p_{wind}, and forecasted solar power p_{solar}, are retrieved from Entsoe [18]. p_{load} is the total load per hour, forecasted by the TSO. The forecast is submitted two hours before market closure. Only when a $\geq 10\%$ change in forecast is realized, the forecast is updated. p_{wind} and p_{solar} are the forecasted wind and solar power, respectively, submitted by the TSO. While the p_{load} is always known before closure, p_{wind} and p_{solar} are sometimes only known after market closure. This is due to the latest submission time of 6 PM, six hours after market closure. However, the forecast is often submitted before market closure. Furthermore, the intraday forecast, which is submitted fourteen hours after the day-ahead forecast, is often equal to the day-ahead forecast. Therefore, it is safe to say that the submission six hours after market closure closely represents the forecast that could be available before market closure. Note that a live implementation requires another data provider that submits the data before market closure. The forecasted wind is given for both offshore and inshore. In this model, p_{wind} is defined as the sum of the offshore and inshore. In addition, p_{load}, p_{wind} and p_{solar} are forecasted for one PTU. Since the day-ahead price only differs per hour, the quarterly values are summed into one hour. Finally, the few missing values for p_{load} and p_{solar} are filled using time interpolation.

Other input variables include weather forecasts from the KNMI [33], which is the Dutch national weather forecasting provider. These weather forecasts are made 48 hours before the occurrence. An average is taken from 90,000 data points, representing all locations in the Netherlands and the North Sea. The forecast is made every six hours. Since the day-ahead market is forecasted for every hour, the missing hours of weather forecasts are set to the last known hour, e.g., for $t \in \{1, 2, 3, 4, 5\}, F^t = F^{t=0}$.

The forecasting of imbalance prices uses other input variables than the day-ahead prices. An overview is given by Table 5. Since the imbalance price is settled after the PTU is passed, real-time information from other electricity markets could provide more insights to get a forecast within the PTU itself. Four of these markets are discussed. The real-time information of these markets has a three minute delay for processing and presentation, and contains data for every minute. The first market, IGCC, states the amount of electricity power that is retrieved from or send to external parties other than TenneT, with their upward or downward quantity respectively. The regulating market shows the regulating capacity that is activated by the national load frequency control. Both upward and downward adjustments are presented. Thirdly, there is the reserve market. There is always an amount of electricity available in the reserve market. For this setting, the amount that is actually send to or taken from the grid is shown in the reserve capacity. Finally, there is the emergency activation, which states whether the emergency capacity is activated. This market is upwards only and used to prevent a power shortage. In addition to the quantities in imbalance forecasting, the corresponding prices are used too. The highest price upward is the highest bid price from the the above mentioned markets excluding the IGCC, while the lowest price downward is the lowest bid price. The mid price P_{mid} is explained by Table 2. Since the highest price upward and lowest price downward could be empty in the dataset, they are filled with -10 and 125, the minimum and maximum of the day-ahead price in 2019, respectively. Note that these prices should not be filled with zero, since those prices could actually be zero. Therefore, a zero could provide false information. Filling the prices with values that would be less achieved lead to a more accurate forecast.

4.2. Input variables analysis

This subsection analyses the input variables by finding autocorrelations on the target variable and correlations between the input variables and the target variables. Before the autocorrelations of the target variables are explained, an introduction to correlation is given. Correlation defines the relation between two variables and is formalized as

$$\rho(X, Y) = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \tag{40}$$

with the standard deviation σ defined in Equation 3, and the covariance $cov(X, Y)$ as

$$cov(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] \tag{41}$$

where μ is defined as the mean with Equation 21 and $\mathbb{E}$ as the expected value, which is in this case also defined as the mean. Then, the autocorrelation can be defined as

$$ACF(y_t, y_{t+1}) = \rho(y_t, y_{t+1}) \tag{42}$$

Informally, the autocorrelation defines what the correlation is between the target and its lagged values, e.g., the correlation between target y_1 and target y_3. The autocorrelation of the day-ahead market is shown in Figure 9.

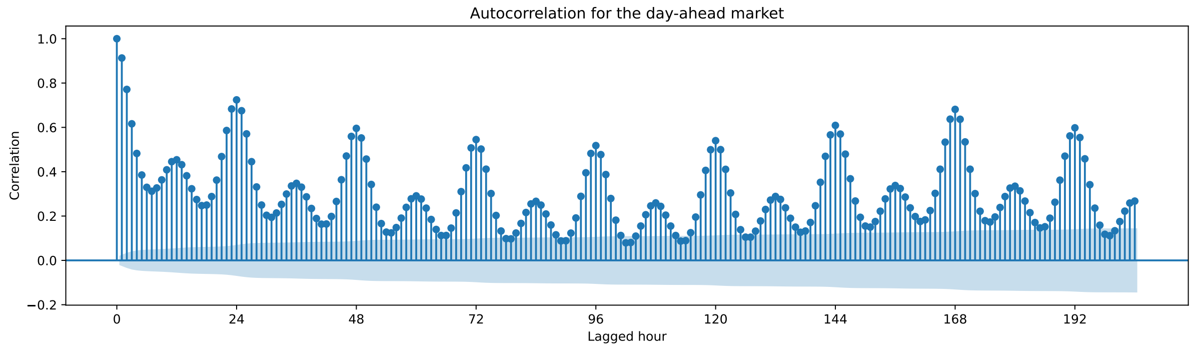


Figure 9: The autocorrelation of the EPEX day-ahead market in 2019. The cone indicates the 95% confidence interval that the null hypothesis is true. If the data point at a certain lagged hour falls outside the confidence interval, it is highly likely that there is a correlation.

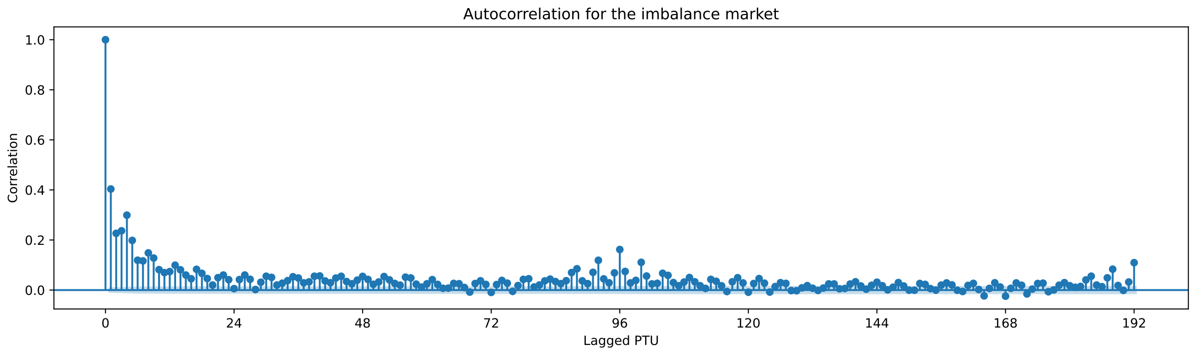


Figure 10: The autocorrelation of the imbalance market in 2019. The cone indicates the 95% confidence interval that the null hypothesis is true. If the data point at a certain lagged hour falls outside the confidence interval, it is highly likely that there is a correlation.

As can be seen, the correlation with lagged hour 0 is equal to 1. Since the lagged hour 0 is equal to itself, the value 1 is according the definition of correlation. Furthermore, lagged hour 1, which is 1 hour apart from the target variable, also has a high correlation with the target variable. However, the previous day-ahead price is not known when the forecast should be submitted, which makes that value inoperable. The first value with a high correlation that is known for half of the target values is the price 24 hours before the target value occurs. That is why the 24 hours lag is included in the input variables for the target variables where this price is known. In addition, the correlation on a daily basis is seen everyday, where two large correlations are found at 48 hours and 168 hours, exactly 2 days and 1 week, respectively. Those two values are included in the input variables too.

In contrast to the autocorrelation of the day-ahead market, the autocorrelation of the imbalance market shows less temporal correlations, which is shown in Figure 10. The only interesting correlations between the target variable and its lagged values is at the previous PTU, 4 PTUs ago, and 96 PTUs ago. Likewise the day-ahead market, the previous imbalance price is not known before the forecast of the imbalance price. Therefore, this variable is not used as an input. Note that the confidence interval of the null hypothesis from the imbalance market is a smaller area than from the day-ahead market. This is due to the number of observations, which is four times as many in the imbalance market as the day-ahead market.

In addition to the autocorrelation, the input variables correlate with the target variable and the other input variables. For the day-ahead target variable, the heatmap of the correlation is shown in Figure 11. The first column indicates the correlation between the input variables and the

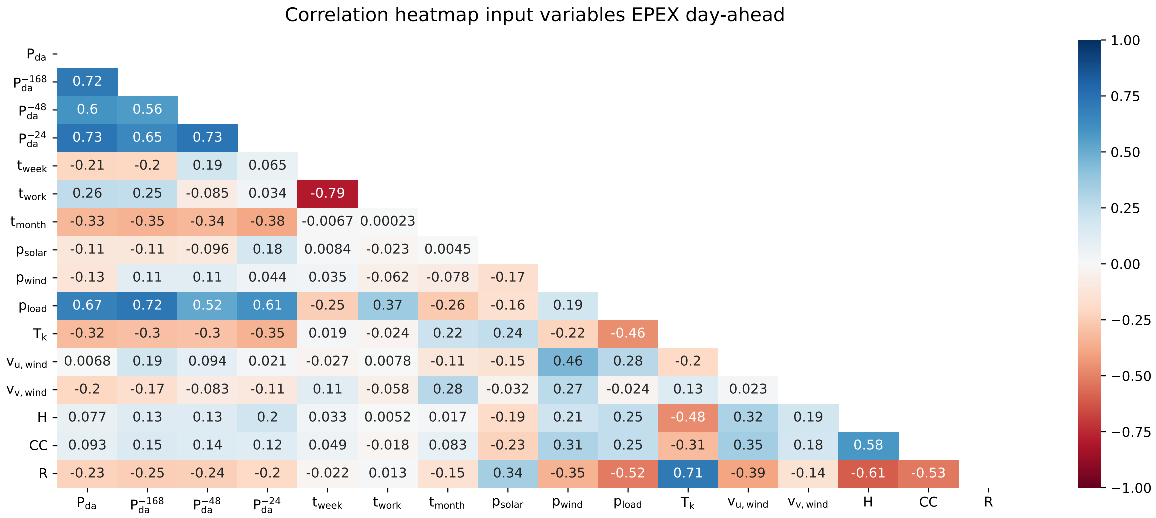


Figure 11: The correlation heatmap of the input variables in respect to the EPEX day-ahead market in 2019. The input variables follow the definitions from Table 4.

day-ahead price. As shown, the three lagging values, P_{da}^{-24}, P_{da}^{-48}, and P_{da}^{-168} all have a strong positive correlation. Note that for the correlation of P_{da}^{-24}, only the hours 0 up to and including 11 are used to calculate the correlation. From the input variables that are not a pricing indicator correlates the forecasted load p_{load} the most with the day-ahead price. Another noteworthy input variable is the forecasted temperature T_k. Their negative correlation indicates that an increasing value of T_k, leads to a decreasing value of P_{da}. This is according to expectations, since a higher temperature indicates more sun and more solar generation. Furthermore, the marginal costs of solar panels are low, which leads to lower prices. The heatmap does not only show the correlations between input variables and target variable, it also shows the correlation between input variables. A remarkable correlation is found at the day of the week and the value to indicate a workday or weekend day. This is not surprisingly, since they both provide calendar information. Moreover, the forecasted weather input variables show relatively strong correlations with each other too.

The features from the imbalance market also correlate to some extent with each other. However, the features and their correlation depend on the variable T_s, which indicates the number of minutes in one PTU that is used to derive the input variables. Even though the results will reflect several values for T_s later on, the correlation heatmap is shown with T_s at 7. The full correlation heatmap is shown in Figure 12. The two strongest positive correlations with the imbalance target price can be found at the P_{up} and the P_{down}. Interestingly, P_{mid} has a weak correlation with the target variable, while the imbalance price is equal to P_{mid} in regulation state 0. One explanation could be the relative low presence of regulation state 0 compared to the other three regulation states. Another interesting strong correlation is found between the highest price upward P_{up}, and the upward regulating volume REG_{up}. That strong correlation could also explain why they both strongly correlate with the imbalance price. The final correlation discussed is the one between the lowest price downward P_{down} and the downward regulating volume Reg_{down}. As expected, their strong negative correlation shows that an increasing volume lowers the price.

4.3. Experimental setup

This subsection discusses the configurations of the methods explained in Section 3. This holds for both the AI approaches and the trading strategies. First, the AI setup is explained.

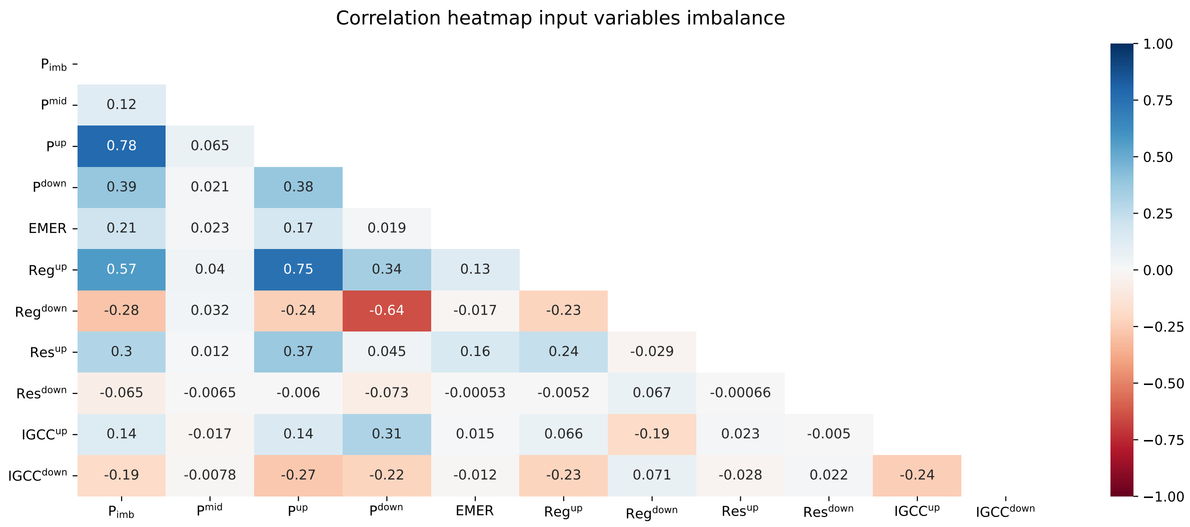


Figure 12: The correlation heatmap of the input variables in respect to the imbalance market in 2019. The input variables follow the definitions from Table 5. The number of minutes to derive the features T_s is set to 7.

