Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

Erling Stray Bugge and Tarje Rusten Wang

Electricity Price Forecasting Benchmark of ENTCN on Nordic Bidding Areas

Master's thesis in Industrial Economics and Technology Management
Supervisor: Stein-Erik Fleten
Co-supervisor: Odd Erik Gundersen
June 2022

NTNU
Kunnskap for en bedre verden

Erling Stray Bugge and Tarje Rusten Wang

Electricity Price Forecasting Benchmark of ENTCN on Nordic Bidding Areas

Master's thesis in Industrial Economics and Technology Management
Supervisor: Stein-Erik Fleten
Co-supervisor: Odd Erik Gundersen
June 2022

Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of
Science and Technology

Preface

This thesis is our masters project in the course TIØ4900 for our Master of Science degree in Industrial Economics and Technology Management at the Norwegian University of Science and Technology (NTNU). The thesis was written during the Spring semester of 2022. Our motivation for the thesis is to benchmark state-of-the-art models on the problem of forecasting regional electricity prices in the Nordics hourly over a seven-day horizon. Our external supervisor Odd Erik Gundersen from TrønderEnergi highlighted the lack of such a benchmark. We were further motivated by the current public attention to energy prices, which have been abnormally high in recent times. With an increasing reliance on renewable generation, more extreme weather, and high tensions with Russia, which might affect the availability of gas imports, efficient forecasting models are crucial for reducing risk and volatility in the electricity market. Furthermore, we wanted to test the hybrid *ENTCN* model developed in our project thesis (T. R. Wang et al. 2021) against state-of-the-art electricity price forecasting models. As the thesis investigates the application of state-of-the-art models to the problem of electricity price forecasting, the target audience of the thesis has a basic understanding of the electricity market and forecasting models.

Furthermore, this master thesis is built upon work done by the same authors in the project thesis in the course TIØ4550 during the Autumn semester of 2021 (T. R. Wang et al. 2021). The project thesis investigated the applicability of *temporal convolutional networks* (TCNs) in the accuracy of mid-term electricity price models. While the project thesis only looked at a single point forecast of the daily NordPool system price 14 days forward in time, promising results inspired us to further develop the model for the problem investigated in the current master thesis.

We want to thank our main supervisor, Professor at NTNU Stein-Erik Fleten, who has supported us throughout the project and master thesis with his invaluable expertise within energy markets and electricity price forecasting. Furthermore, we would like to thank Odd Erik Gundersen from TrønderEnergi, who has guided us with his expertise and practical experience in applying machine learning models for electricity price forecasting.

Lastly, the authors declare no competing financial interests or personal relations that have affected the work or conclusions presented in this thesis.

Abstract

Although electricity price forecasting has seen a myriad of proposed models in the last decades, the field still has a limited number of rigorous benchmarks. In this thesis, we perform a structured benchmark of a large number of state-of-the-art statistical and deep learning electricity price forecasting models on forecasting the hourly NordPool bidding area prices in the 12 Nordic regions over a seven-day horizon. The models implemented include; AR-type models, regression models, feed-forward neural networks, recurrent neural networks, and a naive benchmark. Furthermore, a proprietary hybrid model called *ENTCN*, which consists of an enhanced naive model and a temporal convolutional network, is tested. Although there were significant regional differences in model performance, the statistical models consistently outperformed the deep learning models, across most error metrics and bidding areas. Furthermore, while the ENTCN model outperformed comparable deep learning models, it was consistently outperformed by simpler statistical models. The ARIMA model performed best across all error metrics in Norway, while the SARIMA model was the highest performing in Denmark. However, the linear regression model performed best in both Sweden and Finland. On average, across the 12 NordPool bidding areas, the SARIMA performed best on the absolute error metrics while the ARIMA did best on the relative error metrics. Lastly, to ensure meaningful results and reproducibility, the thesis utilizes open-sourced models and well-known open-access datasets while also performing statistical tests (Diebold-Mariano) to assess the significance of differences in performance.

Sammendrag

Selv om prognoser for strømpriser har sett et mylder av foreslåtte modeller de siste tiårene, har feltet fortsatt et begrenset antall systematiske benchmarkinger. I denne oppgaven utfører vi en strukturert benchmark av et stort antall "state-of-the-art" statistiske og dyplæringsmodeller for prognose for elektrisitetspriser på de timebaserte NordPool prisområdene i de 12 nordiske regionene over en syv-dagers horisont. Modellene som er implementert inkluderer; AR-modeller, regresjonsmodeller, feed-forward nevrale nettverk, konvolusjonelle nevrale nettverk og en naiv benchmark. Videre testes en proprietær hybridmodell kalt *ENTCN*, som består av en utviklet naiv modell og et temproalt konvolusjonelt nettverk. Selv om det var betydelige regionale forskjeller i modellytelse, overgikk de statistiske modellene konsekvent dyplæringsmodellene, på tvers av de fleste feilberegninger og budområder. Videre, mens ENTCN-modellen overgikk sammenlignbare dyplæringsmodeller, ble den konsekvent slått av enklere statistiske modeller. ARIMA-modellen presterte best på tvers av alle feilmålinger i Norge, mens SARIMA-modellen var best i Danmark. Den lineære regresjonsmodellen presterte imidlertid best i både Sverige og Finland. I gjennomsnitt, på tvers av de 12 NordPool-budområdene, presterte SARIMA best på de absolutte feilberegningene, mens ARIMA gjorde det best på de relative feilberegningene. Til slutt, for å sikre meningsfulle resultater og reproduserbarhet, bruker oppgaven åpen kildekode-modeller og velkjente datasett med åpen tilgang samtidig som det utføres statistiske tester (Diebold-Mariano) for å vurdere statistisk signifisikans av forskjeller i ytelse.

Table of Contents

1 Introduction

In this thesis, state-of-the-art electricity price forecasting methods are benchmarked on a mid-term horizon on the Nordic power market. The models implemented are either statistical or deep learning and include *deep neural networks* (DNNs), *long short-term memory* (LSTM), *gated recurrent unit* (GRU), *SARIMA*, and *regression*. In addition, the thesis will investigate the performance of the hybrid enhanced naive temporal convolutional network (ENTCN) model developed by the authors, testing its applicability in electricity price forecasting. The performance testing is done on the 12 *NordPool* bidding regions in Norway, Sweden, Denmark and Finland on an hourly basis over a seven-day horizon.

The models benchmarked are either state-of-the-art statistical or deep learning methods, except for the naive forecast, which is only used as a benchmark. These are selected based on the state-of-the-art methods presented in structured benchmarks or literature reviews (Lago, Ridder et al. (2018), Weron (2014), Lago, Marcjasz et al. (2021)), with the exception of the hybrid ENTCN model developed by the authors. The statistical models implemented are ARIMA, SARIMA, linear regression and quadratic regression. Despite a large number of more sophisticated alternatives, statistical models such as linear regressions and SARIMA are still some of the most widely used electricity price forecasting approaches (Weron 2014). Their strength is that they efficiently model the seasonality prevailing in electricity markets but perform rather poorly in the presence of spikes (Weron 2014). As for the deep learning methods; deep neural networks, long short-term memory (LSTM), gated recurrent unit (GRU), and Enhanced naive temporal convolutional network (ENTCN) are implemented and tested. Deep neural networks, which in this thesis refers to a simple *multi-layer perceptron* (MLP), are often used in electricity price forecasting but more commonly as benchmarks for other more sophisticated deep learning architectures (García-Ascanio and Maté 2010). LSTM and GRU are part of a field of deep learning called recurrent neural networks (Goodfellow et al. 2016). These have traditionally been shown to be highly performing on time series tasks, including electricity price forecasting (Weron 2014). A TCN, which is a form of *convolutional neural network* (CNN) optimized for temporal data, was first proposed by Lea et al. (2016) for video-based segmentation. Two years later, Bai et al. (2018) conducted an empirical study showing that a TCN architecture was able to outperform traditional recurrent neural networks such as LSTM and GRU on several time-series forecasting tasks. Furthermore, a project thesis by the authors of this thesis (T. R. Wang et al. 2021) on the application of TCNs on daily mid-term electricity price forecasting showed that TCNs exhibited promising results.

Market deregulation in the 80s and 90s helped transform the traditionally monopolistic and state-controlled energy sector into one of free markets with competing private companies (Blazquez et al. 2018). Electricity price forecasting then also became an important field, as it was fundamental in companies' decision making in everything from purchasing strategies, production planning and investment decisions (Bunn 2004). As electricity is economically non-storable and with time-lags in changes to both generation or consumption, forecasting models are an essential market adjustment mechanism, helping stabilise prices and reducing the frequency of price spikes (Eichler et al. 2013). Several large-scale power crises highlight the societal importance of good forecasting models. The California energy crisis of 2000-2001, caused by a shortage of generation, caused large scale blackouts in the state. Electric utility companies also suffered, as they generally cannot pass excessive costs to retail consumers (Joskow 2001), with one of the state's largest energy companies collapsing. The more recent Texas energy crisis in February 2021, caused by increased demand during periods of abnormally low temperatures, resulted in prices over 9,000 USD/ MWh at certain hours (200x regular rates) (Pechman and Nethercutt 2021). As many households were left without power, an estimated 210 people died, either directly or indirectly, as a result of the crisis (Hauser and Sandoval 2021). In Norway, abnormally high prices in the southern regions in late 2021 and early 2022 have caused much public debate around electricity prices, as consumers have been especially hard hit. The frequency and severeness of such crises' might only be further magnified in the future. An increased share of variable renewable energy generation (e.g., wind, solar), which have stochastic production patterns, can increase volatility in energy markets (Brancucci Martinez-Anido et al. 2016), as experienced in Germany (Rintamäki et al. 2017). Furthermore, geopolitical tensions between Russia and the EU following the invasion of Ukraine in February 2022 might limit the supply of Russian oil and gas imports. The impact on oil and gas imports might increase electricity prices (Infrastructure 2022), while also leading to more volatile price movements as oil and gas represent an essential share of peaker capacity. Hence, there are clear motivations for researching electricity price forecasting models, as it is both of economic and societal importance. Furthermore, we were motivated to analyze a seven-day horizon, as this closely reflects the typical planning horizon for mid-term hydropower production planning (Fleten and Krogh 2008). This is a crucial aspect of the Nordic power market as 53% of the Nordic power mix is generated from hydropower (Forecasting 2019), with it representing over 90% of electricity generation in Norway (Statista 2019).

The thesis contributes to the literature and the field of electricity price forecasting in two main ways. Firstly, it provides an up to date systematic benchmarking of multiple state-of-the-art methods across multiple NordPool bidding areas in accordance with electricty price foreasting best practices (Jedrzejewski et al. 2022). Although there have been done similar studies previously, incl., Lago, Ridder et al. (2018), Lago, Marcjasz et al. (2021), and Engebretsen et al. (2021), none of these have investigated forecasting on individual NordPool bidding areas. There are also differences in the data used, forecasting horizon, and models investigated. Secondly, the thesis provides development and a state-of-the-art benchmark of a hybrid ENTCN model, which was developed in a project thesis by the authors of this thesis (T. R. Wang et al. 2021). The current thesis has developed the ENTCN model to work on hourly prices over a seven-day horizon on individual NordPool bidding areas. As there is limited research on the use of TCNs in electricity price forecasting, the development and benchmarking of such a model is instrumental in assessing the applicability of TCNs in forecasting NordPool bidding area prices.

In order to obtain generalizable and valid comparisons,

the thesis address three issues with modern benchmarking of electricity price forecasting models identified by Lago, Marcjasz et al. (2021). First, many studies comparing machine learning or statistical models, often develop highly complex models while using colloquial benchmarking models. Examples of this include, Marcjasz, Uniejewski et al. (2019), Cruz et al. (2011), Ugurlu, Oksuz et al. (2018) and W. Zhang et al. (2018), which all use simple statistical or ML models as benchmarks for assessing the performance of a developed model. This was also a weakness with T. R. Wang et al. (2021), in which the authors of this thesis had limited resources designated for developing high-performing benchmarking models. Second, many studies have very limited testing periods; examples such as Darudi et al. (2015) and Ghayekhloo et al. (2019) only test their models over one week. Weron (2014) argue that one should at least have a 1-year testing period in order to produce generalizable results, incorporating seasonal and holiday effects. Thirdly, many studies lack the details needed to ensure reproducibility, preventing others from being able to validate. Problems include not specifying the in-sample and out-of-sample periods (Talari et al. 2017), not specifying the inputs used in the models (Khan et al. (2017), Afrasiabi et al. (2019)), or not specifying the dataset used (Bento et al. 2018). Furthermore, we are in this thesis basing our methodology on the best practices presented in Lago, Marcjasz et al. (2021) and Croonenbroeck and Stadtmann (2019). What separates this paper from the state-of-the-art benchmark conducted by Lago, Marcjasz et al. (2021) is that this thesis will investigate the bidding regions in the NordPool market instead of system prices across different power markets. Furthermore, it investigates a mid-term horizon instead of a short-term horizon while also implementing a broader number of statistical and deep learning models.

Forecasting electricity prices using statistical and data-driven deep learning models is notoriously challenging. Firstly, due to the issue of non-storability and supply/ demand elasticities, prices are highly volatile with numerous price spikes (Zedda and Masala 2019). These seemingly random spikes can be challenging to forecast without suitable data sources. Additionally, electricity prices have exhibited many structural breaks across regions over the 21st century, often coinciding with critical events (e.g., new capacities, policy changes, supply shocks, new power lines) (C. Lee and J. Lee 2009). Therefore, models trained on older data than what they are tested on might be less efficient, as structural breaks might have caused changes in the price dynamics. An example of this is the abnormally high NordPool daily system prices last year. On 29 November 2021, the daily system price was 247 €/MWh, three times the maximum experienced over the five-year in-sample data from 2014 to 2019. One significant difference between the two periods is the operation of new 1,400 megawatts under-water cable between Norway and the UK, which at max capacity can power 1.4 million homes according to National Grid (Chen and Y.-y. Wang 2021). Hence, training on the in-sample data might not be generalised to good performance on the out-of-sample data. Furthermore, good performance on the out-of-sample data might not be generalised to more recent periods. As deep learning models are dependent on large amounts of in-sample data for training, a mid-term forecasting model is therefore still required to use data from many years back in time, if not to utilize training examples from other power markets.

This master thesis is structured as follows: Section 2, Background, provides relevant background on electricity price forecasting, the implemented methods, and modern electricity markets. Section 3, Literature Review, contextualizes the work within relevant literature and highlights both papers proposing state-of-the-art methods and structured reviews of models. Section 4, Data, analyses the in-sample data, looking at the regional electricity prices and the exogenous variables used. Section 5, Methods, explains the implementation of the models while also explaining the error metrics, statistical tests, and experimental design used. Section 6, Results, presents the performance of the implemented models on the out-of-sample data, compares models using the Diebold-Mariano test, and analyses high performing models on specific test example periods. Section 7, Discussion and Conclusion, discusses and draws a conclusion based on the results from section 6. Finally, section 8, Further Work, highlights potential further work within state-of-the-art benchmarking of hourly prices across the Nordic bidding areas.

2 Background

The current section provides background on electricity price forecasting and modern electricity markets. The models implemented in the benchmark are explained and contextualised within the electricity price forecasting field. Furthermore, the section discusses modern electricity markets' key characteristics and dynamics, focusing on the NordPool market, which is essential to better analyse the model results.

2.1 Electricity Price Forecasting

Following the liberalization of electricity markets in the 1980s and 90s (Blazquez et al. 2018) private companies started being exposed to electricity risk. The risk could include over or under contracting and then selling or buying electricity in real-time, which could potentially lead to increased credit risk, or at worst, bankruptcy (Y. Zhang et al. 2022). Hence, electricity price forecasting quickly became a crucial input in electricity companies' decision-making process (Eydeland and Wolyniec n.d.). As a result, during the last two decades, there has been a steady increase in the number of proposed electricity price models (Weron 2014), emphasising their function in modern electricity markets. However, while there is currently a myriad of proposed models, it is essential to note that there are still significant differences across models, including factors such as; modelling approach, time horizon, output type, and target electricity market/ area.

A review of the state-of-the-art forecasting models conducted by Weron (2014) identified five main modelling approaches within electricity price forecasting; statistical, computational intelligence, multi-agent, fundamental and reduced form. In addition, there also exists a category of hybrid models, which combine elements of different types of models. A structured literature review conducted by Engebretsen et al. (2020) found that statistical and computational intelligence accounted for 80% of electricity forecasting models, these are also the two types of models utilized. 90% of the models in discussed in Weron (2014) focus on the day-ahead market. The taxonomy of electricity price forecasting models developed by Weron (2014) can be seen in Figure 1.

Statistical models forecast the current price as a combination of previous prices and exogenous factors (e.g., consumption, production, weather variables). The two most important models are additive (sum of factors) and multiplicative (product of factors). An appealing feature of statistical models is that there is a physical interpretation of the components, allowing for a more straightforward analysis of model behaviour (Weron 2014). Their main drawback is their limited ability to model the highly nonlinear behaviour of electricity prices and related exogenous variables. Despite this drawback, their practical performance is comparable to that of nonlinear models. Computational intelligence has been notoriously hard to define, Ventosa et al. (2007) describing it as "a new buzzword that means different things to different humans". Weron (2014) defines computational intelligence as using elements of learning, evolution and fuzziness to make forecasts, splitting the field into four main types; feed-forward neural nets, fuzzy neural nets, recurrent neural nets and support vector machines. Computational intelligence models can learn features from data, mitigating the need for manual feature engineering (Goodfellow et al. 2016). Another strength is its ability to model nonlinear relationships (Weron 2014). In recent years, computational intelligent models have established their place in the field of electricity price forecasting, with 60% of proposed models using some form of computational intelligence (Engebretsen et al. 2020), many of which exhibiting excellent model performance (Weron 2014). Throughout this thesis, computational intelligence models are interchangeably referred to as deep learning models. In multi-agent methods, one is trying to simulate the operations of multiple heterogeneous agents interacting in the market. The market price is calculated as the intersection between supply and demand. Examples of multi-agent models include; agent-based, Nash-Cournot, and equilibrium functions. Fundamental models utilize structural data (e.g., capacity changes, flow capacities, import shocks) at market breaks to model the electricity price over a longer horizon. Power companies often prefer fundamental models when making long-term decisions (e.g., capacity investments) as they infer a causal relationship between independent and dependent variables (e.g., the impact of a specific market shock), making them easier to understand and validate. Reduced form models attempt to characterise the statistical properties of electricity prices and include markow switching and jump-diffusions. Lastly, hybrid models combine elements of different models, which might use results of one type of model as input to another, or utilize specific models only in the preprocessing phase. In recent years, hybrid models have become increasingly commonplace due to their flexibility and ability to combine the "best of several worlds" (Engebretsen et al. 2020).

When talking about electricity price forecasting, it is the convention to split between short-, medium-, and long-term electricity price forecasting, without there being consensus between what thresholds separate the three (Weron 2014). The span in the horizon is expansive, with everything from forecasts on a 5-minute basis (Yang and Schell 2020) to fundamental forecasting models forecasting the price over a three year period (Ziel and Steinert 2018). Weron (2014) defined short term to be everything shorter than 14 days and long-term as longer than three months, medium-term being everything in-between. Although mid- and long-term models have many unique use cases, such as in production planning, investment decisions, and contract negotiations, most of the proposed models in the literature are defined as short-term (Weron 2014).

The following subsections cover background on the statistical and computational intelligence (deep learning) methods implemented in this thesis. These include; naive forecast, deep neural networks, the recurrent neural networks LSTM and GRU, temporal convolutional networks, and SARIMA.

2.1.1 Naive Forecasts

The naive forecasting method is a univariate method in which one predicts the future to be the same as the present or past. Although it is not a very sophisticated model, it is commonly used as a benchmarking model to better assess the performance of other models.

The naive forecasting model can be expressed as in Equa-

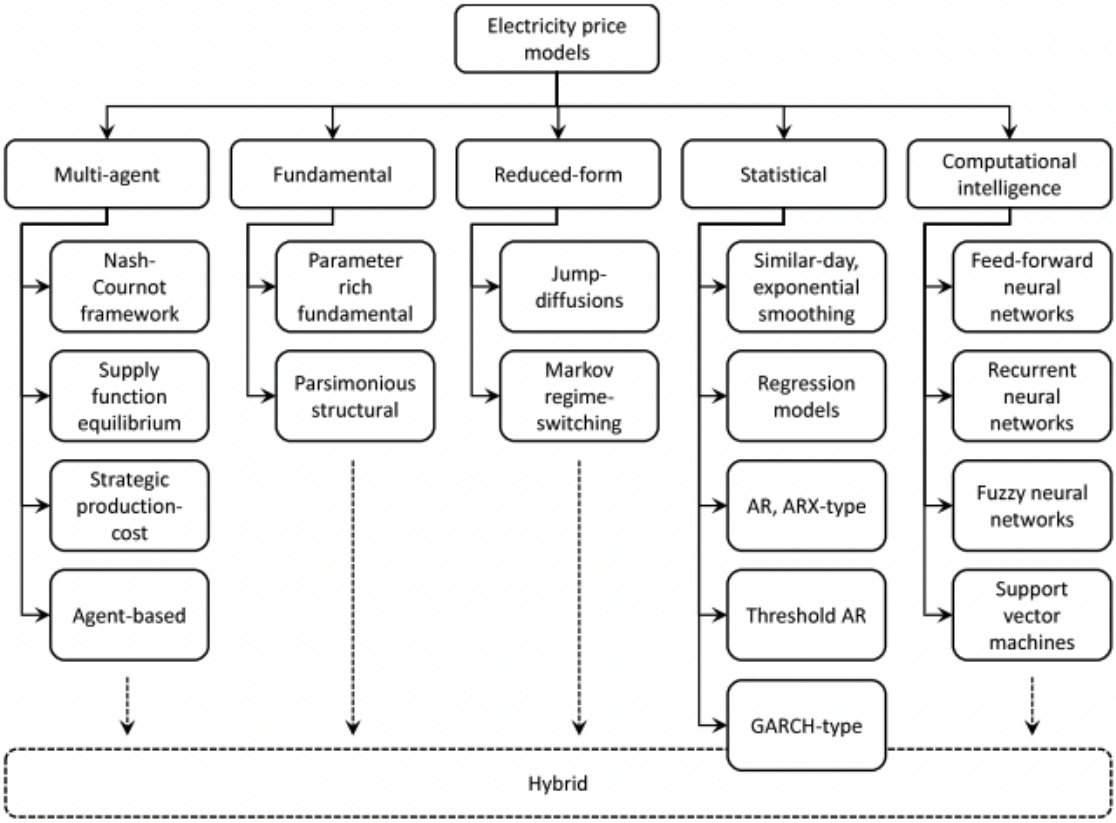


Figure 1: Taxonomy of electricity price modeling approaches (Weron 2014)

tion 1, in which $\hat{y}_i$ and y are the forecasted and actual value at time i, with k being a time interval at which one looks back.

$$\hat{y}_i = y_{i-k} \tag{1}$$

When developing the model, one needs to decide on what time to base the forecast. For example, when forecasting the hourly electricity price at a specific time, one has to decide what to base it on (e.g., the previous hour, the same hour the previous day, or the same hour and day the previous week). What is most rational is often a product of the price dynamics (e.g., degree of time-of-day and weekday seasonalities). Furthermore, a naive forecast is also straightforward to adjust for calendar effects, such as holiday, time of day, and weekday.