4.3.1. AI approaches

Given the four AI approaches from Section 3.1, the hyperparameters require optimization. In this study, the number of iterations for random search to select the best performing hyperparameters from the hyperparameter configuration space is set to 100. The random search algorithm uses a random state of 0, to ensure reproducibility. This also holds for the three AI approaches that use randomization. ANN, RF, GB all have their random state set to 1. Within one iteration of random search, the dataset is split and validated using 5-fold cross validation. All AI approaches are constructed and trained using the Python library Scikit-learn [44]. The hyperparameter configuration spaces for the approaches can be found in Appendix A. Note that the terminology differs from the variables in Section 3. That is due to the terminology used by Scikit-learn.

As stated in Section 3, a hold-out method is used to split the training from the testing set. In this study, the AI models in the forecasting of the day-ahead prices are trained with the full year 2019, while the imbalance pricing models are trained on the last three months of 2019. Since there are four times as many imbalance prices as day-ahead prices, the dataset for training is equally sized. All results are based upon the full year 2020. Then, the best performing AI model from the results in 2020 is used in the trading models. Another decision that affects the results and the trading performance is the setup minutes of the imbalance market T_s. Since there is also a three minute delay between the occurrence and the publication, small numbers of T_s are required to be able to trade with the remaining minutes. Therefore, several setup minutes are tested, with $T_s \in [2, 3, 4, 5, 6, 7]$. The number of datasets N to calculate the CD for the significance test is therefore automatically set to 6. In addition, the significance level α is set to 0.05. 100 Iterations of random search are run parallel on an Amazon AWS server with 32 vCPUs and 128 GiB memory.

4.3.2. Trading models

With the trading models from Section 3.2, several setups are possible. For this study, a single use case is taken to calculate the profit in several configurations. The user data is received from a real-world user with consumption and solar generation. Profits are based on 2020 without the

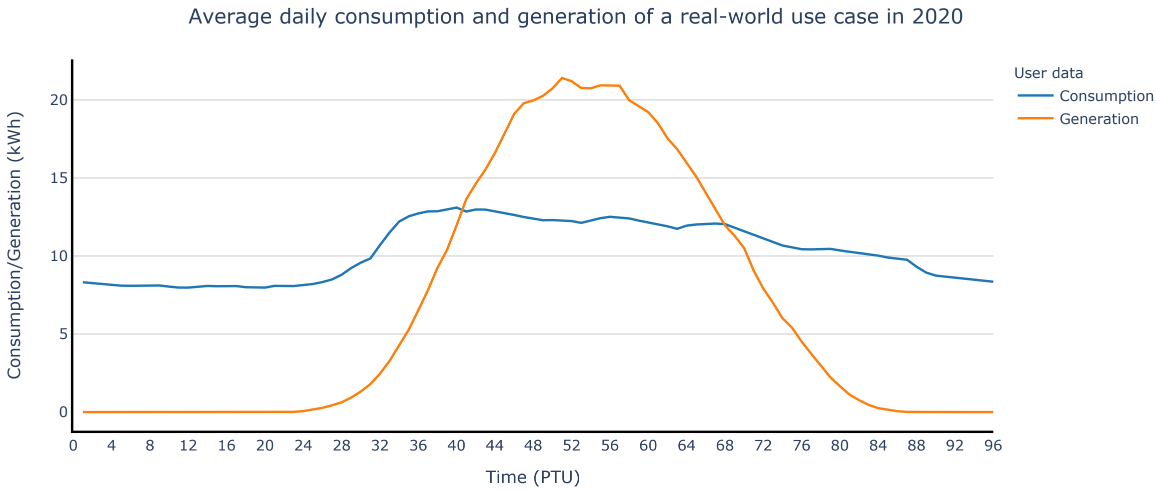


Figure 13: Average daily consumption and generation of a real-world use case in 2020. The generation comes from a 236 kWp solar installation.

leap day. In this year, the consumption was equal to 361912 kWh and there was a total solar generation of 236695 kWh. The daily average of consumption and generation from 2020 can be seen in Figure 13. The figure clearly shows the night and day cycle through the solar generation. Moreover, the start of the working day also shows an increase in consumption. Therefore, this use case represents an average business, such that the results also indicate profits for other businesses.

The results are based upon an installation with one battery and one inverter. The battery has a capacity of 15 kWh, a depth of charge BT_{doc} of $0.8 * 15 = 12$ kWh, and a depth of discharge BT_{dod} of $0.2 * 15 = 3$ kWh. The inverter has a power limitation of 10 kW, which is 2.5 kWh per PTU. $BT_{max,p}$ is therefore fixed to 2.5. At the start of the trading model, the state of charge BT_{soc} is set to 7.5. In addition to the battery installation, X^c_{da}, X^d_{da}, X^c_{imb}, X^d_{imb} should also be assigned. To see the effect of those variables, several values are researched and set as follows: $X^c_{da}, X^c_{imb} \in [1.01, 1.1, 1.2, 1.3, 1.4, 1.5]$ and $X^d_{da}, X^d_{imb} \in [0.99, 0.9, 0.8, 0.7, 0.6, 0.5]$. These variables affect both the profit and the number of battery cycles. Since the number of battery cycles is important for the lifespan of the battery, that value is taken into account too. Even though the strategy performs worse in terms of profit, a strategy could be assigned the best if the number of cycles is significantly smaller. Finally, the number of highest and lowest prices used in $G3$ are set to $X^{max}_{da}, X^{min}_{da} \in [2, 3]$.

4.4. Results

This subsection shows the results of the four AI approaches and the performance of several setups in the trading strategies. At the end, the outcome of two weeks live trading with a BESS is given too. The results of the price forecasting are compared to some baselines and external forecasters. Moreover, the trading strategy is compared to external suppliers, which provide buy and sell signals.

4.4.1. AI approaches

The results of the AI approaches is split into the day-ahead price P_{da} forecasting and the imbalance price P_{imb} forecasting. Both targets have two accuracy performance metrics, MAE

Table 7: An overview of the performance with the AI approaches in the day-ahead market compared to a baseline for 2020. The numbers in brackets indicate the achieved scores if the model is applied on the training set.

AI approach	MAE [training]	MSE [training]	Run time (s)
Baseline: P_{da}^{t-168}	7.95	138.14	-
ANN	9.41 [3.99]	151.01 [28.65]	314
SVR	13.26 [0.79]	293.87 [0.70]	236
RF	9.21 [2.09]	151.93 [9.43]	1498
GB	8.85 [3.41]	143.65 [21.76]	41

Table 8: The best performing hyperparameters of GB on the day-ahead forecasts. These hyperparameters are used in the trading strategies. The terminology is consistent with Scikit-learn [44].

Hyperparameter	Value
$learning\_rate$	0.15817811851184635
$loss$	$'ls'$
max_leaf_nodes	759
$min\_samples\_leaf$	18
$n\_iter\_no\_change$	16
tol	$1e-07$
$validation\_fraction$	0.3620134054564678

and MSE, given by Equation 19 & 20, respectively. Informally, MAE measures the average absolute difference between the forecasted value and the actual value. MSE on the other hand, assigns an extra penalty for large differences in the forecast by squaring the error. Along with the accuracy, the duration to find the optimal hyperparameters with 100 iterations of random search is given too.

Day-ahead price forecasting An overview of the performance metrics for the forecasting of day-ahead prices is found in Table 7. The baseline for model comparison is set to the day-ahead price 168 hours before the forecasted day-ahead price. This value is known before the market closure for all targets and has a strong correlation, which can be seen in Figure 11. Therefore, P_{da}^{t-168} is set as the baseline forecaster.

As shown in Table 7, the baseline forecaster performs best in terms of MAE and MSE. GB on the other hand, is the best performing AI approach. While RF achieves the third place in terms of MAE, ANN is performing slightly better with the MSE. Finally, SVR performs worst for both the MAE and MSE. The differences in forecasters can also be seen in Figure 14. As shown, the shape of all forecasters is often similar to the actual day-ahead price graph.

Furthermore, the run time statistics show some interesting findings too. The best performing model, GB, is also the fastest model to find the optimal hyperparameters with random search. RF on the other hand, is more than 35 times as slow as GB. Notably, that did not result in a bad forecasting model. Due to its performance, GB will be used in the trading strategy. Here, the used hyperparameters are set to the ones that perform best and are shown in Table 8.

Imbalance price forecasting An overview of the performance metrics for the forecasting of imbalance prices is found in Table 9. Instead of P_{da}^{t-48}, the baseline is set to the mid price P_{mid}. According to the definition of P_{mid}, this value is known before the start of the PTU

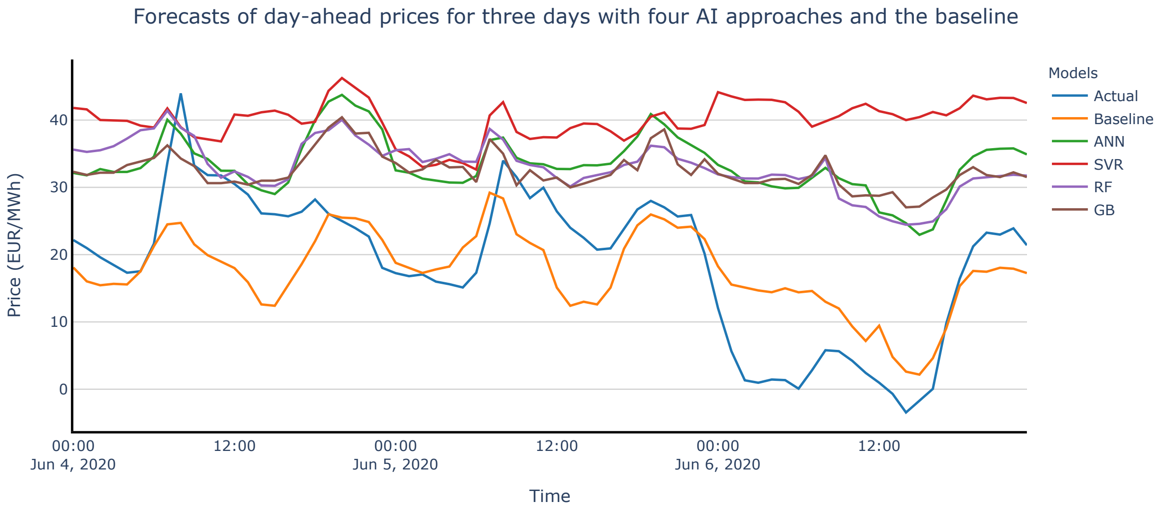


Figure 14: Forecasts of the day-ahead market for three days (Thursday, Friday, Saturday) in June 2020. The blue line represents the actual day-head price. Furthermore, the baseline and four AI approaches are shown.

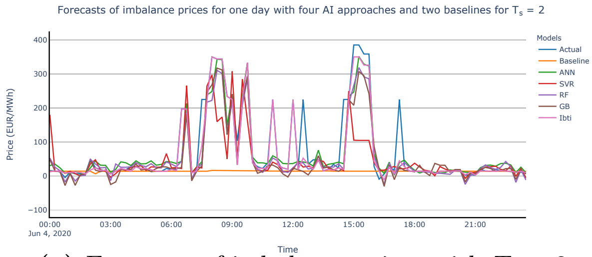
and is used in one out of four regulation states as the actual imbalance price. Therefore, the P_{mid} could be an indication of the imbalance price P_{imb}. Furthermore, two external forecasting suppliers are added as a baseline to compare the results. The first external forecaster is called Ibti. Ibti is a linear model, which requires seven minutes of setup time. The model is retrieved from a subsidy project within Friday Energy [50]. The exact details on the linear model are not publicly available yet. Another party in the project is EP. EP also forecasts the imbalance market and made the results available. However, the results of EP reflect another date range, namely, from 2021/03/24 up to and including 2021/05/31. Both the model type and T_s are unknown for this model. However, multiple forecasts are made within one PTU, indicating that multiple T_s are used too. An average is taken from the forecasts if multiple forecasts were available in one PTU.

An example of how the forecasts fluctuate in one day can be seen in Figure 15. The results show that every AI approach is performing better than the baseline P_{mid} with all setup times. Moreover, there is a strong negative correlation between T_s and the performance metrics, i.e., an increasing T_s leads to better performing models. Furthermore, two AI approaches have at least one T_s where they perform best in terms of MAE. For ANN this is at $T_s = 4, 5, 6$, and for GB this is at $T_s = 2, 3, 7$. Even though SVR and RF do not perform best for any T_s in terms of MAE, they are often close to the others. That is why the Nemenyi test from Figure 16 provides more insights on the best performing AI approach over multiple datasets. It can be seen that ANN outperforms the other models and achieves the highest ranking in terms of MAE. However, there is no significant difference between all AI approaches. Only ANN, RF, and GB differ significantly from the baseline methodology. Interestingly, RF and GB get the same rank on their performance, while RF is never the best performing AI approach.

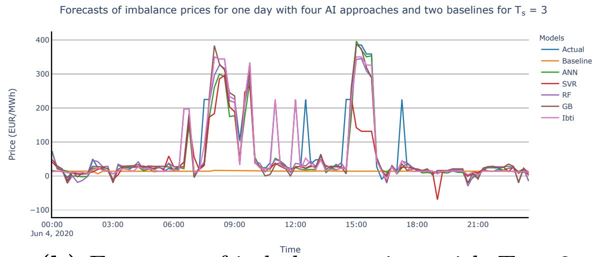
On the contrary of MAE, MSE from Table 9 shows a more clear winner. In 5 out of 6 T_s values, ANN is performing the best. However, Figure 17 also shows that the difference is not significant compared to RF and GB. ANN is the only AI approach that is outperforming both the baseline and the SVR with a significant difference. Another interesting finding is that the best performing AI approach with T_s is not performing better than the model of Ibti in terms of MAE. However, the AI approaches outperform the model of Ibti with the MSE. Furthermore, most AI principles outperform EP when T_s is larger than 2. Finally, an increase in T_s often

Table 9: An overview of the performance with the AI approaches in the imbalance market compared to a baseline. The numbers in brackets indicate the achieved scores if the model is applied on the training set.