The model's strength is that it is extremely simple while providing a basic benchmark to assess model performance. However, the naive forecast is most often not suited to make accurate forecasts as it does not utilize any information from exogenous variables. For example, electricity price forecasting might be less well suited given the effect of time-of-day, day-of-week, and holiday on prices. However, a benchmarking of electricity price forecasting models on a 14-day period of NordPool system prices conducted by Engebretsen et al. (2021) found that the naive model was able to outperform most other models implemented.

2.1.2 Deep Neural Network (DNN)

Deep neural networks (DNNs), a form of machine learning, uses a network of nodes and mathematical operations to make forecasts or classifications. McCulloch and Pitts (1943) published the first systematic study on artificial neural networks, with a computational model of the neural activity of the human nervous system. A neural network is built to simulate the activity of the human brain, with pattern recognition as data is passed through multiple layers of neural connections (Goodfellow et al. 2016). However, their breakthrough came in the 1980s, when better techniques and more processing power allowed for the development of practical neural networks (Ismail 1989). Since then, their popularity has boomed, with a wide array of use cases, such as facial recognition, stock market forecasting, translation and electricity price forecasting (Andina et al. 2007).

The most basic element of a neural network is a "node", also interchangeably called a "neuron", as pictured schematically in Figure 2.

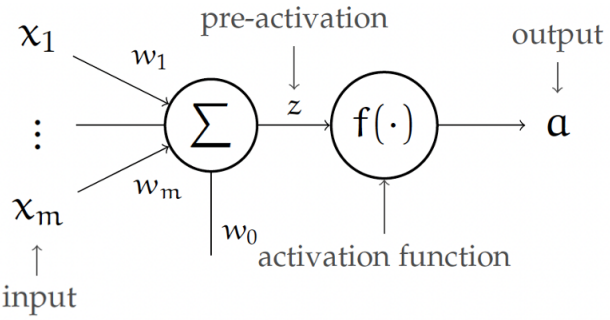


Figure 2: Schematic visualization of a neural net node ("neuron"), consisting of an input and converted to a output value through a activation function (MIT 2020a)

It is a non-linear function, transforming an input vector $x \in \mathbb{R}^m$ to a output value $a \in \mathbb{R}$. The node also has a parameter vector of weights $(w_1, ..., w_m) \in \mathbb{R}^m$, in addition to an offset value $w_0 \in \mathbb{R}$. An actication function $f : \mathbb{R} \to \mathbb{R}$ is used in order for the node to be non-linear. The activation function can be everything from the identity function ($f(x) = x$) to *ReLU* ($f(x) = max(0, x)$), *sigmoid* ($f(x) = \frac{1}{1+e^{-x}}$) or any other. The function represented by the node is as written in Equation 2.

$$a = f(z) = f((\sum_{j=1}^{m} x_j w_j) + w_0) = f(w^T x + w_0) \qquad (2)$$

In general, the network takes an input $x \in \mathbb{R}^m$ and generates an output $x \in \mathbb{R}^n$ through connecting multiple such nodes in an acyclical directed graph (MIT 2020a). The input of one node is the output of a previous node. Such a network is often referred to as a feed-forward network (Goodfellow et al. 2016). In a feed-forward network, the input of a node can never depend on the output of that neuron, with data flowing in one direction and the function of the network being a composite of the functions of the individual neurons (MIT 2020a). A simple example of such a feed-forward network consisting of two hidden layers with four nodes each can be seen in Figure 3.

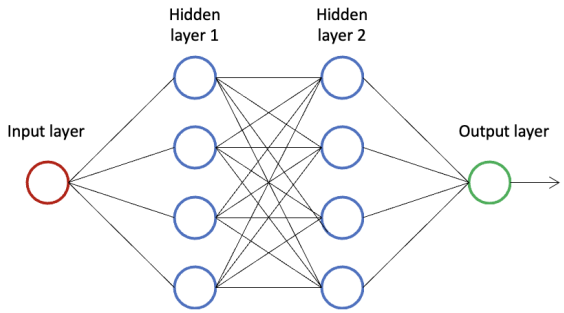


Figure 3: Simple model of a feed-forward deep neural net, in which the input is transformed through two hidden layers and an output layer

When training such a network, a loss is computed based on the generated values from the network y_i and the actual value x_i, e.g., mean absolute error ($\frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$). Training based on a set of known actual values is supervised machine learning. The weights in the network are then tweaked to reduce the network's loss, which can be done using several optimizers such as *stochastic gradient descent* (SGD), *adadelta*, or *adam*. The idea is that the loss will gradually converge towards a local minimum given enough training.

Deep neural networks have several strengths and weaknesses, which affect their suitability for electricity price forecasting:

+ They have routinely shown their ability to outperform alternative models across various tasks and have become a go-to method for many use cases, including electricity price forecasting (Weron 2014) (Lago, Marcjasz et al. 2021).

+ Neural networks do not need any feature engineering, as they can detect patterns and relationships in the data, making them highly generalised (Goodfellow et al. 2016). This is highly relevant when doing electricity price forecasting, where there are many domain-specific variables and market dynamics.

+ Neural networks can model nonlinear relationships, which is highly relevant as this is the case in electricity markets, where there exist many nonlinear relationships between variables (Weron 2014)

- Neural networks often require a large amount of data in order to train (Goodfellow et al. 2016). This can often be a challenge in medium- and long-term electricity price forecasting. Gathering enough real-life training examples might require one to look at different markets or look very far back in time, making the data less relevant.

- Both training and testing neural networks is a highly "black box" process, as there is it practically impossible to analyse or draw any practical relations from the weights of the network (Goodfellow et al. 2016). This is a challenge in electricity price forecasting, as many market participants often prefer models that infer a causal relationship between dependent and independent variables.

- As with other forms of supervised learning, there is a high risk of overfitting, as the model trains to best fit the training data. Furthermore, there is also required that the future one is trying to forecast exhibits the same patterns as the data on which the model is trained. Finally, as with electricity markets, fundamental market shifts (e.g., Russian trade embargo, green shift to more renewable, infrastructure changes) might make models trained on old data less efficient.

- DNNs are a-theoretical as they use little theoretical information about the relationships between variables to guide the specifications of the model (Brooks 2019). Hence they are less tailored to electricity price forecasting, as they are not exploiting known theoretical relationships between variables. This weakness holds for all deep-learning models implemented in this thesis.

2.1.3 *Recurrent Neural Networks: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)*

This section will describe two types of *recurrent neural networks* (RNN) known as *long short-term memory* (LSTM) and *gated recurrent unit* (GRU). We will contextualize by firstly describing a "vanilla" RNN architecture before we present the architectures of the LSTM and GRU and how these two versions of RNN differ from the standard recurrent neural network design. Lastly, we will summarize and present the strength and weaknesses of the long short-term memory and gated recurrent unit.

Recurrent neural networks are neural networks which are designed to process sequential data (Goodfellow et al. 2016). The recurrent neural network differs from the deep neural network, previously described, by having connections between the neurons in each layer. Furthermore, in this section, a neuron in a recurrent neural network will be described as a cell due to more complex operations on the inputs in the neuron/cell for a recurrent neural network than in a deep neural network. Each cell in a recurrent neural network layer is associated with one step of the input sequence. Consequently, the number of cells will be equal to the length of the input in the sequential (and not feature) dimension of the input. The standard recurrent neural network cell receives two inputs and has one output. Each cell process one step of the input variable (x_{t-k}) and the output from the past cell ($c_{t-(k-1)}$). The standard process of an RNN is to perform a function chosen by the developer on the concatenation of the input (x_t) and past cell state (c_{t-k}) to create the new cell state (c_{t-k}). The new cell state is passed to the next cell in the same layer. Therefore, the cell state (c_{t-k}) can be viewed as a state vector which is updated by all the cells in the RNN-layer before it is outputted by the cell processing the last input. The cell processing the input on the most recent element in the input (x_t) will pass its output (c_t) to a new model.

To further extend the idea of layers in recurrent neural networks, there is an opportunity to stack numerous layers of RNN-cells on top of each other for more complex data processing. In a stacked layer, the computational processes are the same. However, a stacked layer will process input from all the cells in the layer below and not only the last cell. Therefore, the number of cells will be the same across different layers in a stacked recurrent neural network. In addition to being able to be stacked, all recurrent neural networks also share a feature known as parameter sharing across a single layer of cells. This means that the parameters of the cells are identical across a layer of RNN-cells processing an input (Goodfellow et al. 2016). By parameters, we mean the weights and biases used in each cell's transformation function. Parameter sharing enables the recurrent neural networks to process variable-length inputs and have fewer parameters to update in training since it only requires the unique set describing one cell. However, it also requires more optimization when training and the networks cannot weigh individual sequence steps differently. Another downside of "vanilla" recurrent neural networks is that the networks struggle with modelling long term dependencies in the input. This is known as the vanishing gradient problem (Goodfellow et al. 2016). As a solutions to the vanishing gradient problem, Hochreiter and Schmidhuber (1997) proposed the long short-term memory cell architecture and Cho et al. (2014) proposed the gated recurrent unit architecture.

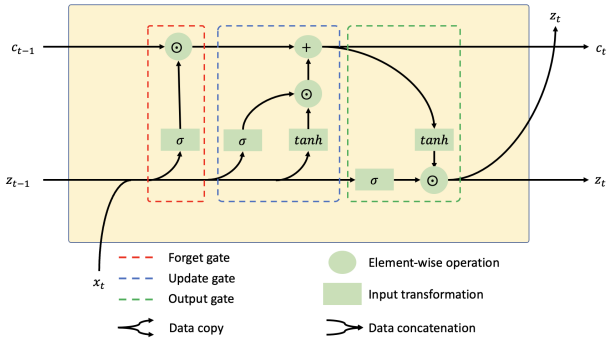


Figure 4: The logic architecture of a long short-term memory cell, which consists of three gates encapsulated by the dashed boxes. The data forward pass is in the direction of the arrows and a transformation or operation cannot be calculated before all input is received.

The long short-term memory model is displayed in Figure 4, and was proposed by Hochreiter and Schmidhuber (1997). The long short-term memory model differs from the standard recurrent neural network in cell architecture. Therefore, it will still have one cell per step in the input segment and use parameter sharing across the cells in a layer. The long short-term memory model in this thesis uses the model architecture proposed by Hochreiter and Schmidhuber (1997), which is a version of a recurrent neural network. They highlighted the model's advantages for long term time dependencies compared to other recurrent networks for time series forecasting. The long short-term memory cell contains three main gates, all marked with a separate box in Figure 4. These are the forget gate, the input gate, and the output gate. z_t indicates what is known as a hidden state, c_t is the cell state, and x_t denotes the input. The orange operator boxes represent transformations. The σ is the sigmoid function on the input, described in Equation 3. $tanh$ is the hyperbolic tangent functions, as described in Equation 4. W are the weights, and b is the bias. x is simply the input to the function. The orange operator circles represent element-wise operations. The $\odot$ represents the *Hadamard product*, also known as pairwise multiplication, and the $+$ represents the addition of two inputs.

$$\sigma(x) = \frac{1}{1 + e^{-(Wx+b)}} \tag{3}$$

$$\tanh x = \frac{e^{Wx+b} - e^{-(Wx+b)}}{e^{Wx+b} + e^{-(Wx+b)}} \tag{4}$$

The long short-term memory cell logic can be divided into three groups, as represented by the gates in Figure 4. Firstly, the forget gate, sigmoid transforms Equation 3 the input (x_t) concatenated with the hidden state (z_t) of the previous cell. Then, the Hadamard product is computed to identify which parts of the previous cell state to remember and which to forget. Secondly, there is is input gate, which has the task to update the cell-state from c_{t-1} to c_t. The sigmoid transformation in the input layer will assign a weight to the input in the range $[0, 1]$. Furthermore, the tanh transformation will scale all input between $[-1, 1]$. By computing the Hadamard product of these two outputs, we will have both a magnitude and direction for updating

the cell state. Lastly, there is the output gate. The hidden state is updated in a manner comparable to that of the cell state. The cell-state (c_t) is scaled through a hyperbolic tangent transformation Equation 4 and the importance of each entry in the state vector is determined by the sigmoid transformation of the concatenated vector of the input (x_t) and previous hidden state (z_{t-1}) (Goodfellow et al. 2016).

The gated recurrent unit was proposed by Cho et al. (2014) and is occasionally tested alongside a long short-term memory architecture when assessing deep learning models for time series forecasting, such as Lago, Ridder et al. (2018). As with the long short-term memory cell, it only differs from the standard recurrent neural network in cell architecture. Therefore, it also has a separate cell for each step in the input sequence and uses parameter sharing. However, the data processing can still be viewed as a cell-wise updating of a state vector throughout the layer. The architecture of each cell is shown in Figure 5.

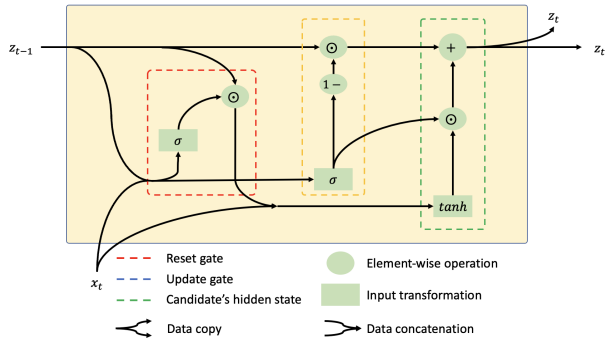


Figure 5: The logic architecture of a gated recurrent unit cell, which consists of three gates encapsulated by the dashed boxes. The data forward pass is in the direction of the arrows, and a transformation or operation cannot be calculated before all input is received.

The gated recurrent unit can be divided into three parts, which are visible in Figure 5 as the different gates. The data in the gated recurrent unit cell is processed in the following manner: Firstly, the reset gate, secondly, the update gate and, lastly, the candidate's hidden state. The reset gate first concatenates the input from the hidden state (z_{t-1}) and the input vector (x_t), before it is sigmoid transformed Equation 3. This is element-wise multiplied with the previously hidden state (z_{t-1}). The sigmoid function transforms the input into a value between 0 and 1. The concatenated x_t and z_{t-1} are in the update gate also transformed through the sigmoid function. By subtracting this output from 1, we find that what should not be updated is still 1, and what should be updated will be closer to 0. This will also be the result after the Hadamard product is calculated at the end of the update gate. The last gate is the gate for updating the candidate's hidden state. Here, a hyperbolic tangent transformation of z_{t-1} and x_t is conducted. By giving them values in the range $[-1, 1]$, the magnitude and direction of each variable is considered. The subsequent Hadamard product is controlled if this is a value that should be updated. Finally, the relevant changes are made to the hidden state z_{t-1}, transforming it to z_t.

Compared to other neural networks, and more specifically recurrent neural networks, there are both strengths and weaknesses to the long short-term memory and gated recurrent unit architectures (Goodfellow et al. 2016).

+ Ability to capture long term dependencies. This is beneficial when processing time series, which exhibit properties of auto-correlation. This is a property electricity prices are known to have.

+ Studies have shown that long short-term memory models perform well on numerous tasks.

+ Parameter sharing lowers the amount of computation required for training. This enables us to train more complex models for longer, enabling the models to capture more relationships in the data.

- Parameter sharing requires more optimization, which is computationally heavy. This demands more resources from us.

- Equal weights across all input means that the model cannot explicitly assume that some past observations are more important than others. This is a limitation because electricity prices often possess calendar effects.

- As with all deep learning algorithms, large amounts of data and adequate hardware are required. We, therefore, cannot blindly test models but must find suitable starting parameters and conduct thorough searches around these parameters.

- The result of a deep learning model depends heavily on the initialization of weights. Therefore, the models must be trained and tested extensively for reliable results.

2.1.4 Temporal Convolutional Network (TCN)

A *temporal convolutional network* (TCN) is a *convolutional neural network* (CNN) optimized for time-series data. The method was first proposed by Lea et al. (2016) for video-based segmentation. Furthermore, in 2018 an empirical study conducted by Bai et al. (2018) showed that the TCN architecture was able to outperform standard neural network sequencing models (incl., LSTM) on time-series forecasting tasks such as sequential MNIST[1] and word-level language modelling. Although still little used, there are currently some proposed papers researching the use of TCNs, such as work by Yan et al. (2020) which used TCN for weather forecasting. In this thesis, the TCN architecture is based proposed architecture by (Bai et al. 2018).

As it is a form of convolutional neural network, it is a feed-forward neural net which uses convolutions in at least one of its layers (Goodfellow et al. 2016), often combined with other layers such as max pooling, flatten and fully connected. A *convolution* is a form of linear mathematical operation, written as seen in Equation 5.

$$s(t) = (x * w)(t) \tag{5}$$

Here x is the two-dimensional independent variable, w the *kernel*, s the dependent variable, and t is time (Goodfellow et al. 2016). An example of how a 2-dimensional convolution is applied on a 3x4 input matrix using a 2x2 kernel is shown in Figure 6. Here, a kernel is slid across the input matrix to produce the output matrix, in which each entry

[1] Large dataset of handwritten digits

is a linear combination of the value in the input matrix dependent on the kernel weights. The use of convolutions is very commonplace within image classification (Bhandare et al. 2016) and natural language processing (Tong et al. 2020). This is what is referred to as a convolutional neural network. However, temporal convolutional networks are defined by several additional features, making them better suited for sequential data.

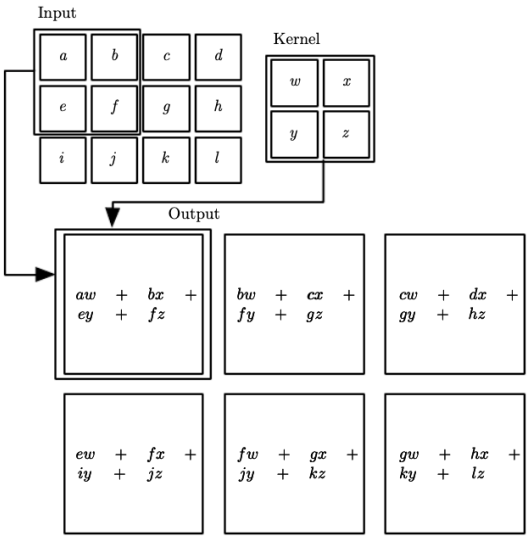


Figure 6: Visualization of a 2-dimensional convolution applied on a 3x4 input matrix using a 2x2 kernel. A kernel is slid across the input matrix to produce the output matrix, in which each entry is a linear combination of the values in the input matrix weighted based on the kernel weights (Goodfellow et al. 2016)

The TCN method described in this thesis is based on the architecture proposed by Bai et al. (2018), which is based upon two key principles:

1. The TCN produces an output of equal length as the input. Equal length is ensured through the use of *zero-padding*, which is adding 0-values at the end of sequences such that each layer is of equal length (Long et al. 2015).
2. There is a causal relationship between independent and dependent variables, with no data leakage from the future to the past, ensuring validity as a time-series forecasting model. This is done through causal convolutions in which the output at time t is only convolved with values from time t or earlier.

Furthermore, the proposed architecture by Bai et al. (2018) includes a number of integrated techniques from modern CNN architectures, which solve the problem of allowing for a long effective history (memory) and the use of many layers. Two of these techniques are *dilated convolutions* and *residual blocks*, which are visualized in Figure 7.

Dialated convolutions (as shown in Figure 7 (a)) is a form of causal convolution using only values at certain intervals of the input layer. For a 1-dimensional sequence $x \in \mathbb{R}$ and a filter $f : \{0, 1, ..., k-1\} \rightarrow \mathbb{R}$ this can be formulated as in Equation 6.

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i} \tag{6}$$

In Equation 6 s is the index in the sequence, d is the dilation factor, k is the filter size. Using dilated convolutions allows for the use of exponentially larger receptive fields (Oord et al. 2016), one can then easily increase the memory of the network.

Residual blocks (as shown in Figure 7 (b,c)) consist of a number of operations (e.g., causal convolution, dropout, ReLU) (He et al. 2015). This allows for layers to be trained to learn modifications to the identity mapping (Bai et al. 2018), which in many instances have shown to improve the performance of complex neural networks. The residual block also includes an optional convolutional layer which can be used when the input and output are of different dimensionality.

Compared to deep neural networks and traditional sequencing models such as LSTM, TCNs have several strengths making them interesting for electricity price forecasting. Unfortunately, TCNs also possess some weaknesses.

+ Unlike recurrent neural networks such as LSTM, TCNs use the same filter in each layer. There is, therefore, no need for previous forecasts in each time step, allowing for parallelism with in- and out-of-sample sequences processes as a whole (vs sequentially).
+ TCNs have different paths for backpropagation than the temporal direction of the sequence. This difference in backpropagation paths ensures stable gradients, exploding/ vanishing gradients being a common pitfall for neural networks and recurrent neural networks.
- There is a significant need for data storage when running the model out-of-sample as the network needs to be fed with input sequences of equal length to that of the history.
- The optimal dilatation and kernel size might need to be adjusted when changing the domain or with changing dynamics between variables. Different regional electricity price bidding areas might require manual tweaking of parameters to ensure optimal performance.