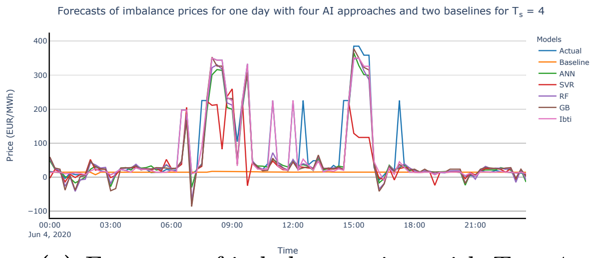
AI approach	T_s	MAE [training]	MSE [training]	Run time (s)
Baseline: P_{mid}	-	30.52	5792.41	-
Baseline: Ibti	7	14.28	2258.32	-
Baseline: EP	-	24.97	3259.24	-
ANN	2	27.19 [20.49]	3382.70 [1577.51]	289
	3	24.56 [17.69]	3183.50 [1274.56]	311
	4	23.26 [16.64]	2844.62 [1159.73]	289
	5	20.93 [14.33]	2611.22 [997.89]	302
	6	19.38 [13.21]	2337.38 [890.23]	304
	7	17.08 [11.59]	1982.65 [719.86]	286
SVR	2	27.11 [13.19]	4389.77 [1393.17]	6065
	3	25.25 [12.66]	4275.88 [1290.95]	7253
	4	23.47 [11.51]	3806.24 [1114.94]	7355
	5	22.35 [10.32]	3528.91 [946.39]	7848
	6	20.53 [9.17]	3224.24 [821.12]	8025
	7	18.59 [7.94]	2901.32 [646.09]	8462
RF	2	26.79 [15.94]	3439.94 [1107.99]	1066
	3	24.90 [14.61]	3188.63 [968.09]	1072
	4	23.50 [13.37]	2911.87 [859.90]	1101
	5	22.25 [12.02]	2691.01 [746.12]	1104
	6	19.72 [11.03]	2362.59 [683.05]	1133
	7	15.86 [7.04]	2019.17 [534.58]	1333
GB	2	26.45 [18.53]	3426.48 [1371.93]	12
	3	24.53 [17.45]	3160.03 [1249.52]	13
	4	24.05 [16.05]	2958.70 [1123.90]	14
	5	22.74 [14.52]	2697 [930.49]	14
	6	19.74 [13.39]	2363 [890.67]	15
	7	15.72 [9.50]	2014.57 [704.11]	17



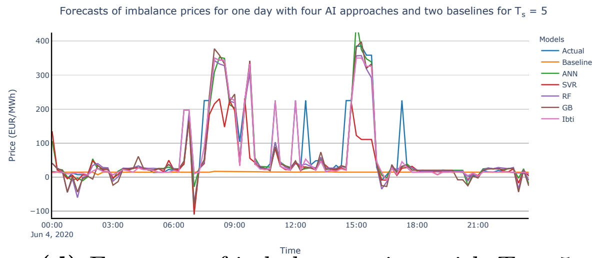
(a) Forecasts of imbalance prices with $T_s = 2$



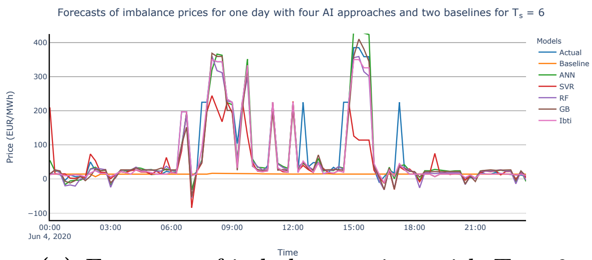
(b) Forecasts of imbalance prices with $T_s = 3$



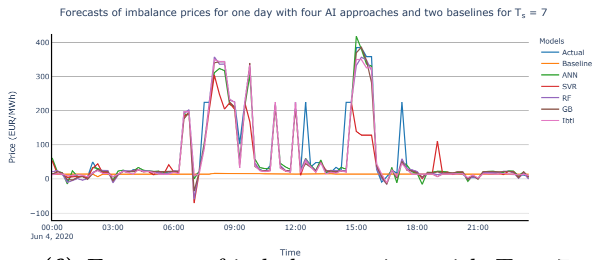
(c) Forecasts of imbalance prices with $T_s = 4$



(d) Forecasts of imbalance prices with $T_s = 5$



(e) Forecasts of imbalance prices with $T_s = 6$



(f) Forecasts of imbalance prices with $T_s = 7$

Figure 15: Forecasts of the imbalance prices for one day in June 2020. The blue line represents the actual day-head price. Furthermore, the two baselines and four AI approaches are shown. Each graph differs in the number of setup minutes T_s.

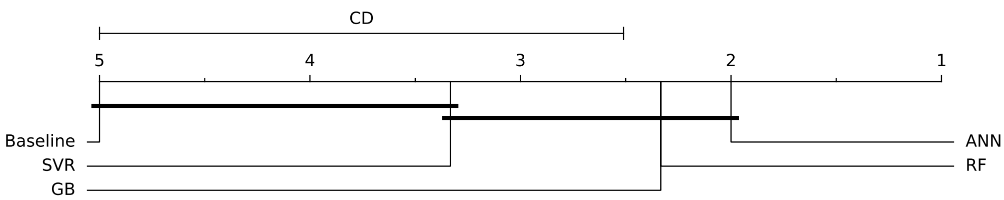


Figure 16: A ranking of the AI approaches in terms of MAE for the different T_s values in the imbalance market. The best performing AI approach achieves the lowest and rank and is the rightmost value. A Nemenyi test determines the significant performance differences between several models, i.e., if they are at least the critical distance (CD) apart.

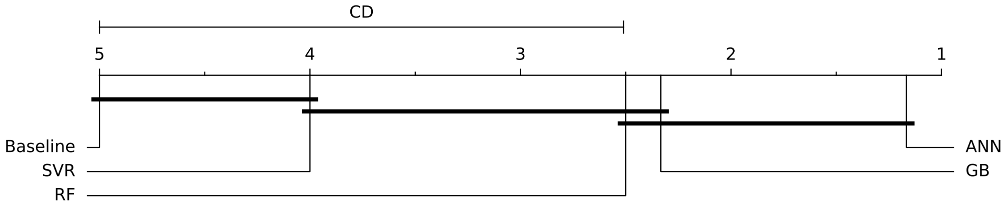


Figure 17: A ranking of the AI approaches in terms of MSE for the different T_s values in the imbalance market. The best performing AI approach achieves the lowest and rank and is the rightmost value. A Nemenyi test determines the significant performance differences between several models, i.e., if they are at least the critical distance (CD) apart.

Table 10: The best performing hyperparameters of ANN on the imbalance price forecasting. These hyperparameters are used in the trading strategies. The terminology is consistent with Scikit-learn [44].

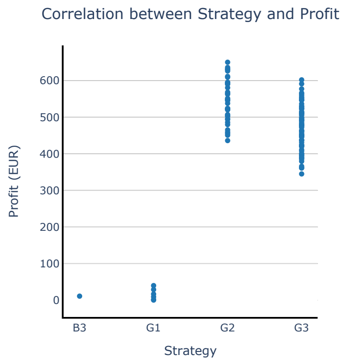
Hyperparameter	Value
$activation$	$'relu'$
$alpha$	0.09755215294813809
$early\_stopping$	$True$
$hidden\_layer\_sizes$	$(32, 32, 32)$
$learning\_rate\_init$	0.005955870684082486
$n\_iter\_no\_change$	32
$solver$	$'adam'$
tol	0.0001
$validation\_fraction$	0.1

leads to an increase in run time too. Except for ANN, there is no clear correlation between T_s and the run time.

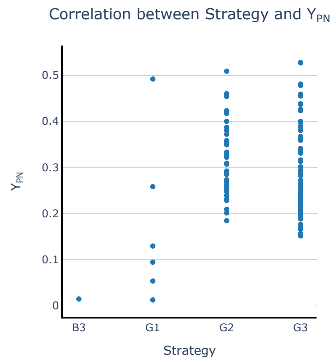
Since ANN is the best performing AI approach, albeit insignificant, that model is used in the trading strategies. The best performing hyperparameters that have been found with randomized search at $T_s = 5$ are shown in Table 10.

4.4.2. Trading strategies

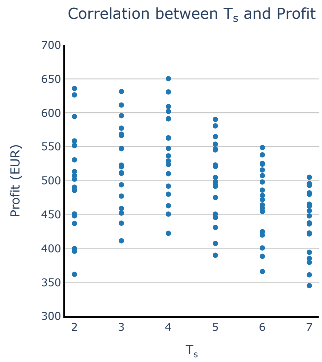
Given the best forecasts for 2020 with GB in the day-ahead market and ANN in the imbalance market, the trading strategies from Section 3.2 are run with several setups. All trading results are shown in Appendix B. First, the results from the trading strategies are analyzed with Figure 18, which shows an overview of the influence of the parameters with the profit and Y_{PN}. Strategy G_2 earns the highest profit, followed by G_3 and G_1, respectively. However, the differences in strategies in respect to Y_{PN} is less clear. The highest Y_{PN} is seen at G_3. However, many G_3 trading strategies also perform worse in terms of Y_{PN} than G_2. Since G_3 has more data points, these two trading strategies are hard to distinguish in terms of Y_{PN}. A much clearer difference is visible at the influence of T_s on the profit. It seems that there is an optimum at $T_s = 4$. At $T_s = 2$ and $T_s = 3$, the profit could be increased by adding one minute to T_s. At $T_s = 4$, there is a turning point, where an increase of T_s would lead to lower profits. That turning point is not visible for the correlation between T_s and Y_{PN}. A higher T_s leads to higher Y_{PN} too. An interesting finding can be seen by comparing Figure 18e & 18f, and by comparing Figure 18g & 18h. While the profit increases with a higher X^c_{da} or X^c_{imb}, the Y_{PN} decreases. Finally, small X^{max}_{da} values lead to both higher profits and better performance in terms of Y_{PN}, although the outperformance in Y_{PN} is less clear.



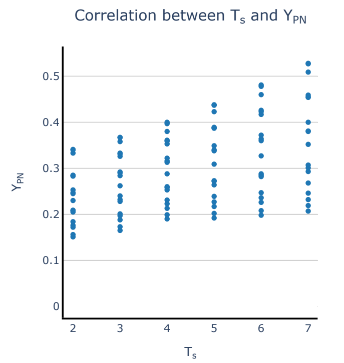
(a) Trading results plot between strategy and profit



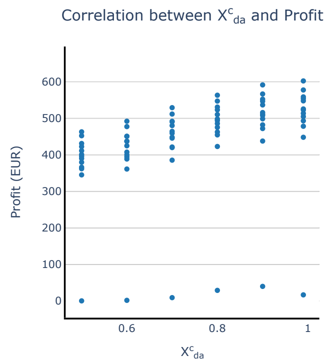
(b) Trading results plot between strategy and Y_{PN}



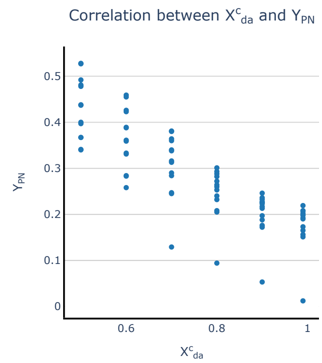
(c) Trading results plot between T_s and profit



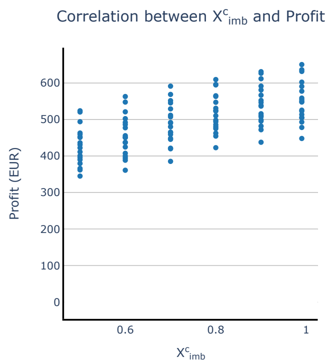
(d) Trading results plot between T_S and Y_{PN}



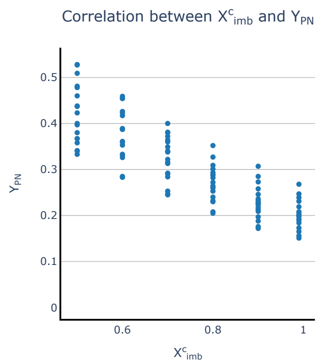
(e) Trading results plot between X^c_{da} and profit



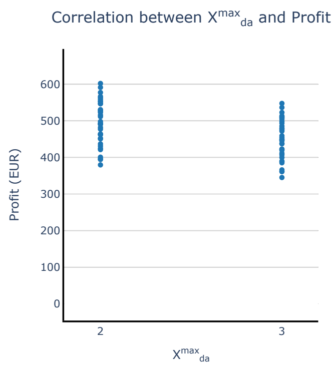
(f) Trading results plot between X^c_{da} and Y_{PN}



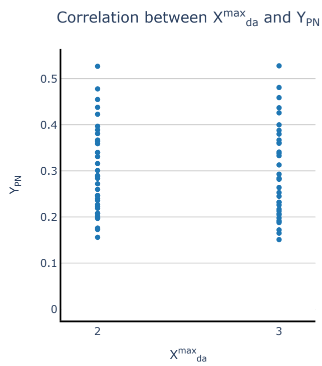
(g) Trading results plot between X^c_{imb} and profit



(h) Trading results plot between X^c_{imb} and Y_{PN}



(i) Trading results plot between X^{max}_{da} and profit



(j) Trading results plot between X^{max}_{da} and Y_{PN}

Figure 18: Plots of several inputs in the trading strategies and the results (profit or Y_{PN}).

Table 11: An overview of the best performing trading strategies, which is a shorter version of Appendix B. The profit is calculated by subtracting the electricity costs of the strategy from the electricity costs of base line B_1. The numbers in brackets indicate the performance that could be achieved if the forecast was equal to the actual value.

Strategy	T_s	X_{da}^c	X_{da}^d	X_{imb}^c	X_{imb}^d	X_{da}^{max}	X_{da}^{min}	Electricity costs [perfect]	Profit [perfect]	N_{BT} [perfect]	Y_{PN} [perfect]
B_1	-	-	-	-	-	-	-	-5672.04	-	-	
B_2	-	-	-	-	-	-	-	-7648.67	-1976.63	-	-
B_3	-	-	-	-	-	-	-	-5661.07	10.97	790	0.014
G_1	-	0.99	1.01	-	-	-	-	-5655.37 [-5627.84]	16.67 [44.2]	1350 [1398]	0.012 [0.032]
G_2	4	-	-	0.99	1.01	-	-	-5021.67 [-4837.58]	650.37 [834.46]	2820 [2247]	0.231 [0.371]
G_2	2	-	-	0.99	1.01	-	-	-5035.95 [-4673.74]	636.09 [998.3]	3449 [2671]	0.184 [0.374]
G_3	7	0.5	1.5	0.5	1.5	3	3	-5326.94 [-5389.63]	345.1 [282.41]	653 [575]	0.528 [0.491]
G_2	5	-	-	0.6	1.4	-	-	-5151.05 [-4966.65]	520.99 [705.39]	1347 [1139]	0.387 [0.619]
G_3	4	0.7	1.3	0.7	1.3	2	2	-5142.98 [-5126.1]	529.06 [545.94]	1677 [1443]	0.316 [0.378]

Secondly, the trading strategies are analyzed on an individual level, where the highlighted strategies in this paragraph are also shown in Table 11. The highest profit of 650 EUR is achieved with G_2, using the settings $T_s = 4, X_{imb}^c = 0.99, X_{imb}^d = 1.01$. However, N_{BT} is high as well, leading to a relative low Y_{PN}. Compared to the highest profit with perfect forecasts of 998 EUR, the AI approaches are able to take 65% of the profit. G_3 with $T_s = 7, X_{da}^c, X_{imb}^c = 0.5, X_{da}^d, X_{imb}^d = 1.5, X_{da}^{max}, X_{da}^{min} = 3$ performs best in terms of Y_{PN}. Here, the profit and battery cycles ratio is equal to 0.528. Interestingly, the profit that is achieved with that trading strategy and setup is higher than the perfect profit that would be achieved with the actual imbalance and day-ahead prices. That also led to a higher N_{BT} than the perfect setup would achieve.