2.1.5 Regression

Regression is a supervised learning machine learning method that tries to fit a function based on a set of training examples. In addition to being used for making forecasts, it can also be used to calculate the causal relationship between dependent and independent variables. It is a relatively simple model with a wide range of practical use cases today.

A regression model can be linear, as shown in Equation 7, or of a higher degree, second degree shown in Equation 8. Here a is the intercept, b_i and c_i the weights, and u the residual. A set of training data is used when fitting

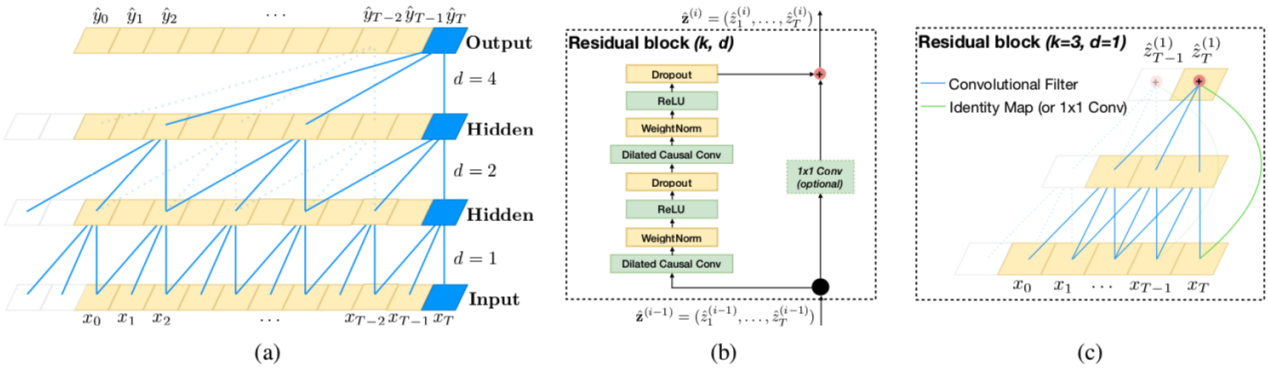


Figure 7: Illustaration of techniques used in the TCN architecture. The left panel (a) shows a dilated convolution, with different dilation factors ($d = 4, 2, 1$), which decides at what intervales values from the input layer are read. The middle panel (b) shows a residual block, transforming an input through several operations. The right panel (c) shows the use of a residual block in a TCN network. Figure from Bai et al. (2018)

the model, and the model's weights are adjusted to minimize a set loss function. In the absence of additional information about the problem, squared error ($Loss = (guess - actual)^2$) is typically used (MIT 2020b).

$$y_x = a + b_1 x_1 + b_2 x_2 + ... + b_k x_k + u \tag{7}$$

$$y_x = a + b_1 x_1 + c_1 x_1^2 + b_2 x_2 + c_2 x_2^2 + ... + b_k x_k + c_k x_k^2 + u \tag{8}$$

The problem of finding a linear hypothesis that minimizes the mean squared error is referred to as the *ordinary least squares* (OLS) problem. A closed-form method for solving for ordinary least squares was first proposed by Legendre (1805), in which one directly computes the weights (θ), minimizing the objective function given in Equation 9. A drawback of this is that finding the analytical solution takes $O(d^3)$ time, in which d is the number of features. Hence, when dealing with high-dimensional data, gradient descent is often used instead (MIT 2020b). Furthermore, since the objective function for ordinary least squares is convex (meaning they only have one minimum), using gradient descent with a small enough step size is guaranteed to find the global optimum.

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} (\theta^T x^{(i)} - y^{(i)})^2 \tag{9}$$

Regression models have a number of strengths and weaknesses in the context of electricity price forecasting compared to alternative models.

+ Regression models do not need a large amount of training data to fit a model, which is useful in mid- and long-term electricity price forecasting where one might have limited relevant training data.
+ Regression models are easy to implement and analyse. They also help provide causal relationships between the variables. However, one must be aware that correlation does not mean causation and that any explainable variable should be included in the model to draw inferences.
- As with all supervised learning, there is a risk of overfitting, with the model being made to fit the training data best. A response is to use a ridge or lasso regression, which penalises high parameters, reducing the risk of overfitting.
- Linear regression cannot model nonlinear relationships, often present in electricity markets. However, one might also implement regression models of higher orders (e.g., quadratic regression), which can model some nonlinear relationships.

2.1.6 Seasonal, AR, MA and SARIMA Models

The *SARIMA* model is an a-theoretical statistical model which uses past observations to forecast future values of a sequence and will be described in this section. The SARIMA model we have chosen is from Durbin (2012). The model consists of an *autoregressive* component (AR), a *moving average* component (AR), an integrated component (I) and a seasonal component (S). The autoregressive component (AR) of the SARIMA-model looks at lagged values and the average in the model. It uses these for forecasting future values of the sequence and the average value for the sequence, μ. t_{t-i} represent the i^th lagged value, which is given a weight of ϕ_i. The model will have a total of p lags, which are added to create the prediction $\hat{y}$. ζ represents the error. This is showed in Equation 10.

$$y_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i} + \zeta_t \qquad \zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2) \tag{10}$$

Moving average aims to model the error deviations (ζ) from the average (μ) to make a more accurate forecast for future values. This process is described in Equation 11. For the moving average process, the weights for the lagged errors are denoted θ_{t-j}, with j being the j^{th} lag, going up to a total of q lags. The j^{th} lagged errors themselves are ζ_{t-j}.

$$y_t = \mu + \sum_{j=1}^{q} \theta_j \zeta_{t-j} + \zeta_t \qquad \zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2) \tag{11}$$

The integrated component in a SARIMA-model aims to make a sequence with a trend stationary (Durbin 2012).

This is done by looking at the difference between two elements in the sequence. The difference factor of a SARIMA model is often denoted d. We have described first difference, second and the d^{th} difference, in equation Equation 12, Equation 13 and Equation 14, respectively. This is done to give the reader an understanding of the recursive working of the difference operator. Hence, by looking at the d^{th} difference in Equation 14, one will look at the difference for the $(d-1)^{th}$ difference for the values. However, it is important to note that often only the first difference is required, as there are only a few sequences where a higher order is necessary (Brooks 2019). To synthesize, by utilising an integrated component different from 0, one will not aim to model the actual values (y_t) but rather the differences from past values.

$$\Delta y_t = y_t - y_{t-1} \tag{12}$$

$$\Delta^2 y_t = \Delta(\Delta y_t) = \Delta y_t - \Delta y_{t-1} \tag{13}$$

$$\Delta^d y_t = \Delta(\Delta^{d-1} y_t) \tag{14}$$

The last component of the SARIMA model is the seasonal component. This component consists of multiple sub-components: A seasonal autoregressive component, a seasonal moving average component, and a seasonal integrated component. Additionally, it is defined by a cycle frequency (m), which describes the number of sequence steps in a single cycle. The seasonal autoregressive component (P) works similarly to the regular component with p lags. However, where the regular autoregressive will have lagged values between 1 and p, the seasonal component will have lagged values at each cycle frequency. This is described by Equation 15, which results in $m, 2m, ..., Pm$ as lagged values. Similarly, we can describe the seasonal moving average process (Q). This models a moving average process, but only at each cycle frequency, $m, 2m, ..., Qm$. This property is also described by Equation 16. The final seasonal component is the integrated seasonal differences (D), which models the difference similar to the non-seasonal difference operator. However, the seasonal difference will have one cycle and not one lag between the two variables. The equations for the integrated seasonal component is summarized in Equation 17.

$$y_t = \mu + \sum_{k=1}^{P} \Phi_k y_{t-mk} + \zeta_t \qquad \zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2) \tag{15}$$

$$y_t = \mu + \sum_{l=1}^{Q} \Theta_l \zeta_{t-ml} + \zeta_t \qquad \zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2) \tag{16}$$

$$\Delta_m^D y_t = \Delta_m(\Delta_m^{D-1} y_t) \tag{17}$$

Combining the autoregressive, moving average, integrated and seasonal components described so far in this section makes up the SARIMA model in Equation 18. A SARIMA model is often described as $SARIMA(p, d, q)(P, D, Q, m)$, where the letters in the brackets describe the hyperparameters. The meaning of each of the letters is described above. Notably, Equation 18 describes a SARIMA(p, 0, q)(P, 0, Q, m)-model. The value of the different lag coefficients, $\phi_{t-i}, \Phi_{t-mk}, \theta_{t-j}$ and Θ_{t-lm} are estimated through a *maximum likelihood process*.

$$y_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \zeta_{t-j} + \sum_{k=1}^{P} \Phi_k y_{t-mk} + \sum_{l=1}^{Q} \Theta_l \zeta_{t-ml} + \zeta_t \qquad \zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2) \tag{18}$$

An *information criteria* is often used for evaluating the hyperparameters when creating a SARIMA-model. Furthermore, it is desirable with a model capable of modelling sufficiently complex relationships without overfitting (Brooks 2019). Overfitting is undesirable since it leads to more significant out-of-sample errors. Information criteria are evaluation metrics for models which aim to address this issue. The metrics reward high accuracy for the model but penalize the increasing number of parameters. There exist numerous different information criteria formulas, but one of the most common in electricity price forecasting literature is *Akaike's information criteria* (AIC) (Akaike 1974; Croonenbroeck and Stadtmann 2019). The Akaike's Information Criteria is described in Equation 19. The $ln(\hat{\sigma}^2)$ is the natural logarithm of the standard error of the model, k is the number of parameters, and T is the number of observations. Consequently, a more accurate model will reduce the standard error $\hat{\sigma}^2$ and, therefore, the AIC-score. A more parameter rich model will increase k, which increases the AIC. More observations will lower the AIC score since it is inversely proportional to T. A model is therefore likely to be preferable to another model by having a lower information criteria score, and we seek to find the hyperparameters which minimize the AIC-score (Brooks 2019).

$$AIC = ln(\hat{\sigma}^2) + \frac{2k}{T} \tag{19}$$

The SARIMA model has both strengths and weaknesses, the most relevant ones listed below (Brooks 2019):

+ SARIMA models are simple to understand and analyse. The causal relationship between the dependent and independent variables is easily observable. Therefore, plots of electricity prices can infer approximate starting values for hyperparameters prior to a search around these values.

+ A few hyperparameters have to be chosen when creating a SARIMA model. As a result, there are few dimensions to a grid search, making it less computationally expensive. A grid search is desirable since it evaluates all proposed options (Goodfellow et al. 2016).

+ Often useful for forecasts out-of-sample (Brooks 2019).

- These models as a-theoretical. Meaning they do not utilize any known relationships between the variables. This can result in spurious relationships if not careful. Furthermore, one cannot implement domain knowledge about the sequence's relationship to other variables. This is unfortunate as, for example, high future wind production (from weather forecasts) will lower the price, but it is a feature a SARIMA model cannot implement.

- It can be challenging to choose the optimal hyperparameters for the model while avoiding overfitting. The use of information criteria is an option to reduce the risk of overfitting.

- Many parameters to estimate through maximum likelihood. Hence, demanding a lot of computational resources, which otherwise could have been used for other models. This results in less extensive benchmarking of the models in the research.

2.2 NordPool and Modern Electricity Markets

Electricity markets constitute all markets where participants trade electricity, both as a good or as financial contracts for future trade. As with most other liberal markets, its genuine role is to match supply and demand to determine a market-clearing price across its areas. However, as electricity is economically non-storable, demand inelastic and production dependent on planning, electricity prices are often highly volatile, seasonal and hard to predict (Weron 2014). While there are variations in mechanisms and price dynamics across different power markets, this thesis focuses on the *NordPool* power market. NordPool, operating in Northern Europe and covering 16 European countries (most notably the Nordics, Germany, and the UK), accommodate all common mechanisms of a liberal wholesale electricity market, incl. trading, clearing and settlement (NordPool 2021).

The liberal market situation in the Nordics, with private market participants, is a result of a number of pro-market reforms in the late 80s and early 90s (Blazquez et al. 2018). Schweppe et al. (1988) presented the idea that free electricity markets might increase social welfare. Not soon after, several political electricity free-market reforms were enacted, with the Norwegians parliament's decision to deregulate trading of electrical energy going into effect in 1991 (NordPool 2022). Then five years later, in 1996, NordPool was established as a joint Norwegian-Swedish power exchange (NordPool 2022). Finland and Denmark joined the exchange in 1998 and 2000, making the Nordic electricity market fully integrated. A market previously controlled by vertically-integrated state monopolies had now shifted over to a competitive market, with private players all across the value chain. Figure 8 shows a simplified visualization of the current free-market model, in which private power producers (e.g., TrønderEnergi) sells their electricity through a wholesale market (NordPool) either directly to the consumer or through a retail market. In a market such as NordPool, a marginalist pricing model is used, in which the price for all buyers and sellers is set equal to the price of the last sold unit, the market-clearing price, at which supply equals demand (Blazquez et al. 2018). Bye and Hope (2005) investigated the effect of deregulation in Norway, concluding that it had resulted in lower electricity prices, reduced price inequality across regions, and reduced investment cost while also increasing return on investment on new production capacity. In 2021 and 2022, there has been much public debate around the subject of free electricity markets, as abnormally high electricity prices have hurt consumers while many power companies have experienced record-high financial returns. Given the necessity of electricity, many experts argue that there needs to be stricter regulation to ensure that consumers are protected.

Compared to other commodities such as oil, an important characteristic of electricity markets is that electricity is economically non-storable. There is, therefore, a requirement for a constant balance between consumption and generation, which need to be connected through the power grid (Kaminski 2013). The need for constant balance is one of the main reasons that there often are significant variations in prices across different times of the day or that there might be sudden spikes in prices. When talking about electricity generation, one often talks about the merit order, which is a way of ranking different sources of generation according to marginal cost (Roldan-Fernandez et al. 2016). A typical merit order in electricity markets is shown in Figure 9, in which *variable renewable energies* (VRE) such as solar and wind have the lowest cost, followed by *baseload* such as nuclear and hydro. In the end, there is *peaker* generation (used at times of peak demand) such as oil & gas. The logic is that the higher the price, the more expensive generation sources are utilised. At the end of the spectrum, one can also talk about load shedding, which is the practice of shutting down energy-intensive processes to save electricity. Hence, there is a large gap in potential generation prices, from variable renewable energy, which can even be negative, to oil gas or load shedding, which are costly sources of electricity. The constant balance can be shown in Figure 10, in which there must be put as much electricity grid as there is taken out.

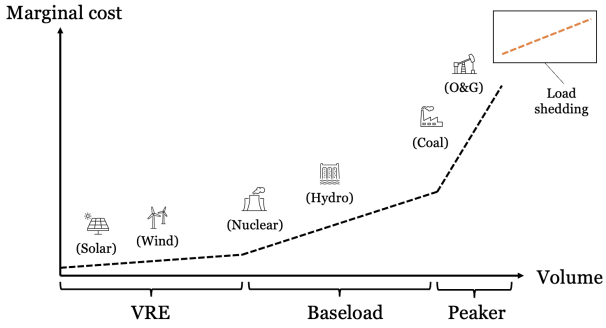


Figure 9: Electricity merit order, ranking sources of electricity generation from VRE to baseload and peaker capacities. Also added simplified explanation of load shedding in the merit order graph, although it technically not being a source of electricity

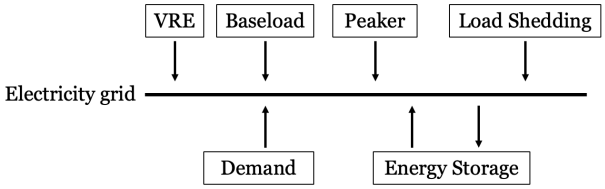


Figure 10: The balance equation of the electricity grid, which must be satisfied at all times. The input, consisting of VRE, baseload generation, peaker generation, load shedding and potential stored energy used, must equal the output, consisting of the demand and energy stored

Most of the NordPool trading is done through day-ahead auctions, as system operators require advance notice to deliver electricity to ensure a constant balance between generation and consumption (Weron 2014). In the day-ahead market, actors bid on electricity with delivery in specific hour on the next day, as shown in Figure 11. In addi-

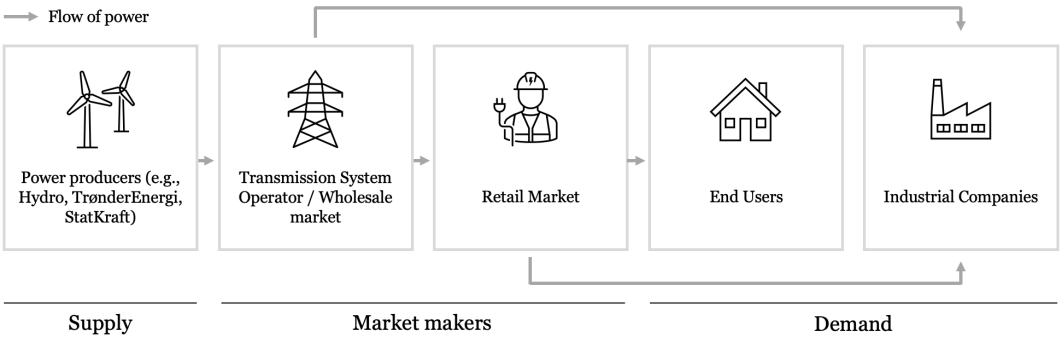


Figure 8: Visualization of the Nordic electricity market model, in which private power producers sells their electricity through a wholesale market (NordPool) either directly to consumer or through a retail market

tion, the system operator operates the real-time or intra-day market at very short horizons before delivery. This auxiliary market is used to price minor deviations in the day-ahead market to ensure a perfect balance in the electricity grid (Weron 2014). As most of the trade is made through the day-ahead market, it is in Europe convention to refer to the day-ahead price as the spot-price (Weron 2014). Like other power markets, NordPool hosts several pool-type auctions. The NordPool auctions are two-sided auctions in which a uniform market clearing price is set at the interception between the supply and demand curve, as seen in Figure 12. It is worth noting that NordPool serves several bidding areas (as seen in Figure 13), which might have differences in price due to capacity bottlenecks. However, when ignoring this and looking at aggregated supply and demand curves across the whole market, the market-clearing price is referred to as the *system price*.

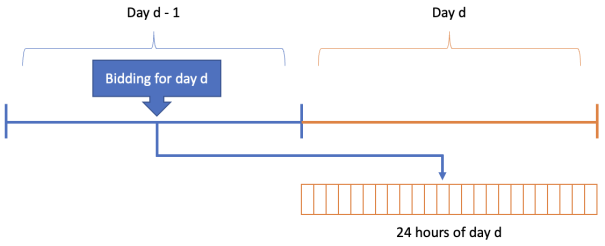


Figure 11: The bidding structure in a day-ahead market, in which bids for day d must be submitted before a certain closing time at day $d-1$. This as system operators require advance notice to deliver electricity to ensure a constant balance between generation and consumption

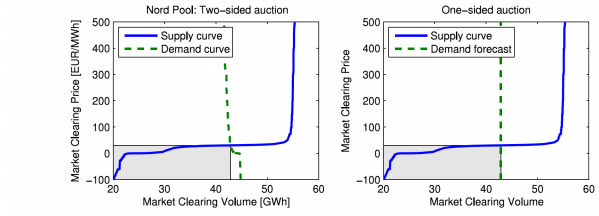


Figure 12: Types of auctions on power exchanges. Left panel represents a two-sided auction, as used on NordPool, in which the market clearing price is the intercept between aggregate supply and demand curves. Right panel represents a hypothetical one sided auction in which there is a set demand. Figure from (Weron 2014)

The NordPool power market is separated into over 20 bidding areas, as seen in Figure 13. However, this thesis only focuses on the 12 bidding areas across the Nordic region (Norway, Sweden, Denmark and Finland). Each region has an individual electricity price at each time interval. As seen in Figure 14, regional price differences occur when there are bottlenecks in the flow capacity between different regions. Given available capacity, transfer of electricity between regions through the NordPool market would ensure price convergence (Weron 2014). This is why neighbouring regions with a lot of grid capacity often exhibit near-identical prices, such as SE2 and SE3. However, in late 2021, bottlenecks created huge regional price differences across the Norwegian bidding regions. At its most extreme, on 16 October 2021, the daily price of electricity in Oslo was 13x that in Trondheim (98.8 €/MWh vs 7.6 €/MWh), a relief to the Trondheim based authors of this thesis.

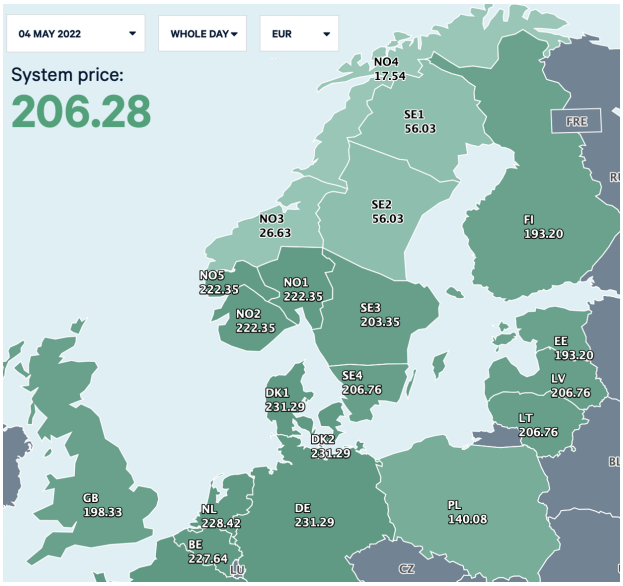


Figure 13: Visualization of the 21 NordPool bidding regions, with daily prices (EUR/ MWh) on the 4 May 2022. Only the 12 bidding regions in Norway (NO1-5), Sweden (SE1-4), Denmark (DK1-2), and Finland (FI) are included in this thesis

Given the state of the electricity market, with a continuous need for market-clearing, there are large variations in price, including a large number of price spikes. Furthermore, this effect has been further strengthened by the