While Y_{PN} could be the only parameter for the optimization of the trading model, the battery should also be paid back in an acceptable period of time. Therefore, the minimal amount of profit is set to 500 EUR for selecting the final trading model. Then, the model with the highest Y_{PN} is used in live trading. Using this criteria, the best performing trading model is G_2 with $T_s = 5, X_{imb}^c = 0.6, X_{imb}^d = 1.4$. For trading strategy G_3, the same criteria leads to a setup of $T_s = 4, X_{da}^c, X_{imb}^c = 0.7, X_{da}^d, X_{imb}^d = 1.3, X_{da}^{max}, X_{da}^{min} = 3$. This setup is mentioned too, because the best performing setup for all three general trading strategies are compared in Figure 19. The settings for G_1 are selected on N_{BT}, shown in Figure 19, which should be close to the selected G_2 and G_3. That led to the trading strategy G_1 with $X_{da}^c. = 0.99, X_{da}^d = 1.01$. G_2 and G_3 have a similar cumulative profit curve, which is close to a linear line. As already seen in Table 11, G_3 achieves a slightly higher profit than G_2. Note that N_{BT} and Y_{PN} are both higher in this case, leading to a worse performing strategy.

4.4.3. Live trading

Since the best performing trading strategy only trades on the imbalance market, organizations do not have a trade off between implementation costs and profit. However, other selections still remain, which are based on the software application provided in Algorithm 2. First of all, the time window to train the model is set from 2021/01/01 to 2021/06/01 for the real-world scenario. Secondly, the frequency in which BT_{soc} is checked and the input data is collected is

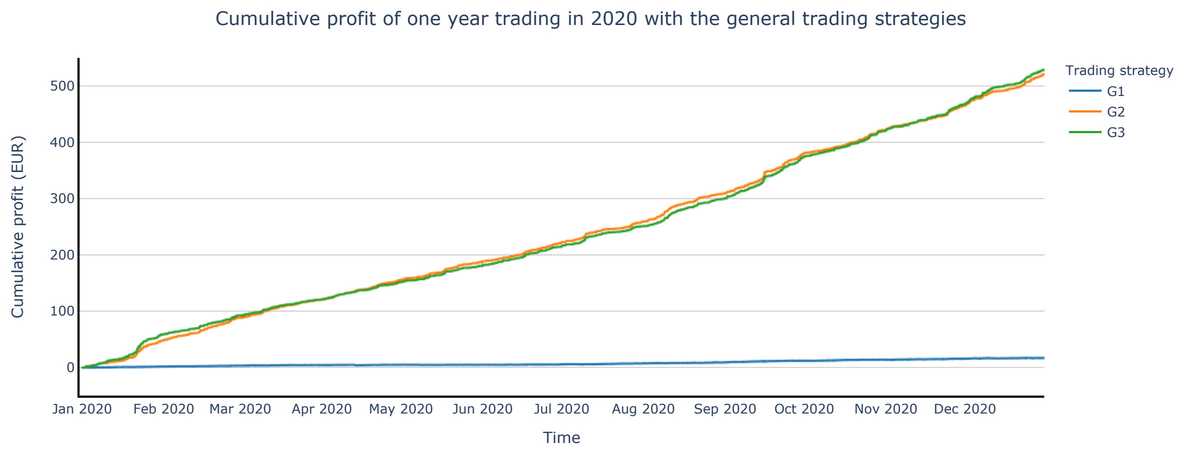


Figure 19: A graph representing the cumulative profit of three general trading strategies. The input variables are discussed in Section 4.4.2.

Table 12: A performance overview of the live trading strategy. Two separate weeks are tested and shown. The number in brackets at the profit column indicates the profit that would have been achieved if the profit from the given period is achieved a full year.

Strategy	T_s	X^c_{imb}	X^d_{imb}	Date from	Date to	number of days	MAE	MSE	Profit [year]	N_{BT} [year]	Y_{PN}
G_2	5	0.6	1.4	2021/07/26 17:00	2021/07/30 23:45	4.3	48.81	6068.99	9.63 [817.76]	15.2 [1292]	0.633
G_2	5	0.6	1.4	2021/08/25 17:15	2021/09/01 17:00	7.0	40.23	11781.30	13.87 [723.22]	12.6 [656]	1.10

equal to one minute. Thirdly, two data providers are selected due to their free of charge real-time data [59, 18]. Finally, the requirements for the architecture from Figure 6 and Algorithm 2 led to Amazon's web services (AWS) as the web server to host the software application.

An overview of live trading results is shown in Table 12. In both weeks, the performances in terms of MAE and MSE are worse than the performances at the imbalance forecasting with the testing set, shown in Table 9. However, the trading results in terms of profit and Y_{PN} show opposite performances. Here, the profit that would have been achieved in one year is increased with 1.5 and 1.4 times in the first and second week, respectively. Moreover, Y_{PN} is increased with 1.6 and 2.8 times in the first and second week, respectively. Figure 20 shows how the performance statistics are achieved. The bottom two figures show the energy prices from the imbalance and day-ahead market over time. As can be seen here, the forecasted imbalance price is often in line with the actual prices. However, the forecast predicts an average of the feed and consume price. In some rare cases, the consume price differs significantly from the feed price. Other findings are the extreme negative prices in the first week, while the second week shows more extreme positive prices. The top two figures show the reaction of the battery to the market signals, by charging and discharging, as shown with the red line. BT_{soc} is in this case expressed in percentage. The cumulative profit is also shown in this graph with the blue line. In both weeks, the profit increases steadily over time, with few drops. This is in line with the findings from Figure 19, where the cumulative profit for one year of trading is shown. Figure 20 also shows that BT_{soc} movements almost always lead to an increase in profit.

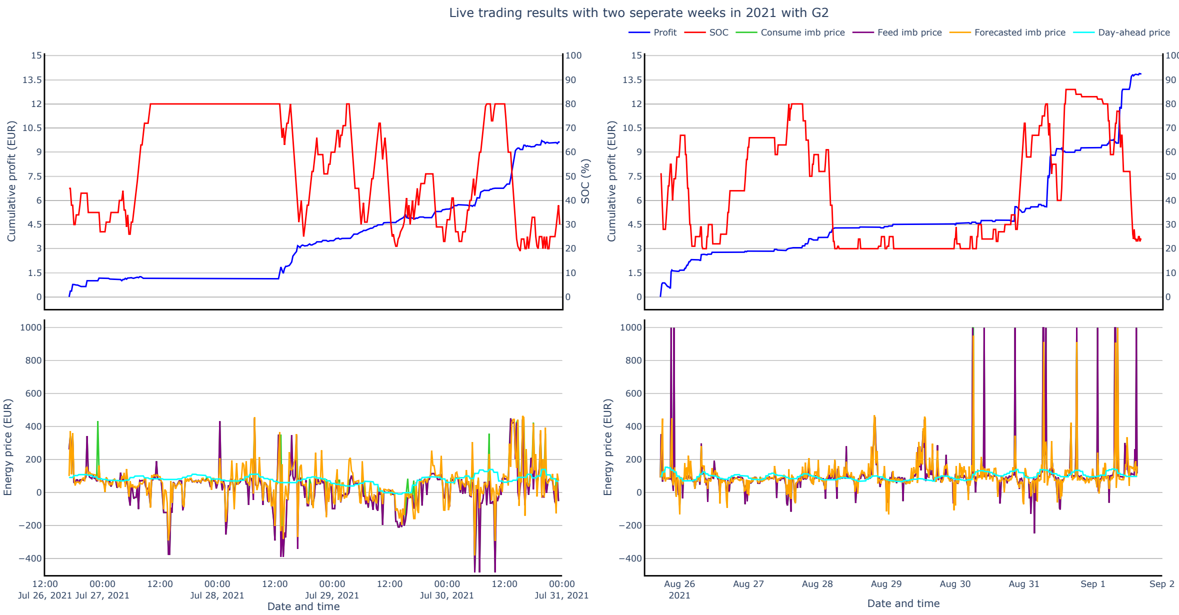


Figure 20: Four graphs that represent several statistics of two weeks live trading in a real-world setting. The left two graphs reflect 5 days of real-world trading, while the right two graph reflect a full week.

5. Discussion

This section continues with the meaning of the results from the previous section. Several findings were given for the AI approaches in two markets, the results of the static trading model and the results of the live trading model. Here, an analysis of these results is given.

5.1. AI approaches

Notably, all AI approaches perform worse in forecasting the day-ahead market price compared to the baseline, while the shape of the forecasting graphs is similar to the actual prices. One explanation for the large forecasting errors is the pandemic called COVID-19. In 2020, COVID-19 entered the Dutch borders and affected the economy. This led to a decrease in electricity demand and a drop in electricity prices [24]. The average day-ahead prices in 2020 dropped with 8.95 EUR/MWh compared to 2019. This is also visible in the average hourly prices in 2020 compared to 2019, shown in Figure 21. The two years show a visible shape. However, the day-ahead prices in 2020 are significantly lower. That also explains why the baseline forecaster, P_{da}^{t-168}, performs better than the other models. The baseline method does not construct its forecast on the higher prices from 2019.

Besides the lesser performance of the AI approaches due to differences in prices from 2019 and 2020, one AI approach is not performing well at all. The performances of the models on the training sets show that SVR overfits its model to the 2019 dataset, such that is less applicable on the dataset of 2020. Even 5-fold cross validation is not preventing overfitting in this setup. Another interesting finding is the comparable performance of ANN and RF. While RF performs better in terms of MAE, ANN performs better in terms of MSE. This indicates that the RF is more often closer to the actual day-ahead price than the ANN. However, the ANN has less often large errors in the forecast than the RF.

In accordance with day-ahead market forecasting, the SVR performs worst with imbalance pricing in terms of both MAE and MSE. However, for imbalance pricing, the model shows

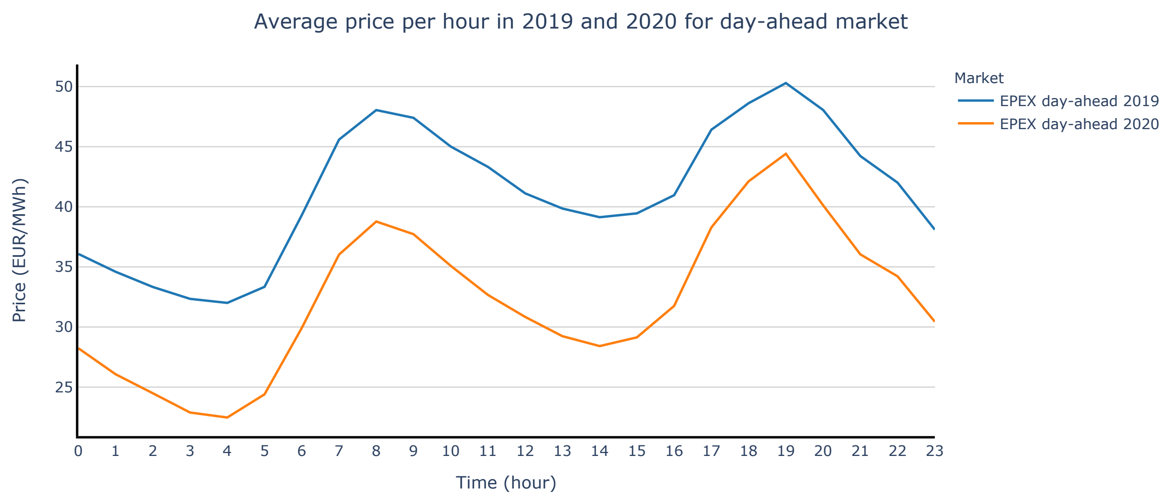


Figure 21: A comparison of the average hourly EPEX day-ahead prices in 2019 and 2020.

Table 13: An overview of the statistics for the EPEX day-ahead market and imbalance market in 2020. The prices of these two markets were forecasted with the AI approaches.

Statistical measure	EPEX day-ahead	Imbalance	Unit
Mean	32.24	34.55	EUR/MWh
Median	31.67	21.98	EUR/MWh
Minimum	-79.19	-561.17	EUR/MWh
Maximum	200.04	797.23	EUR/MWh

less sensitivity to overfitting. Furthermore, SVR has a larger run time in the forecasts of the imbalance prices, while the other models are faster on the imbalance market. Since the randomized search algorithm tried the same hyperparameters in both the day-ahead market and the imbalance, it could not depend on a time expensive hyperparameter. Moreover, the number of input variables in the day-ahead market is larger than in the imbalance market. That could explain why all other AI approaches are faster. It seems that for SVR, the run time is dependent on the values of the input variables. That could also explain the increasing run time with a larger T_s for SVR, RF and GB.

Due to the larger price fluctuations in the imbalance market, the larger errors of the AI approaches in the imbalance market is in accordance with expectations. In addition, the imbalance market in 2020 showed different characteristics than in 2019, which was also seen for the day-ahead market. Table 13 shows that the average imbalance price in 2020 is smaller than the average imbalance price in 2019, which is shown in Table 6. Furthermore, the minimum and maximum are both smaller too. However, Figure 22 shows that the average price peak at certain PTUs are larger.

Since the AI approaches in the imbalance market outperform the baseline with ease, these models look more promising than the forecasts of the day-ahead, where the AI approaches do not outperform the baseline. However, from Figure 14 can be seen that the trend of the day-ahead market is forecasted accurately. This could be exploited in a trading strategy, even when the errors are relatively high.

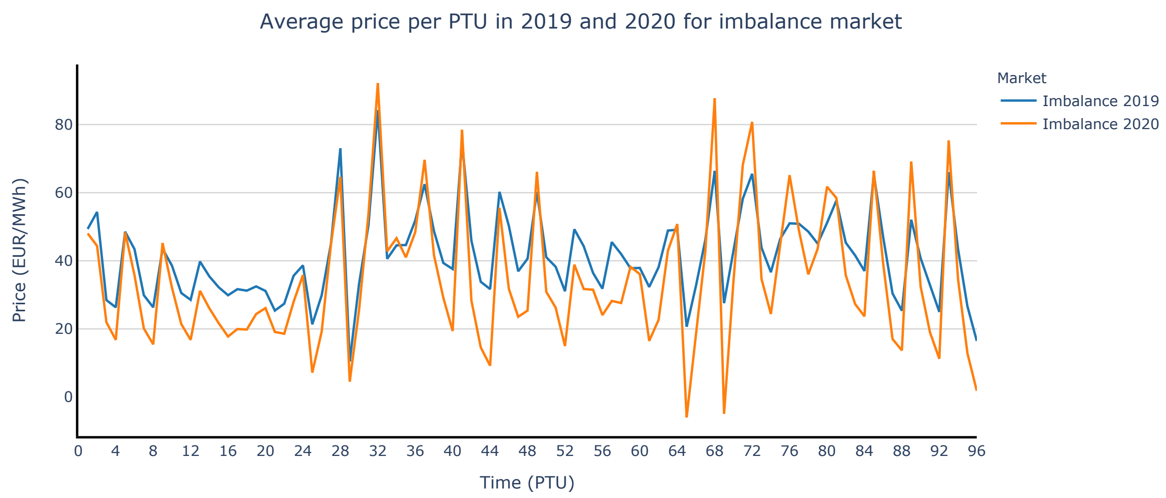


Figure 22: A comparison of the average imbalance prices per PTU in 2019 and 2020.