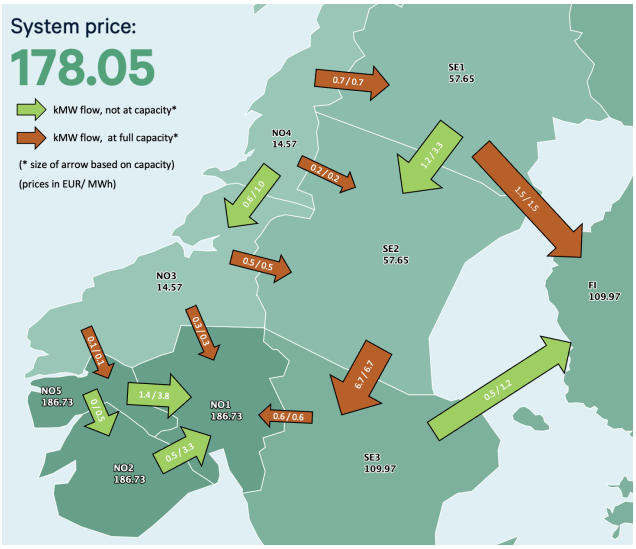


Figure 14: Prices (EUR/ MWh) and cross regional flow in selected Nordic regions 14.00-15.00 on 24 March 2022. Regions with capacity bottlenecks experience exhibit significant price differences (e.g., NO3 and NO1), while regions with available capacity exhibit near identical prices (e.g, NO1, NO2 and NO5)

increased share of variable renewable energy generation, which according to Khare et al. (2016) has contributed to more volatile prices due to its stochastic production patterns. Additionally, highly inelastic demand in the short term, few even conscious of the electricity, makes the prices even more volatile (Burke and Abayasekara 2017). Looking at Figure 15, showing the hourly NordPool over an arbitrary three day period (19.06.2017 - 25.06.2017), shows that there are also large variations in price during the day. The long-term demand is somewhat more elastic due to changing demand from the energy-intensive industry. According to Härdle and Trueck (2010) demand is also the main explanatory factor for seasonal variations. Ensuring stable electricity prices is an important challenge within the energy sector. In February 2021, Texas experienced a major power crisis due to freezing temperatures and bad weather. At its most extreme the electricity price reached as high as 9,000 $/MWh (200x regular rates) (Pechman and Nethercutt 2021). Over 200 people died as a result of the crisis (Hauser and Sandoval 2021), highlighting the societal importance of ensuring steady and reliable prices and access to energy during extreme spikes.

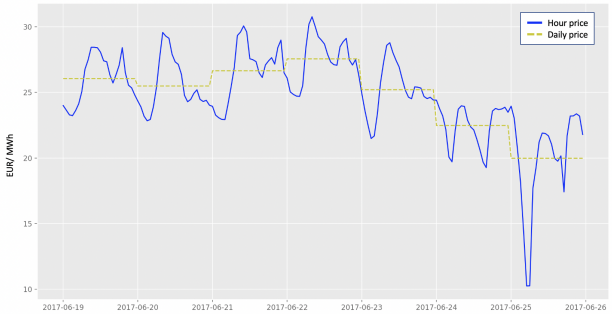


Figure 15: Hourly and daily system price (€/MWh) in period 00:00 19.06.2017 - 23:00 25.06.2017, showing significant variations in price compared to other more easily storable commodities such as oil

In Norway, over 90% of electricity generation comes from hydropower (Statista 2019), which requires a high degree of production planning to optimise cost-efficiency. The Norwegian hydropower system gathers precipitation from a cascade of rivers through a number of reservoirs. Therefore, operators require efficient forecasting models to optimise their available water resources. Production planning often works with requirements such as settling time, rise time, bandwidth and disturbance rejection (Kragelund et al. 2010). In addition to being an essential part of the energy supply, hydropower is also seen as an important part of the intra-day balancing market. 14 days is a typical planning horizon for short-term production planning (Fleten and Krogh 2008), emphasising the importance of mid-term forecasting models in the NordPool regions, which are heavily reliant on hydropower.

3 Literature Review and Contributions

This literature will present relevant research in the realm of computationally intelligent methods (Section 2) for electricity price forecasting, in addition to literature assessing the performance of temporal convolutional neural networks. The practical uses of accurate electricity price forecasts are substantial since an accurate forecast will enable electricity producers to assess the value of production capacity and plan for future production in an economically sustainable manner. This section consists of two main parts. First, we describe research where computationally intelligent methods have been used for electricity price forecasting. Second, we present literature for sequence analysis by temporal convolutional neural networks. The papers presented in this section are summaries in Table 1.

The use of computationally intelligent methods for electricity price forecasting has become increasingly frequent, which is highlighted by both Lago, Marcjasz et al. (2021) and Engebretsen et al. (2020). The methods available for this type of forecasting are numerous, such as long short-term memory, deep neural networks and support vector regression models. The growth in the research field and diversity among forecasting models makes it an interesting field to study. Our method of study has been based on the literature review conducted by Lago, Marcjasz et al. (2021). They used a *Scopus web crawler*[2]. We have conducted the same search query as Lago, Marcjasz et al. (2021) and read the approximately 105 papers which have appeared after the time of writing Lago, Marcjasz et al. (2021), which also highlights the growth in this research area. Naturally, we have also read the papers referenced by Lago, Ridder et al. (2018). Additionally, a Scopus query was conducted to find papers in which the forecasting horizon was longer than the day-ahead market while also using computationally intelligent methods. A common feature, which has also been highlighted by other papers, including Lago, Marcjasz et al. (2021), is that there are often capable models which are being proposed. However, these models often struggle with two limitations. Firstly, the models often suffer from limited testing. This can be either a limited testing horizon, one of only a few markets, or both. Secondly, the models are not necessarily benchmarked against what is known as state-of-the-art models, but only simpler models, with which a lot less work is done. Naturally, this demonstrates some of the forecasting capabilities of the model. However, it is challenging to understand if the model is at the same level as what is regarded as state-of-the-art forecasting models. A reason for little benchmarking against other state-of-the-art models might be the unavailability of the code or hyperparameters used for model creation. Thus, making state-of-the-art benchmarking challenging. The reason for the unavailability of code is unknown, but Croonenbroeck and Stadtmann (2019) describe how some papers are tied by non-disclosure agreements. A contrast to this problem is the *epftoolbox* created by Lago, Marcjasz et al. (2021), who also highlighted this issue. The *epftoolbox* contains two CI models; A deep neural network and a lasso estimated autoregressive model. These models are beneficial for assessing the performance of our proposed model.

Using computationally intelligent models for electricity price forecasting often involves deep learning models, and more specifically, deep neural networks and recurrent neural networks are deployed. The long short-term memory and gated recurrent architectures are the most frequent of the latter category. These architectures are described in Section 2.The first of the two architectures is the more frequent architecture of the two. Likely due to the long short-term memory architecture being older than the gated recurrent unit, although papers such as Yang and Schell (2022) highlight that gated recurrent units are more computationally efficient. One of the most comprehensive benchmarks of both gated recurrent units and long short-memory models was conducted by Lago, Ridder et al. (2018). They compared 27 models in the Belgian electricity market. The long short-term memory and the gated recurrent unit performed almost identically and were only beaten by the deep neural network. These results were likely due to the deep neural network having fewer assumptions about the input data (Lago, Ridder et al. 2018). These results are engaging since the Belgian market is geographically close to the NordPool regions. These findings are somewhat dissimilar to the findings of Meier et al. (2019). They found that the deep neural networks and long short-term memory performed similarly, likely since they both had lagged variables as input. However, the two studies are not entirely comparable since the first study was conducted in Belgium and the second in Germany. A long short-term memory model was also proposed by Zihan et al. (2019). Additionally, this model had advanced preprocessing techniques, such as wavelet decomposition of the electricity price. This modified long short-term memory model outperformed all benchmarks, of which a regular long short-term memory model was one, on all weekdays for the french market. The test period for the models was approximately 40 days. Therefore, one should further test the proposed model to obtain more reliable results (Croonenbroeck and Stadtmann 2019). These test improvements include both other regions and longer periods, making it simpler to compare it to models such as that proposed by Meier et al. (2019). The forecasting powers of the long short-term memory were also displayed by Aineto et al. (2019). Furthermore, the researchers also displayed how incrementally adding additional relevant exogenous variables for electricity price forecasting reduces the mean average percentage error. Although the researchers did not benchmark their models against other deep learning models or statistical models, their findings are interesting and point towards additional explanatory value when including exogenous variables. Looking at markets in NordPool, Atef and Eltawil (2019) compared a long short-term memory model to a support vector regression model. The proposed deep learning model was notably better at forecasting. However, more conclusions could have been made from this study with a broader range of benchmarking models, which would have made it more comparable to the study by Lago, Ridder et al. (2018).

Long short-term memory models have also been used with other deep learning techniques. An example of such is

[2]With the search query: TITLE-ABS-KEY((((“forecasting electric- ity”) OR (“predicting electricity”)) AND ((“electric- ity spot”) OR (“electricity day-ahead”) OR (“electric- ity price”))) OR (((“price forecasting”) OR (“price prediction”) OR (“forecasting price”) OR (“predict- ing price”) OR (“forecasting spikes”) OR (“forecast- ing VAR”)) AND ((“electricity spot price”) OR (“elec- tricity price”) OR (“electricity market”) OR (“day- ahead market”) OR (“power market”))) AND (“deep”) AND (“learning”))

Rantonen and Korpihalkola (2020). They used a combination of a convolutional neural network and a long short-term memory for electricity price forecasting in the Finnish NordPool market. However, the encoding of the convolutional neural network, which was inputted in the long short-term memory, could not improve the forecast. On the contrary, a regular long short-term memory model proved better. Comparing this finding to that of Zihan et al. (2019), who found the wavelet transform to be helpful, it is notable that adding processing or encoding features to a long short-term memory model will not necessarily improve the model. Therefore, hyperparameters and processes must be tuned deliberately. A more successful adaptation to a long short-term model was made by F. Zhang et al. (2022), who added bidirectionality to a long short-term model. The model was benchmarked against a regular gated recurrent unit and a long short-term memory. The bidirectional long short-term memory model proved superior in the Swedish regions 1-3. However, both the regular long short-term memory model and the gated recurrent unit performed well. A benchmarking of different complex long short-term memory architectures in the NordPool area was conducted by Li and Becker (2021). Although their findings are interesting, they are hard to contextualize since there are no statistical methods or similar used as benchmarks.