5.2. Trading models

This subsection continues with highlighting several interesting findings from the trading results. First of all, the results have shown that G_2 is performing best in terms of profit and performs equally good with G_3 in terms of Y_{PN}. Therefore, G_2 is assigned the best performing trading model. That means that the imbalance market is more profitable than the day-ahead market. Even the most promising hours from the day-ahead market could not lead to a better trading model than a trading strategy which settles the electricity from the battery exclusively on the imbalance market. That is interesting, since the electricity costs were higher in B_2, where the user electricity is settled on the imbalance market, than $B1$, where the user electricity is settled on the day-ahead market. Furthermore, the forecasting errors in the imbalance market are larger than in the day-ahead market. One explanation for the higher profits could therefore be that the exact imbalance price is less important than a correct indication if the price will be at least the threshold higher or lower than the day-ahead price. That could also explain the relatively high profits compared to the profits made with perfect forecasts. Even though the MAE is half the average imbalance price, the profits and Y_{PN} are often close to the theoretically achievable profits and Y_{PN} with perfect forecasts. Another explanation for the better performance in the imbalance market is the presence of extremely large and extremely small values in the imbalance that could be exploited. These values are never seen in the day-ahead market.

A closer look at the strategies with imbalance trading also reveals that a wrong forecast could lead to more profit and a higher Y_{PN}. An example that led to higher profit and Y_{PN} is G_3, with $T_s = 7, C^c_{da}, X^c_{imb} = 0.5, X^d_{da}, X^c_{imb} = 1.5, X^{max}_{da}, X^{min}_{da} = 3$. This indicates that the input selection of G_3 were not the best, otherwise the perfect forecast should outperform the forecasted model. Furthermore, small X^c_{imb}, and therefore large X^d_{imb}, led to large Y_{PN}. However, investments should be earned back in a acceptable period of time. Therefore, the absolute profit and Y_{PN} should both be considered to select the final trading strategy. In this study, the best trading strategy achieved an annual profit of 521 EUR and an Y_{PN} of 0.387. This was also shown in the cumulative profit curve from Figure 19. An interesting finding in this profit curve is the increasing linear line. The linear line indicates that almost all decisions to buy or sell electricity are correct. One explanation could be the threshold in the trading strategy. For G_2 to get a sell signal, the forecasted price must be at least 1.4 times the day-ahead price. A buy signal is triggered if the forecasted price is smaller or equal to 0.6 times the day-ahead price. Therefore, the threshold ensures that only the forecasts with large differences in respect to the day-ahead price are used to trade. The large difference could increase the likelihood that at least the correct side of the day-ahead price is chosen, i.e., the imbalance price is smaller than the day-ahead price on a buy signal and higher on a sell signal. This would then still lead to a profitable trade, even though the threshold would not have been met with the actual imbalance price. That also explains why the achieved profit is smaller than the achievable profits with perfect forecasts. If the forecasts indicate a price that meets the threshold even though the actual price would not meet the threshold, the battery is charged or discharged with less favorable prices. The capacity of the battery is then already used and that could lead to a scenario where the battery cannot be exploited in times when the threshold would have been met.

5.3. Live trading

From the MAE and MSE in the live trading strategy compared to the static models can be concluded that the AI model performs worse in live trading. One explanation could be the difference in pricing during the training of the AI model and the prices in the live trading period. In other words, the prices in the training set do not reflect the prices in the live trading set. Another explanation is the extreme fluctuation in pricing during two weeks of live trading. The distribution of imbalance prices, visible in Figure 7b, show that extreme values which occurred during live trading were scarce in 2019. This could indicate that the imbalance market is changing over time and extreme values are achieved more often. That also explains the larger error in 2021 compared to 2020. A wrong forecast at extreme imbalance prices could rapidly lead to a larger MAE and even larger MSE. This is in line with the profits that are achieved in two weeks. More extreme values allow for a larger profit with the same amount of trades. G_2, which is used during live trading, could not even make the same profits with perfect forecasts as the live trading weeks made. However, it is hard to draw final conclusions with two weeks of live trading, whereas the static models are run for a year.

In contrast to static models, live trading results also reflect performances of the implementation within an organization. One finding during live trading was the mismatch between the hypothetical power of the inverter and the actual power of the inverter. As such, the real-world scenario often achieved a maximum charging power of 6 kW, instead of 10 kW that was used in static models. Nonetheless, the real-world strategy performed well. One explanation is the extreme prices described above. In that case, live trading would have performed even better in terms of profit and Y_{PN} when the theoretically charging power of 10 kW was met. Another explanation is that the prices are often low in consecutive PTUs. Missing charging power from one PTU is then compensated in the next PTU. This scenario holds for week one, where BT_{soc} is often close to BT_{doc}. However, week two shows that BT_{soc} is often close to BT_{dod}, which means that an improvement is possible if the battery is charged with more power. A second real-world implementation issue that was discovered during live trading is a jump in BT_{soc}. This jump led to a scenario where the battery discharged for at maximum one minute, before it was stopped by the software application. An improvement in the software application that only stops the battery from charging or discharging if BT_{soc} is above or below BT_{doc} and BT_{dod}, respectively, would lead to even higher profits. Whether Y_{PN} also would have increased in this scenario is hard to say. However, the problem occurred during extreme high imbalance prices, which could have ensured a larger value. One final difficulty that was encountered in this study is the operability of the architecture given in Figure 6. Since the architecture depends on separate devices and communication standards, it is highly likely that the communication flow will be interrupted. In this study, the communication flow was interrupted at the IoT platform. Due to a licensing problem, market signals from the software application did not reach the inverter. Therefore, the results for live trading is based on a shorter time period than expected.

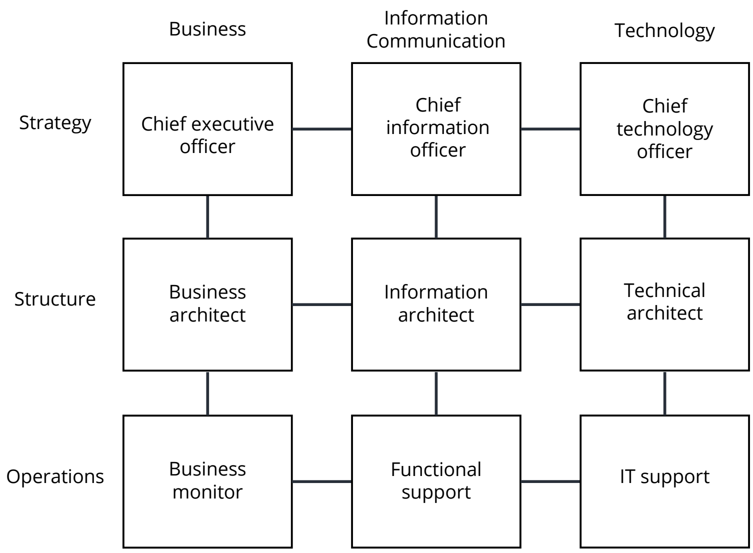


Figure 23: A recommendation for a future organizational architecture that combines business and IT by adding an information communication layer. The model is adapted from Maes [38] and added with roles within the organization.

5.4. Future organizational structure

On top of the findings in terms of MAE, MSE, profit and Y_{PN}, a future organizational structure to maintain the live trading system is also a result of this study. First of all, the maintenance of the system concerns both information technology (IT) and business departments. From a high level perspective, IT should maintain AI models, software application and IoT platform, while business should focus on changing market environments and model selection criteria. The two departments should together discuss what architecture is required for the best implementation. From the experiences in this study, a suggestion is made for a future organizational structure to implement and maintain the trading system.

As mentioned, business and IT should cooperate in the maintenance of the trading system. One mentality that disturbs the cooperation between two departments is silo thinking. With silo thinking, departments within organizations, e.g., business, IT and administration, are organized into separate groups. Communication flows between those groups are hardly present. This often leads to inefficient workarounds, undermining of other groups or even duplicate work [35]. Silos often arise from the organizational structure in a company, where people within a department only focus on their own job [35]. Silo thinking between business and IT can be broken down by introducing a third department called information communication [38].

This third department connects the business aspects to the IT or technology aspects by structuring the communication flow on three organizational levels: strategy, infrastructure and operations. The framework that arises with the introduction of a information communication column is shown in Figure 23. As can be seen, every junction between the organizational level and department is filled with a role. These roles all have their own responsibility in maintaining and developing the trading system, which is shown in Table 14.

In addition to the organizational structure with roles and responsibilities, a suggestion for the mindset within the organization is also given. As discussed with the live trading results, fast movements in market prices has been spotted in 2021. This indicates fast changing market structures within the electricity segment. Moreover, new electricity markets are introduced that require new technology and methods. Grid operators platform for congestion solutions

Table 14: Overview of roles and responsibilities regarding the recommendation for a future organizational structure.

Role	Responsibility
CEO	Determines the focus of optimization in the trading strategy. In this study, a strategy was selected on the highest Y_{PN}, with a profit minimum. The CEO determines future selection criteria.
CIO	Communication between the CEO and CIO, and the architects from all three departments. Desires of the CEO may not be technologically achievable, which should be communicated clearly. Furthermore, new policies from strategy level should be implemented in the structure level.
CTO	Look forward to new technologies to implement the trading strategy more effectively, efficiently and make future implementations possible.
Business architect	Determines which markets could lead to high profits or any other selection criteria that the CEO has set. This study only looked at two markets due to market restrictions, while a future implementation with more batteries could meet requirements from other markets too.
Information architect	Communication between the business architect and the technical architect, and the communication with operations. The electricity markets that are selected by the business architect have a great influence on the technical architecture. Moreover, the information architect is responsible for the usability of the actual implemented trading system.
Technical architect	Adjustments to the IT architecture that is shown in Figure 6 for this study. Adjustments are desirable if that leads to more stable or faster trading systems. Moreover, new markets could require adjustments too.
Business monitor	Checks whether the achieved results are in line with the expected results. Explains the difference in theoretically achievable results and the actual achieved results. An example in this study was given with the difference in theoretically achievable inverter power and actual inverter power.
Functional support	Handles the communication with the customer and between the business monitor and IT support. Complaints or improvements from customers or employees should be communicated with the architects.
IT support	Actual implementation of the IT architecture. This consist of the software and hardware to run the AI models and the software to steer the battery externally.

Table 15: Overview of roles and responsibilities for scrum. The responsibilities are adapted from Abrahamsson et al. [1].

Role	Responsibility
Product owner	Final responsibility for the whole software development. Determines the prioritization of the backlog. Keeping the development within budget.
Scrum master	Bridge between product owner and scrum team. Removes obstacles for software development, both within the team and due to external factors. Managing scrum framework.
Scrum team	Deliver increments of functional software within a sprint. Define when an increment is functional and finished. Keep track of progress and sprint planning.

is such a market that was first introduced in 2019 [58]. Therefore, a mindset that adapts to changing environments and allows fast and efficient software development is desirable. This can all be realized with agile, namely, one of the four main values from the manifesto is described as: "Responding to change over following a plan" [19]. Another value that is applicable in this study is the importance of working software. Instead of spending much time in documentation, time is spent in getting a software product to work. This is beneficial for future implementations, since fast changing electricity markets may only be profitable for a short period of time.

Several methods are available to implement agile, such as extreme programming, scrum and crystal [1]. For this study, several requirements to the method could lead to better performances. First of all, the implementation of future requirements need a high level planning. Even though fast changing markets could change the requirements in the near future, a planning that covers the current requirements is helpful. The difference in requirements with more traditional project management is that they come from all stakeholders, such as the customer, IT support or architects. Secondly, planning variables such as resources, time frame and tools should be determined regularly, instead of only at the beginning. Changing market environments may require new hardware to implement software, or a requirement may take much time to implement. These requirements for a method are enclosed in scrum [1]. Planning and re-planning is achieved with a backlog, where requirements are prioritized. Re-planning is possible due to so-called sprints, a period of two to four weeks in which a functional increment is added to the product. The advantage of these sprints is that a functional software could already be implemented. Furthermore, if market conditions change, requirements are adjusted within one sprint. Another advantage with scrum is testing. In each sprint, software is tested carefully before it is defined as functional. This could prevent an error that stops the battery from discharging which was encountered in this study by a jump in BT_{soc}.

Scrum also introduces new roles and responsibilities to software development. These include the product owner, scrum master and scrum team. Optimally, the scrum team consists of employees from multiple departments and layers. From the structure recommendation in Table 14, this would be employees from the business monitor, functional support, IT support and three architects from business, information communication and technology. The scrum master could be one person from the same six clusters as the scrum team. Finally, the product owner would be one person from higher management. This could be the CIO, which has both knowledge of the business and technology side. The responsibilities for the scrum roles is shown in Table 15.

One method to prioritize the requirements from the backlog is with MoSCoW, where each requirements is labeled as must have, should have, could have or will not have this time [3]. For must have requirements, the requirements must be implemented in the current sprint. Should have requirements, however, should also be implemented in the final product but may be delayed to a future sprint if there is no time left. Could have requirements are only implemented when there is much time left and little effort is needed. Finally, will not have this time requirements will not be implemented in the current release of the final product. They are still added to the backlog to let the stakeholders know that the requirement is taken into consideration. A recommendation for a future planning with the scrum and MoSCoW methodology is given in the conclusion, when future work is identified.

6. Conclusion

Due to an increasing generation of renewable energy, a change in electricity provisioning is required. One solution is a BESS, where locally generated solar energy is temporarily stored. This leads to smaller peaks in the power grid and solves the mismatch between supply and demand. However, the costs of a BESS are often high, which makes the investment less interesting. Therefore, this study tried to find a trading model based on electricity price forecasting with AI principles to increase the profit. In order to find that model, several questions were defined.

First of all, two Dutch electricity markets were investigated. The most volume is traded on the day-ahead market. In here, participants should submit their bids one day before delivery. The bids and offers settle to a single price for every hour of the delivery day. Although the day-ahead bidding limits the usage of real-time information, the market is suited for a trading strategy. Especially since the price fluctuates often in one day, while adhering to a certain pattern. That pattern was not shown in the imbalance market. In this market, real-time imbalance between supply and demand is settled. The imbalance price reflects the imbalance of 15 minutes and is determined afterwards. This market also showed potential for a trading strategy, due to its large fluctuations and simple market access. Furthermore, the range of imbalance prices is large compared to its mean.

Secondly, four AI principles were selected to apply in electricity price forecasting. Two of them, ANN and SVR, were often seen in literature on electricity price forecasting. Interestingly, these two models were only applied on the day-ahead price forecasting in several countries. The imbalance market was never discussed. The comparison in this study is also made with two more recent AI principles, namely, RF and GB. The difference between these four models is the architecture and algorithms to forecast the electricity prices. All principles were trained with supervised learning. However, ANN builds a network of nodes that are activated if a certain threshold is met. SVR on the other hand, minimizes a non linear function while keeping below a certain error margin. Both RF and GB build multiple decision trees, whereas RF takes an average of the individual outcomes and GB builds the individual trees based on the predecessor trees.

Thirdly, most related work do not optimize the hyperparameters of the AI principles extensively. Some try a few different settings, while others only use fixed hyperparameters. This study improves the AI principles with an extensive search of the best performing hyperparameters. This is achieved by 100 iterations of random search in a comprehensive configuration space, while using 5-fold cross validation to prevent overfitting. For the day-ahead market, this led to an MAE of 8.85 and an MSE of 143.65 with GB; the best performing AI principle in the day-ahead market. Unfortunately, the simple baseline forecaster outperformed all AI principles. One explanation is the COVID-19 pandemic, which caused demand differences and price drops. The imbalance price forecasting on the other hand, showed more promising results. All AI principles outperform the simple baseline P_{mid} with ease in terms of both MAE and MSE, even with a small T_s. ANN is the best performing AI approach over all T_s, although the differences are insignificant. As expected, an increasing T_s leads to a decreasing forecasting error. The best performing forecasting model in terms of MAE achieved a remarkable score of 15.72, while the best performing model in terms of MSE achieved a score of 1982.65 at $T_s = 7$. Ibti, the other baseline method, was only outperformed in terms of MSE. That indicates that Ibti is often close to the actual values, except when large price fluctuations occur. EP on the other hand, is often outperformed by the AI principles if T_s is larger than 2.

Fourthly, several trading strategies are tested and optimized in their parameters. These strategies had three base scenarios. B_1 and B_2 did not include the battery and bought the net user consumption from the day-ahead and imbalance market, respectively. B_3 settled the net consumption on the day-ahead market, while using the battery for solar generated energy if the production was larger than the consumption. From the baseline methods, B_3 performed the best, although the difference with B_1 is small. The three general trading strategies included the forecasts of the best performing AI principle. The strategies showed that the imbalance market has the most potential for a trading strategy. That strategy, G_2, has three parameters that affect the profit and number of cycles, namely, T_s, X^c_{imb}, and X^d_{imb}. T_s shows an optimum around $T_s = 4$ in profits, while the Y_{PN} keeps increasing with a larger T_s. Furthermore, while an increasing X^c_{imb} leads to higher profits, the Y_{PN} decreases. Therefore, both outcomes need to be considered when selecting the best performing parameters. The highest profit achieved in one year is 650 EUR, using a 15 kWh battery with 10 kW power. Compared to a profit of 998 EUR when there are no errors in the forecasts, the forecasts are performing well in the trading strategy. However, that trading strategy is not classified as best performing. That is due to the large N_{BT} and small Y_{PN}. Since the number of cycles is also important, the trading strategy that is classified as best achieves a profit of 521 EUR and an Y_{PN} of 0.387.

Finally, the best performing trading strategy is implemented in an organization, which led to the following results. Both MAE and MSE showed a decrease in performance in two weeks of live trading. It is highly likely that the worse performance is achieved due to high fluctuations in imbalance prices. Although the AI models performed worse, the profit and Y_{PN} have increased. On average, the profit would have increased with a multiplier of 1.45 in one year, compared to the best performing static model. Moreover, Y_{PN} has increased with a multiplier of 2.2. Although the live implementation showed much potential and outperformed the static models, a few problems were encountered. Firstly, the theoretically charging power was not met. Secondly, a mismatch between actual and received values in combination with an error in the software application led to missed profit. Thirdly, a software architecture that depends on multiple parties showed instability, causing a shorter time period to run the live implementation. When these problems are mitigated, the live implementation could perform even better and a conclusion can be drawn over a longer time span.

All in all, two AI principles showed much promising results in forecasting electricity prices, namely, GB and ANN. Even though GB is less used by previous studies, its potential is discovered. Furthermore, the AI principles achieve high profits on the imbalance market. A real-world implementation also showed that the static models are applicable on real-time electricity markets in combination with a BESS. Therefore, an arbitrage is possible on Dutch electricity markets by trading energy from a BESS using price forecasts from GB.

6.1. Future work

Although the results look promising already, a few recommendations are given. First of all, the input of the imbalance forecasting models could be expanded. Currently, the sum, maximum or minimum price is taken from all T_s values, leading to omitted information. An improvement could be the separation of the input variables for every T_s. For example, $IGCC_{up}$ would have five values if $T_s = 5$. Furthermore, more markets can be added to the forecasting and trading models. In this study, the day-ahead and the imbalance markets were researched, where the battery can only be active in one market at a time. However, some markets allow for value stacking, i.e., the battery is active in two or more markets, while the electricity from the battery is used in at most one market. The Dutch frequency containment reserves (FCR) market is such an example. Here, a compensation is given when electricity is offered, even if the electricity is not consumed. However, a fine is received if the offer is activated and the supplier fails to deliver the electricity. This could happen if the electricity is then used in another market or the battery is empty due to other markets. Therefore, the forecasting model could change the target

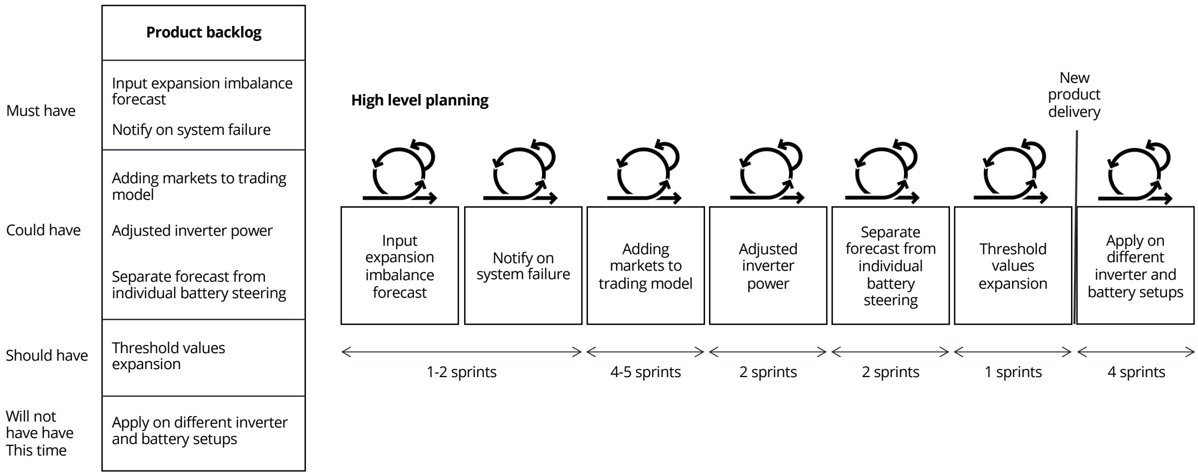


Figure 24: Road map of future work with scrum and MoSCoW methodology. Scrum methods allow deviation from the planning and is checked regularly due to incremental workflows.

to the activation of the FCR. Another recommendation can be made at the trading strategies. In this study, the power of the battery to charge and discharge is fixed to 10kW. However, the power is adjustable and could depend on the forecasted price. If the forecasted price exceeds the threshold with a small amount, the power is set to a smaller level than in times the forecasted price exceeds the threshold with ease. This could ensure that the battery is not at its maximum or minimum capacity when extremely low or high prices occur, respectively. Furthermore, the threshold values for the strategies could also be further investigated. Currently, both X^c and X^d decrease and increase at the same time, respectively. The strategies may improve if these values are not connected, i.e., one parameter is relatively far apart from value one, while the other is not.

Improvements for live implementation within organizations are also possible. In this study, the software application handles both the market signals with their forecasts and the interaction with the battery. In a setup with multiple batteries, market signals and forecasts should be separated from the battery interaction, since the former is only required once, while the latter differs per battery. What architecture is needed for this setup could be researched further. This architecture could also include an alarming system to notify the maintainer when the system is down. Furthermore, the influence of difference between theoretically achievable power from the inverter and the actual power could be researched more in-depth.

Besides the improvement for this research, it is important to take into account the threats as well. One threat is a change in imbalance market prices, such that an arbitrage is less profitable. The live implementation showed that the market changes over time. In 2021, this led to higher profits. However, it is also possible that the market changes the other way around. Then, there are less extreme prices and an arbitrage is less profitable. Another threat is the scalability of this study. If the number of batteries is limited, the batteries do not contribute to the imbalance price. However, the price could be influenced when many batteries participate. The effect of scalability can be further researched.

Finally, a recommendation for a planning is given with the scrum and MoSCoW methodology, based on the future work that could be done. The planning is given for a team of 6 people with a sprint period of two weeks. At the end of a sprint, a functional increment should be added. When a requirement is finished after a given number of sprints, it should be added to the product. Both the sprint deliveries and the requirement deliveries should be tested thoroughly. A planning recommendation is given in Figure 24.

References

[1] Pekka Abrahamsson et al. 'Agile software development methods: Review and analysis'. In: *arXiv preprint arXiv:1709.08439* (2017).

[2] Rahul Kumar Agrawal, Frankle Muchahary and Madan Mohan Tripathi. 'Ensemble of relevance vector machines and boosted trees for electricity price forecasting'. In: *Applied Energy* 250 (2019), pp. 540–548.

[3] Khadija Sania Ahmad et al. 'Fuzzy_MoSCoW: A fuzzy based MoSCoW method for the prioritization of software requirements'. In: *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*. IEEE. 2017, pp. 433–437.

[4] Nima Amjady. 'Day-ahead price forecasting of electricity markets by a new fuzzy neural network'. In: *IEEE Transactions on power systems* 21.2 (2006), pp. 887–896.

[5] Nima Amjady, A Daraeepour and F Keynia. 'Day-ahead electricity price forecasting by modified relief algorithm and hybrid neural network'. In: *IET generation, transmission & distribution* 4.3 (2010), pp. 432–444.

[6] Anne-Marie Anderson and Edward A Dyl. 'Market structure and trading volume'. In: *Journal of Financial Research* 28.1 (2005), pp. 115–131.

[7] James Bergstra and Yoshua Bengio. 'Random search for hyper-parameter optimization.' In: *Journal of machine learning research* 13.2 (2012).

[8] Sanne de Boer. *De energietransitie uitgelegd*. De Groene Waterlelie, 2020.

[9] Leo Breiman. 'Random forests'. In: *Machine learning* 45.1 (2001), pp. 5–32.

[10] CBS. *Electricity and heat; production and input by energy commodity*. 2020. URL: https://opendata.cbs.nl/statline/#/CBS/en/dataset/80030eng/table?ts=1618917289949 (visited on 30/03/2021).

[11] Najeh Chaâbane. 'A hybrid ARFIMA and neural network model for electricity price prediction'. In: *International Journal of Electrical Power & Energy Systems* 55 (2014), pp. 187–194.

[12] Chih-Chung Chang and Chih-Jen Lin. 'LIBSVM: a library for support vector machines'. In: *ACM transactions on intelligent systems and technology (TIST)* 2.3 (2011), pp. 1–27.

[13] Tianmei Chen et al. 'Applications of lithium-ion batteries in grid-scale energy storage systems'. In: *Transactions of Tianjin University* 26.3 (2020), pp. 208–217.

[14] Alberto Cruz et al. 'The effect of wind generation and weekday on Spanish electricity spot price forecasting'. In: *Electric Power Systems Research* 81.10 (2011), pp. 1924–1935.

[15] Giovani Dematos et al. 'Feedforward versus recurrent neural networks for forecasting monthly japanese yen exchange rates'. In: *Financial Engineering and the Japanese Markets* 3.1 (1996), pp. 59–75.

[16] Janez Demšar. 'Statistical comparisons of classifiers over multiple data sets'. In: *Journal of Machine Learning Research* 7.Jan (2006), pp. 1–30.

[17] Paul Ekins. 'Rethinking the costs related to global warming: A survey of the issues'. In: *Environmental and Resource Economics* 6.3 (1995), pp. 231–277.

[18] Entsoe. *Transparency Platform*. 2021. URL: https://transparency.entsoe.eu/ (visited on 01/04/2021).

[19] Martin Fowler, Jim Highsmith et al. 'The agile manifesto'. In: *Software development* 9.8 (2001), pp. 28–35.

[20] Jerome H Friedman. 'Greedy function approximation: a gradient boosting machine'. In: *Annals of statistics* (2001), pp. 1189–1232.

[21] Gasunie. *Verkenning 2050*. 2018. URL: `https://www.gasunie.nl/expertise/aardgas/energiemix-2050/$3170/$3171` (visited on 30/03/2021).

[22] Alicia Mateo Gonzalez, AM Son Roque and Javier Garcia-Gonzalez. 'Modeling and forecasting electricity prices with input/output hidden Markov models'. In: *IEEE Transactions on Power Systems* 20.1 (2005), pp. 13–24.

[23] Ying-Chun Guo, Dong-xiao Niu and Yan-Xu Chen. 'Support vector machine model in electricity load forecasting'. In: *2006 International Conference on Machine Learning and Cybernetics*. IEEE. 2006, pp. 2892–2896.

[24] Stephanie Halbrügge et al. 'How did the German and other European electricity systems react to the COVID-19 pandemic?' In: *Applied Energy* 285 (2021), p. 116370.

[25] Veronika Henze. *Battery Pack Prices Fall As Market Ramps Up With Market Average At $156/kWh In 2019*. 2019. URL: `https://about.bnef.com/blog/battery-pack-prices-fall-as-market-ramps-up-with-market-average-at-156-kwh-in-2019/` (visited on 30/03/2021).

[26] Peter Huitsing et al. 'Attack taxonomies for the Modbus protocols'. In: *International Journal of Critical Infrastructure Protection* 1 (2008), pp. 37–44.

[27] Hussein Ibrahim, Adrian Ilinca and Jean Perron. 'Energy storage systems—Characteristics and comparisons'. In: *Renewable and sustainable energy reviews* 12.5 (2008), pp. 1221–1250.

[28] IEA. *Data and statistics; Energy Supply*. 2020. URL: `https://www.iea.org/data-and-statistics` (visited on 30/03/2021).

[29] Daniel R Jiang and Warren B Powell. 'Optimal hour-ahead bidding in the real-time electricity market with battery storage using approximate dynamic programming'. In: *INFORMS Journal on Computing* 27.3 (2015), pp. 525–543.

[30] Farshid Keynia. 'A new feature selection algorithm and composite neural network for electricity price forecasting'. In: *Engineering Applications of Artificial Intelligence* 25.8 (2012), pp. 1687–1697.

[31] Diederik P Kingma and Jimmy Ba. 'Adam: A method for stochastic optimization'. In: *arXiv preprint arXiv:1412.6980* (2014).

[32] Gro Klæboe, Anders Lund Eriksrud and Stein-Erik Fleten. 'Benchmarking time series based forecasting models for electricity balancing market prices'. In: *Energy Systems* 6.1 (2015), pp. 43–61.

[33] KNMI. *Data Platform*. 2021. URL: `https://dataplatform.knmi.nl/` (visited on 16/06/2021).

[34] Ron Kohavi and George H John. 'Wrappers for feature subset selection'. In: *Artificial intelligence* 97.1-2 (1997), pp. 273–324.

[35] Meghan Kowalski. 'Breaking down silo walls: Successful collaboration across library departments'. In: *Library Leadership & Management* 31.2 (2017).

[36] Ali Lahouar and Jaleleddine Ben Hadj Slama. 'Random forests model for one day ahead load forecasting'. In: *IREC2015 The Sixth International Renewable Energy Congress*. IEEE. 2015, pp. 1–6.

[37] Nicole Ludwig, Stefan Feuerriegel and Dirk Neumann. 'Putting Big Data analytics to work: Feature selection for forecasting electricity prices using the LASSO and random forests'. In: *Journal of Decision Systems* 24.1 (2015), pp. 19–36.

[38] Rik Maes. *A generic framework for information management*. Universiteit van Amsterdam, Department of Accountancy & Information ..., 1999.

[39] Edward Maibach et al. 'Health professionals, the Paris agreement, and the fierce urgency of now'. In: *The Journal of Climate Change and Health* 1 (2020), p. 100002.

[40] P Manimekalai, R Harikumar and S Raghavan. 'An overview of batteries for photovoltaic (PV) systems'. In: *International Journal of Computer Applications* 82.12 (2013).

[41] Machiel Mulder and Bert Scholtens. 'The impact of renewable energy on electricity prices in the Netherlands'. In: *Renewable energy* 57 (2013), pp. 94–100.

[42] Bijay Neupane et al. 'Artificial neural network-based electricity price forecasting for smart grid deployment'. In: *2012 International Conference on Computer Systems and Industrial Informatics*. IEEE. 2012, pp. 1–6.

[43] MSH Nizami et al. 'Transactive energy trading of residential prosumers using battery energy storage systems'. In: *2019 IEEE Milan PowerTech*. IEEE. 2019, pp. 1–6.

[44] F. Pedregosa et al. 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[45] Jeroen Peters. 'Modeling the Dutch Frequency Restoration Reserve Market'. In: (2016).

[46] Javier Pórtoles, Camino González and Javier M Moguerza. 'Electricity price forecasting with dynamic trees: a benchmark against the random forest approach'. In: *Energies* 11.6 (2018), p. 1588.

[47] Damien C Sansom, Tom Downs, Tapan K Saha et al. 'Evaluation of support vector machine based forecasting tool in electricity price forecasting for Australian national electricity market participants'. In: *Journal of Electrical & Electronics Engineering, Australia* 22.3 (2003), p. 227.

[48] Annalisa Savaresi. 'The Paris Agreement: a new beginning?' In: *Journal of Energy & Natural Resources Law* 34.1 (2016), pp. 16–26.

[49] Alex J Smola and Bernhard Schölkopf. 'A tutorial on support vector regression'. In: *Statistics and computing* 14.3 (2004), pp. 199–222.

[50] Soluzon. *RE-USE: REgenerative Utility of Saved Energy*. 2021. URL: `https://research.tue.nl/en/projects/re-use-regenerative-utility-of-saved-energy` (visited on 19/07/2021).

[51] Dipa Soni and Ashwin Makwana. 'A survey on mqtt: a protocol of internet of things (iot)'. In: *International Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017)*. Vol. 20. 2017.

[52] Ugur Soytas and Ramazan Sari. 'Can China contribute more to the fight against global warming?' In: *Journal of Policy Modeling* 28.8 (2006), pp. 837–846.

[53] EPEX SPOT. *EPEX SPOT Annual Market Review 2020*. 2021. URL: `https://www.eex-group.com/fileadmin/EEX_Group/News-Reports/20211401_EPEX_SPOT_Annual_Press_Release-2020.pdf` (visited on 15/04/2021).

[54] Stedin. *Tarieven Grootzakelijk*. 2021. URL: `https://www.stedin.net/zakelijk/betalingen-en-facturen/tarieven` (visited on 30/03/2021).

[55] Souhaib Ben Taieb and Rob J Hyndman. 'A gradient boosting approach to the Kaggle load forecasting competition'. In: *International journal of forecasting* 30.2 (2014), pp. 382–394.

[56] Fehmi Tanrisever, Kursad Derinkuyu and Michael Heeren. 'Forecasting electricity infeed for distribution system networks: An analysis of the Dutch case'. In: *Energy* 58 (2013), pp. 247–257.

[57] Fehmi Tanrisever, Kursad Derinkuyu and Geert Jongen. 'Organization and functioning of liberalized electricity markets: An overview of the Dutch market'. In: *Renewable and Sustainable Energy Reviews* 51 (2015), pp. 1363–1374.

[58] TenneT. *Annual Market Update 2020; Electricity market insights*. 2020. URL: https://www.tennet.eu/fileadmin/user_upload/Company/Publications/Technical_Publications/Annual_Market_Update_2020.pdf (visited on 17/09/2021).

[59] TenneT. *Balance delta IGCC*. 2021. URL: https://www.tennet.org/english/operational_management/System_data_relating_implementation/system_balance_information/balancedeltaIGCC.aspx (visited on 11/06/2021).

[60] TenneT. *Bid price ladder of* $\leq$ *15 minutes*. 2021. URL: https://www.tennet.org/english/operational_management/system_data_preparation/offering_regulating_reserve_capacity/bid_price_ladder.aspx (visited on 16/04/2021).

[61] TenneT. *Imbalance Pricing System*. 2020. URL: https://www.tennet.eu/fileadmin/user_upload/SO_NL/Imbalance_pricing_system.pdf (visited on 21/04/2021).

[62] TenneT. *Settlement prices*. 2021. URL: https://www.tennet.org/english/operational_management/System_data_relating_processing/settlement_prices/index.aspx (visited on 01/04/2021).

[63] Nils R. Terpstra. *Day-ahead and imbalance price forecasting on the Dutch Electricity Market: a comparison between time series and artificial neural networks models*. 2020. URL: https://research.tue.nl/en/studentTheses/day-ahead-and-imbalance-price-forecasting-on-the-dutch-electricit (visited on 23/04/2021).

[64] Rafał Weron. 'Electricity price forecasting: A review of the state-of-the-art with a look into the future'. In: *International journal of forecasting* 30.4 (2014), pp. 1030–1081.

[65] Rafał Weron, Michael Bierbrauer and Stefan Trück. 'Modeling electricity prices: jump diffusion and regime switching'. In: *Physica A: Statistical Mechanics and its Applications* 336.1-2 (2004), pp. 39–48.

[66] WordWeatherOnline. *Weather API and Data for Businesses and Developers*. 2021. URL: https://www.worldweatheronline.com/developer/ (visited on 26/04/2021).

[67] HY Yamin, SM Shahidehpour and Zuyi Li. 'Adaptive short-term electricity price forecasting using artificial neural networks in the restructured power markets'. In: *International journal of electrical power & energy systems* 26.8 (2004), pp. 571–581.

[68] Jun Hua Zhao et al. 'A statistical approach for interval forecasting of the electricity price'. In: *IEEE Transactions on Power Systems* 23.2 (2008), pp. 267–276.

Appendix

A. Hyperparameter configuration spaces

Hyperparameter configuration space for 100 iterations of random search with ANN. The hyperparameters follow the definition from Scikit-learn [44].

hyperparameter	lower bound	upper bound	default value	log
alpha	$1e-7$	$1e-1$	$1e-4$	true
learning_rate_init	$1e-4$	0.5	$1e-3$	true
hyperparameter	value			
n_iter_no_change	32			
validation_fraction	0.1			
tol	$1e-4$			
solver	$'adam'$			
hyperparameter	choices		default	
hidden_layer_sizes	(16), (32), (64), (128), (256), (16,16,16), (32,32,32), (64,64,64), (128,128,128), (256,256,256)		(32)	
activation	$'tanh', 'relu'$		$'tanh'$	
early_stopping	$true, false$		$true$	

Hyperparameter configuration space for 100 iterations of random search with SVR. The hyperparameters follow the definition from Scikit-learn [44].

hyperparameter	lower bound	upper bound	default value	log
C	0.03125	32768	1	true
epsilon	0.001	1	0.1	true
degree	2	5	3	true
gamma	$3.0517578125e-05$	8	0.1	true
$coef0$	-1	1	0	
tol	$1e-5$	$1e-1$	$1e-3$	true
hyperparameter	value			
max_iter	-1			
hyperparameter	choices		default	
kernel	$'rbf', 'sigmoid'$		$'rbf'$	
shrinking	$true, false$		$true$	

Hyperparameter configuration space for 100 iterations of random search with RF. The hyperparameters follow the definition from Scikit-learn [44].

hyperparameter	lower bound	upper bound	default value
$max\_features$	0.1	1	1
$min\_samples\_split$	2	20	2
$min\_samples\_leaf$	1	20	1
hyperparameter	value		
$min\_weight\_fraction\_leaf$	0.		
$min\_impurity\_decrease$	0.0		
hyperparameter	choices		default
$bootstrap$	$true, false$		$true$

Hyperparameter configuration space for 100 iterations of random search with GB. The hyperparameters follow the definition from Scikit-learn [44].

hyperparameter	lower bound	upper bound	default value	log
$learning\_rate$	0.01	1	0.1	true
$min\_samples\_leaf$	1	200	20	true
max_leaf_nodes	3	2047	31	true
$n\_iter\_no\_change$	1	20	10	
$validation_fraction$	0.01	0.4	0.1	
hyperparameter	value			
tol	$1e-7$			
hyperparameter	choices		default	
$loss$	$'ls','lad','huber'$		$'ls'$	

B. Trading results

An overview of the performances with the trading models using the forecast of the best performing AI approaches. The profit is calculated by subtracting the electricity costs of the strategy from the electricity costs of base line B_1. The numbers in brackets indicate the performance that could be achieved if the forecast was equal to the actual value.

(a)

Strategy	T_s	X_{da}^c	X_{da}^d	X_{imb}^c	X_{imb}^d	X_{da}^{max}	X_{da}^{min}	Electricity costs [perfect]	Profit [perfect]	N_{BT} [perfect]	Y_{PN} [perfect]
B_1	-	-	-	-	-	-	-	-5672.04	-	-	
B_2	-	-	-	-	-	-	-	-7648.67	-1976.63	-	-
B_3	-	-	-	-	-	-	-	-5661.07	10.97	790	0.014
G_1	-	0.99	1.01	-	-	-	-	-5655.37 [-5627.84]	16.67 [44.2]	1350 [1398]	0.012 [0.032]
	-	0.9	1.1	-	-	-	-	-5632.19 [-5612.53]	39.85 [59.51]	756 [1106]	0.053 [0.054]
	-	0.8	1.2	-	-	-	-	-5643.09 [-5604.85]	28.95 [67.19]	308 [818]	0.094 [0.082]
	-	0.7	1.3	-	-	-	-	-5662.97 [-5617.51]	9.07 [54.53]	70 [510]	0.129 [0.107]
	-	0.6	1.4	-	-	-	-	-5670.36 [-5634.6]	1.68 [37.44]	6 [294]	0.258 [0.127]
	-	0.5	1.5	-	-	-	-	-5671.79 [-5649.28]	0.25 [22.76]	0 [168]	0.492 [0.135]
G_2	2	-	-	0.99	1.01	-	-	-5035.95 [-4673.74]	636.09 [998.3]	3449 [2671]	0.184 [0.374]
		-	-	0.9	1.1	-	-	-5045.57 [-4680.83]	626.47 [991.21]	3004 [2242]	0.209 [0.442]
		-	-	0.8	1.2	-	-	-5077.45 [-4698.89]	594.59 [973.15]	2582 [1897]	0.23 [0.513]
		-	-	0.7	1.3	-	-	-5120.49 [-4710.83]	551.55 [961.21]	2178 [1669]	0.253 [0.576]
		-	-	0.6	1.4	-	-	-5169.75 [-4752.95]	502.29 [919.09]	1765 [1477]	0.285 [0.622]
		-	-	0.5	1.5	-	-	-5221.15 [-4812.12]	450.89 [859.92]	1352 [1297]	0.333 [0.663]
	3	-	-	0.99	1.01	-	-	-5040.55 [-4753.03]	631.49 [919.01]	3146 [2465]	0.201 [0.373]
		-	-	0.9	1.1	-	-	-5060.49 [-4757.61]	611.55 [914.43]	2685 [2072]	0.228 [0.441]
		-	-	0.8	1.2	-	-	-5076.28 [-4774.04]	595.76 [898]	2271 [1755]	0.262 [0.512]
		-	-	0.7	1.3	-	-	-5103.45 [-4785.79]	568.59 [886.25]	1949 [1546]	0.292 [0.573]
		-	-	0.6	1.4	-	-	-5124.4 [-4820.84]	547.64 [851.2]	1679 [1371]	0.326 [0.621]
		-	-	0.5	1.5	-	-	-5151.61 [-4870.4]	520.43 [801.64]	1455 [1209]	0.358 [0.663]

(b)

Strategy	T_s	X_{da}^c	X_{da}^d	X_{imb}^c	X_{imb}^d	X_{da}^{max}	X_{da}^{min}	Electricity costs [perfect]	Profit [perfect]	N_{BT} [perfect]	Y_{PN} [perfect]
G_2	4	-	-	0.99	1.01	-	-	-5021.67 [-4837.58]	650.37 [834.46]	2820 [2247]	0.231 [0.371]
		-	-	0.9	1.1	-	-	-5041.16 [-4838.8]	630.88 [833.24]	2441 [1893]	0.258 [0.44]
		-	-	0.8	1.2	-	-	-5062.72 [-4856.29]	609.32 [815.75]	2114 [1603]	0.288 [0.509]
		-	-	0.7	1.3	-	-	-5080.86 [-4866.57]	591.18 [805.47]	1837 [1414]	0.322 [0.57]
		-	-	0.6	1.4	-	-	-5109.19 [-4893.9]	562.85 [778.14]	1593 [1258]	0.353 [0.619]
		-	-	0.5	1.5	-	-	-5148.23 [-4935.01]	523.81 [737.03]	1378 [1112]	0.38 [0.663]
	5	-	-	0.99	1.01	-	-	-5081.59 [-4923.11]	590.45 [748.93]	2466 [2023]	0.239 [0.37]
		-	-	0.9	1.1	-	-	-5091.28 [-4925.28]	580.76 [746.76]	2129 [1705]	0.273 [0.438]
		-	-	0.8	1.2	-	-	-5107.14 [-4938.8]	564.9 [733.24]	1826 [1446]	0.309 [0.507]
		-	-	0.7	1.3	-	-	-5126.68 [-4945.6]	545.36 [726.44]	1564 [1276]	0.349 [0.569]
		-	-	0.6	1.4	-	-	-5151.05 [-4966.65]	520.99 [705.39]	1347 [1139]	0.387 [0.619]
		-	-	0.5	1.5	-	-	-5178.13 [-5000.01]	493.91 [672.03]	1167 [1012]	0.423 [0.664]
	6	-	-	0.99	1.01	-	-	-5123.22 [-5011.59]	548.82 [660.45]	2224 [1788]	0.247 [0.369]
		-	-	0.9	1.1	-	-	-5134 [-5014.2]	538.04 [657.84]	1888 [1509]	0.285 [0.436]
		-	-	0.8	1.2	-	-	-5147.66 [-5025.73]	524.38 [646.31]	1603 [1281]	0.327 [0.504]
		-	-	0.7	1.3	-	-	-5164.7 [-5029.7]	507.34 [642.34]	1365 [1131]	0.372 [0.568]
		-	-	0.6	1.4	-	-	-5185.95 [-5043.73]	486.09 [628.31]	1165 [1013]	0.417 [0.62]
		-	-	0.5	1.5	-	-	-5212.55 [-5071.29]	459.49 [600.75]	999 [902]	0.46 [0.666]
	7	-	-	0.99	1.01	-	-	-5167 [-5106.86]	505.04 [565.18]	1883 [1540]	0.268 [0.367]
		-	-	0.9	1.1	-	-	-5176.41 [-5110.18]	495.63 [561.86]	1612 [1301]	0.307 [0.432]
		-	-	0.8	1.2	-	-	-5192.1 [-5119.05]	479.94 [552.99]	1364 [1106]	0.352 [0.5]
		-	-	0.7	1.3	-	-	-5206.67 [-5123.07]	465.37 [548.97]	1163 [975]	0.4 [0.563]
		-	-	0.6	1.4	-	-	-5216.4 [-5131.26]	455.64 [540.78]	1005 [876]	0.454 [0.617]
		-	-	0.5	1.5	-	-	-5235.84 [-5150.66]	436.2 [521.38]	857 [783]	0.509 [0.666]
G_3	2	0.99	1.01	0.99	1.01	2	2	-5113.46 [-5014.46]	558.58 [657.58]	3584 [2962]	0.156 [0.222]
		0.99	1.01	0.99	1.01	3	3	-5158.59 [-5141.51]	513.45 [530.53]	3400 [2671]	0.151 [0.199]
		0.9	1.1	0.9	1.1	2	2	-5119.89 [-5018.52]	552.15 [653.52]	3133 [2419]	0.176 [0.27]
		0.9	1.1	0.9	1.1	3	3	-5164.33 [-5143.72]	507.71 [528.32]	2960 [2149]	0.172 [0.246]
		0.8	1.2	0.8	1.2	2	2	-5141.37 [-5028.33]	530.67 [643.71]	2546 [1970]	0.208 [0.327]
		0.8	1.2	0.8	1.2	3	3	-5186.41 [-5144.93]	485.63 [527.11]	2371 [1731]	0.205 [0.304]

(c)

Strategy	T_s	X_{da}^{c}	X_{da}^{d}	X_{imb}^{c}	X_{imb}^{d}	X_{da}^{max}	X_{da}^{min}	Electricity costs [perfect]	Profit [perfect]	N_{BT} [perfect]	Y_{PN} [perfect]
G_3	2	0.7	1.3	0.7	1.3	2	2	-5181.6 [-5033.8]	490.44 [638.24]	1983 [1600]	0.247 [0.399]
		0.7	1.3	0.7	1.3	3	3	-5223.95 [-5150.22]	448.09 [521.82]	1826 [1372]	0.245 [0.38]
		0.6	1.4	0.6	1.4	2	2	-5235.09 [-5058.75]	436.95 [613.29]	1540 [1280]	0.284 [0.479]
		0.6	1.4	0.6	1.4	3	3	-5272.12 [-5163.87]	399.92 [508.17]	1413 [1089]	0.283 [0.467]
		0.5	1.5	0.5	1.5	2	2	-5276.08 [-5103.02]	395.96 [569.02]	1165 [1003]	0.34 [0.567]
		0.5	1.5	0.5	1.5	3	3	-5310.14 [-5197.03]	361.9 [475.01]	1062 [847]	0.341 [0.561]
	3	0.99	1.01	0.99	1.01	2	2	-5094.55 [-5060.87]	577.49 [611.17]	3333 [2814]	0.173 [0.217]
		0.99	1.01	0.99	1.01	3	3	-5149.08 [-5178.7]	522.96 [493.34]	3167 [2553]	0.165 [0.193]
		0.9	1.1	0.9	1.1	2	2	-5105.52 [-5064.24]	566.52 [607.8]	2877 [2305]	0.197 [0.264]
		0.9	1.1	0.9	1.1	3	3	-5161.14 [-5179.84]	510.9 [492.2]	2719 [2062]	0.188 [0.239]
		0.8	1.2	0.8	1.2	2	2	-5124.94 [-5074.71]	547.1 [597.33]	2281 [1879]	0.24 [0.318]
		0.8	1.2	0.8	1.2	3	3	-5178.19 [-5181.32]	493.85 [490.72]	2126 [1664]	0.232 [0.295]
		0.7	1.3	0.7	1.3	2	2	-5160.48 [-5078.54]	511.56 [593.5]	1762 [1523]	0.29 [0.39]
		0.7	1.3	0.7	1.3	3	3	-5212.92 [-5186.44]	459.12 [485.6]	1614 [1314]	0.284 [0.37]
		0.6	1.4	0.6	1.4	2	2	-5194.79 [-5099.76]	477.25 [572.28]	1443 [1215]	0.331 [0.471]
		0.6	1.4	0.6	1.4	3	3	-5234.69 [-5198.25]	437.35 [473.79]	1314 [1037]	0.333 [0.457]
		0.5	1.5	0.5	1.5	2	2	-5219.82 [-5140.25]	452.22 [531.79]	1233 [947]	0.367 [0.562]
		0.5	1.5	0.5	1.5	3	3	-5260.8 [-5228.17]	411.24 [443.87]	1122 [801]	0.367 [0.554]
	4	0.99	1.01	0.99	1.01	2	2	-5069.81 [-5110.09]	602.23 [561.95]	3025 [2657]	0.199 [0.212]
		0.99	1.01	0.99	1.01	3	3	-5124.34 [-5218.07]	547.7 [453.97]	2880 [2428]	0.19 [0.187]
		0.9	1.1	0.9	1.1	2	2	-5080.68 [-5112.67]	591.36 [559.37]	2657 [2182]	0.223 [0.256]
		0.9	1.1	0.9	1.1	3	3	-5135.52 [-5218.24]	536.52 [453.8]	2524 [1969]	0.213 [0.23]
		0.8	1.2	0.8	1.2	2	2	-5108.93 [-5121.8]	563.11 [550.24]	2163 [1782]	0.26 [0.309]
		0.8	1.2	0.8	1.2	3	3	-5161.82 [-5218.67]	510.22 [453.37]	2013 [1591]	0.253 [0.285]
		0.7	1.3	0.7	1.3	2	2	-5142.98 [-5126.1]	529.06 [545.94]	1677 [1443]	0.316 [0.378]
		0.7	1.3	0.7	1.3	3	3	-5191.96 [-5224.89]	480.08 [447.15]	1536 [1252]	0.313 [0.357]
		0.6	1.4	0.6	1.4	2	2	-5180.08 [-5145.1]	491.96 [526.94]	1370 [1144]	0.359 [0.461]
		0.6	1.4	0.6	1.4	3	3	-5221.32 [-5236.63]	450.72 [435.41]	1249 [979]	0.361 [0.445]
		0.5	1.5	0.5	1.5	2	2	-5209.08 [-5180.34]	462.96 [491.7]	1166 [886]	0.397 [0.555]
		0.5	1.5	0.5	1.5	3	3	-5249.6 [-5264.89]	422.44 [407.15]	1056 [750]	0.4 [0.543]

(d)

Strategy	T_s	X^c_{da}	X^d_{da}	X^c_{imb}	X^d_{imb}	X^{max}_{da}	X^{min}_{da}	Electricity costs [perfect]	Profit [perfect]	N_{BT} [perfect]	Y_{PN} [perfect]
G_3	5	0.99	1.01	0.99	1.01	2	2	-5118.2 [-5159.24]	553.84 [512.8]	2746 [2495]	0.202 [0.205]
		0.99	1.01	0.99	1.01	3	3	-5167.26 [-5257.67]	504.78 [414.37]	2633 [2301]	0.192 [0.18]
		0.9	1.1	0.9	1.1	2	2	-5125.44 [-5161.74]	546.6 [510.3]	2405 [2055]	0.227 [0.248]
		0.9	1.1	0.9	1.1	3	3	-5173.49 [-5256.41]	498.55 [415.63]	2296 [1874]	0.217 [0.222]
		0.8	1.2	0.8	1.2	2	2	-5148.93 [-5170.61]	523.11 [501.43]	1926 [1681]	0.272 [0.298]
		0.8	1.2	0.8	1.2	3	3	-5196.9 [-5257.7]	475.14 [414.34]	1797 [1516]	0.264 [0.273]
		0.7	1.3	0.7	1.3	2	2	-5180.29 [-5175.02]	491.75 [497.02]	1445 [1360]	0.34 [0.366]
		0.7	1.3	0.7	1.3	3	3	-5226.41 [-5264.64]	445.63 [407.4]	1317 [1188]	0.338 [0.343]
		0.6	1.4	0.6	1.4	2	2	-5221.35 [-5193.69]	450.69 [478.35]	1159 [1068]	0.389 [0.448]
		0.6	1.4	0.6	1.4	3	3	-5264.61 [-5277]	407.43 [395.04]	1050 [916]	0.388 [0.431]
		0.5	1.5	0.5	1.5	2	2	-5240.98 [-5223.84]	431.06 [448.2]	984 [822]	0.438 [0.545]
		0.5	1.5	0.5	1.5	3	3	-5282.02 [-5304.08]	390.02 [367.96]	893 [696]	0.437 [0.529]
	6	0.99	1.01	0.99	1.01	2	2	-5146.47 [-5214.01]	525.57 [458.03]	2528 [2320]	0.208 [0.197]
		0.99	1.01	0.99	1.01	3	3	-5193.88 [-5301.33]	478.16 [370.71]	2420 [2162]	0.198 [0.171]
		0.9	1.1	0.9	1.1	2	2	-5156.28 [-5214.91]	515.76 [457.13]	2186 [1921]	0.236 [0.238]
		0.9	1.1	0.9	1.1	3	3	-5200.11 [-5297.99]	471.93 [374.05]	2091 [1774]	0.226 [0.211]
		0.8	1.2	0.8	1.2	2	2	-5173.99 [-5222.41]	498.05 [449.63]	1726 [1574]	0.288 [0.286]
		0.8	1.2	0.8	1.2	3	3	-5217.49 [-5299.2]	454.55 [372.84]	1614 [1437]	0.282 [0.259]
		0.7	1.3	0.7	1.3	2	2	-5207.83 [-5226.67]	464.21 [445.37]	1276 [1270]	0.364 [0.351]
		0.7	1.3	0.7	1.3	3	3	-5252.24 [-5306.47]	419.8 [365.57]	1165 [1118]	0.36 [0.327]
		0.6	1.4	0.6	1.4	2	2	-5247.33 [-5245.59]	424.71 [426.45]	1003 [985]	0.423 [0.433]
		0.6	1.4	0.6	1.4	3	3	-5283.56 [-5320.22]	388.48 [351.82]	912 [849]	0.426 [0.415]
		0.5	1.5	0.5	1.5	2	2	-5271.29 [-5272.14]	400.75 [399.9]	839 [752]	0.478 [0.532]
		0.5	1.5	0.5	1.5	3	3	-5306.04 [-5345.68]	366 [326.36]	761 [637]	0.481 [0.512]
	7	0.99	1.01	0.99	1.01	2	2	-5179 [-5270.91]	493.04 [401.13]	2249 [2137]	0.219 [0.188]
		0.99	1.01	0.99	1.01	3	3	-5224.03 [-5347.11]	448.01 [324.93]	2169 [2018]	0.207 [0.161]
		0.9	1.1	0.9	1.1	2	2	-5190.08 [-5269.71]	481.96 [402.33]	1957 [1780]	0.246 [0.226]
		0.9	1.1	0.9	1.1	3	3	-5234.36 [-5342.52]	437.68 [329.52]	1888 [1667]	0.232 [0.198]
		0.8	1.2	0.8	1.2	2	2	-5209.56 [-5275.87]	462.48 [396.17]	1535 [1462]	0.301 [0.271]
		0.8	1.2	0.8	1.2	3	3	-5249.23 [-5343.01]	422.81 [329.03]	1443 [1350]	0.293 [0.244]

(e)

Strategy	T_s	X^c_{da}	X^d_{da}	X^c_{imb}	X^d_{imb}	X^{max}_{da}	X^{min}_{da}	Electricity costs [perfect]	Profit [perfect]	N_{BT} [perfect]	Y_{PN} [perfect]
G_3	7	0.7	1.3	0.7	1.3	2	2	-5250.62 [-5281.39]	421.42 [390.65]	1107 [1174]	0.381 [0.333]
		0.7	1.3	0.7	1.3	3	3	-5286.69 [-5350.32]	385.35 [321.72]	1014 [1041]	0.38 [0.309]
		0.6	1.4	0.6	1.4	2	2	-5277.63 [-5299.89]	394.41 [372.15]	868 [898]	0.455 [0.414]
		0.6	1.4	0.6	1.4	3	3	-5310.95 [-5365.15]	361.09 [306.89]	787 [776]	0.459 [0.396]
		0.5	1.5	0.5	1.5	2	2	-5292.44 [-5324.79]	379.6 [347.25]	721 [676]	0.527 [0.513]
		0.5	1.5	0.5	1.5	3	3	-5326.94 [-5389.63]	345.1 [282.41]	653 [575]	0.528 [0.491]