Apart from recurrent neural networks, the deep learning forecasting models preferred are deep neural networks (Lago, Ridder et al. 2018), described in Section 2. More specifically, Lago, Ridder et al. (2018) found the deep neural network to significantly outperform all other deep learning models when researching the Belgian market, including architectures based on recurrent neural networks. To build upon these findings, Lago, Marcjasz et al. (2021) found the Deep Neural Network to outperform the lasso estimated autoregressive model, which they also believed to be state-of-the-art. This finding was done in the NordPool area and is highly relevant to our research. However, the out-of-sample period was from the end of 2016 to 2018. This period deviates from our out-of-sample period, which is the year 2020. Furthermore, their forecasts were made on the system price and not the individual bidding areas. Therefore, we might obtain different results, but using a deep neural network for electricity price forecasting in the NordPool area will be important for the validation of our study. A reason for the excellent performance of the deep neural network is highlighted by Lago, Ridder et al. (2018). The models do not assume anything about the input and can model the non-linear relationships when forecasting electricity prices. The results found by Ugurlu, Tas et al. (2018) found that the neural network had the lowest mean absolute error compared to other benchmarks such as gated recurrent unit and long short-term memory. However, there was no significant improvement in the artificial neural network compared to the gated recurrent unit and the long short-term memory to conclude that it is a superior forecasting model. These findings are also in line with the findings of Meier et al. (2019), as described previously. Neural networks were also benchmarked in the German and Austrian EEX-markets by Schnürch and Wagner (2020) against a random forest model. Although the neural networks proposed were performing superior to the random forest, the differences in error were insufficient to indicate significantly better forecasting properties than the simpler proposed methods. Post-processing is an available tool for enhancing model performance. For example, Kontogiannis et al. (2022) used post-processing to enhance the performance of the deep neural network created by Lago, Marcjasz et al. (2021). Their post-processing was an autoregressive model with exogenous variables. This was able to halve the mean absolute error of the neural network. So far, we have presented papers with a short forecasting horizon. The reason for this is that most research is within this horizon. Nevertheless, Windler et al. (2019) proposed a deep neural network for forecasting the next 29 days. They conducted more than 1350 out-of-sample evaluations and found that the deep neural network was notably forecasting better than the benchmarks. However, the benchmarks consisted of rarely used models in the literature, and it is unknown how generalizable these results are.

Temporal convolutional neural networks (TCN) were first benchmarked by (Bai et al. 2018), who achieved strong results in time series forecasting when comparing the capabilities of a TCN to machine learning models specialized for time series forecasting such as long short-term memory (LSTM) and gated recurrent unit (GRU). However, despite the novelty of the temporal convolutional neural network, some research has investigated its capabilities for different time series forecasting problems. Most adjacent to the work in this thesis is the work of (Wan et al. 2019), who benchmarked the performance of two different TCN-networks in the ISO-NE region. Their standard temporal convolutional neural network was on par with leading machine learning architectures for time series forecasting, such as the long short-term memory. However, Wan et al. (2019) also found that by implementing a feature called attention, the performance of the temporal convolutional neural network further increased. Implementing attention in a neural network enables the deep learning model to distinctly weigh different input steps of the models. Hence, being able to extract more information from the input data. Previous to Wan et al. (2019), Kuo and Huang (2018) implemented a hybrid model, consisting of both long short-term memory architecture and convolutional neural network-architecture, named EPNet. EPNet forecasted the electricity price in the PJM Regularization zone, outperforming statistical and deep learning architecture-based benchmarks. In addition, the study also compared a convolutional neural network with both long short-term memory, multi-layer perceptron, random forest, and a support vector machine. Of these forecasting models, the multi-layer perceptron, long short-term memory, and EPNet were capable of marginally obtaining a lower root mean squared error. These findings are a contrast to the findings of both Wan et al. (2019) and Bai et al. (2018), who found the temporal convolutional neural network to be superior for time series forecasting. There can be several reasons for these differences. One such reason is implementation differences between the different research teams. Furthermore, power markets differ in energy production mixes and consumption, resulting in some models being more capable in specific electricity regions than others.

As previously mentioned, Lago, Ridder et al. (2018) conduct an extensive benchmarking of both statistically and deep learning-based forecasting models. They find that the convolutional neural network is as accurate as the gated recurrent unit, long short-term memory, or the deep neural network for electricity price forecasting. Although the convolutional neural network was capable of outperforming

the majority of statistical models, the study by Lago, Ridder et al. (2018) points toward the fact that there are deep learning architectures that are more capable of electricity price forecasting. However, the research was solely conducted in the Belgian market, and convolutional neural networks might perform better in the Nordic regions.

From the literature we have read and presented here, we seek to fill the following gaps in the research literature.

1. We will use computationally intelligent models to forecast the hourly prices for all the NordPool regions for the next seven days. Computationally intelligent models have been tested in the NordPool area. However, not with a seven-day forecasting horizon on individual regions.

2. Implementation and benchmarking of a hybrid model for the forecasting task described above. Our proposed model (ENTCN) consists of both a statistical (Enhanced Naive) and computationally intelligent (TCN) component. As we have highlighted in this section, the temporal convolutional neural network has not been tested extensively for electricity price forecasting.

Table 1: Papers references in the literature review

Category	Reference	Description/model	Market	Horizon	Input data
Literature review/ Benchmark	(Engebretsen et al. 2020)	Literature review	N/A	N/A	
	(Lago, Ridder et al. 2018)	Benchmarking of 27 models	EPEX-Belgium	Day-ahead	EPEX-Belgium and -France prices, load forecast
	(Lago, Marcjasz et al. 2021)	comparing LEAR and DNN	NordPool	Day-ahead	NordPool, PJM, EPEX-DE, BE and FR
CI	(Yang and Schell 2022)	Hybrid model of GRU and CNN	NY-ISO	Real-time	Price data from 49 generators
	(Meier et al. 2019)	ANN compared to LSTM	EPEX-DE	Monthly	EPEX-DE prices
	(Zihan et al. 2019)	LSTM with wavelet transformed input	EPEX-FR	Day-ahead prices	Day-ahead EPEX-FR prices
	(Schnürch and Wagner 2020)	ANN and random forrest	EPEX-DE and EPEX-AU	Day-ahead price	Bidding curves and past prices
	(Aineto et al. 2019)	LSTM and DNN	MIBEL	Day-ahead market	Past prices, demand and supply forecasts
	(Atef and Eltawil 2019)	LSTM	NordPool-DK		Past prices, load and wind load
	(Rantonen and Korpihalkola 2020)	Hybrid of CNN and LSTM	NordPool-FI	Day-ahead	Past NordPool-FI prices, generation and consumption forecasts and UK electricity prices
	(F. Zhang et al. 2022)	Bidirectional LSTM compared to LSTM, GRU, MLP and SVR	NordPool-(SE1-3)	200 hrs	Past prices
	(Li and Becker 2021)	Different LSTM versions	NordPool and sub-markets	Day-ahead	All past NordPool area prices, consumption and production forecasts and transmission capacities and past load
	(Kontogiannis et al. 2022)	DNN with post processing	NordPool System price wind and load	Day ahead	Past prices and forecasts for day-ahead
	(Windler et al. 2019)	DNN	EEX-DE and -AU	day-ahead to one month	Past prices
TCN/CNN	(Bai et al. 2018)	TCN seed paper	Sequences analysis tasks	N/A	Problem dependent
	(Wan et al. 2019)	TCN and TCN with attention	ISO-NE	Day-ahead	Past prices
	(Kuo and Huang 2018)	Hybrid model of CNN and LSTM compared to multiple benchmarks	PJM	1 hour	Past prices

4 Data

In this section, the data used in this thesis is presented. The data can be divided into two main groups - area electricity prices and exogenous variables. Firstly, we will present the area electricity prices, the dependent variables. Secondly, we will analyze the exogenous variables and their relation to the electricity price. The sampling frequency and source of the variables are displayed in Table 7. The variables which have a lower sampling frequency than hourly are filled forward. Examples are oil and gas prices.

4.1 Area Prices

This section analyses the NordPool area prices, the independent variables in this thesis. More specifically these are the regions *NO1, NO2, NO3, NO4, NO5, SE1, SE2, SE3, SE4, DK1, DK2* and *FI*. The regions will be analysed individually and with respect to calendar effects. The geographical span of each region can be viewed in Figure 13. Prices in the different areas range from -60.26 €/MWh to 255.02 €/MWh, as seen in Table 2. Although the price regions are adjacent, both the average price and the price risk differ across all regions, with the latter observable as the difference in standard deviation. The least expensive price zone is NO5, with an average price of 27.53 €/MWh FI has the highest average price with 35.73 €/MWh. The price risk is lowest in the NO4 with a standard deviation of 12.88, and the highest price risk is in FI, with a standard deviation of 16.12. Furthermore, all area prices exhibit stationarity through an *augmented Dickey-Fuller* test, with a high degree of significance. This finding means a constant mean and variance in the data. Data with stationary properties will enable us to utilize models such as ARMA and regression for price forecasting (Brooks 2019).

Table 2, Table 3 and Figure 17 can be used to describe the distribution of the electricity price in the NO-areas. Looking at the skew, which is positive for all price areas, it is evident that the prices are shifted to the left. The skew is also visible from Figure 17 with a mean higher than the median and a longer tail on the right-hand side. Note that not the entire price range is included, as prices reached as high as 255 €/MWh in the in-sample data, and minimally additional insight was obtained by including these in the histograms. This is also evident by comparing the value of the 99^{th} percentile in Table 3 with the maximum value in Table 2. At most 1% of the prices in all NO-regions are above 60 €/MWh. The kurtosis of the data is described by the *Fischer kurtosis* (Brooks 2019). The Fischer kurtosis describes the fatness of the tails and should not be used for describing the peakedness of the data, as stated by Westfall (2014). All the areas have positive Fisher kurtosis levels, indicating leptokurtic distributions. This means that the tails are fatter than a regular, symmetric, Gaussian distribution. Consequently, the variation in our dataset is greater than a dataset with a regular Gaussian distribution. It might be tempting to conclude that this also implies that the forecasting problem becomes more challenging, but that also assumes less correlation between the forecasted values and the input variables. Therefore, the kurtosis solely states how the dependent variables vary in the training data. Comparing the different kurtosis values of the different areas in Table 2, it is noticeable that they significantly differ in magnitude. An example is comparing NO1 with NO2, which have 13.55 and 1.26 in kurtosis, respectively. There can be numerous reasons for these differences, such as energy mixtures, weather variations, and market coupling.

$$r_k = \frac{\sum_{t=k+1}^{n}(y_t - \bar{y})(y_{t-k} - y)}{\sum_{t=1}^{n}(y_t - \bar{y})^2} \tag{20}$$

Electricity prices are known to exhibit auto-correlation properties. This is why statistical methods such as auto regression can be used to obtain fairly decent accuracy for electricity price forecasting (Lago, Ridder et al. 2018). The *auto-correlation plot* (ACF-plot) and *partial auto-correlation plot* (PACF-plot) are in Figure 18 and Figure 19, respectively. Additionally, the formula for auto-correlation is represented in Equation 20. From looking at the graphs, some inferences can be made. For example, a reasonable prediction for the next hour is the last hour. This strengthens the reasons for using simple, naive approaches for electricity price forecasting. This is evident as both the substantial spike at lag 1 in the Figure 19 and the high value at lag 1 in the Figure 18. The PACF-plot has some lagged values different from 0, up to lag 16. Therefore, the range of lagged input values can be reasonably long. Looking at the different types of SARIMA models described in Section 2.1.6, one can argue that AR-type models of order 16 can be relevant. This range must naturally be viewed in the context of our available computational power. Furthermore, the ACF plot forms a downwards pointing arc over 24 hours. This arc, naturally, represents the electricity price the previous day at the same hour and is information that can be valuable when implementing SARIMA models. The frequency cycle is likely 24 time steps. This peak in the ACF plot differs slightly from the peak observed at lag 20 (20 hrs. previously) in the PACF-plot, which has values of 0 at lag 24 for the NO-regions. Hence, from these plots, we can conclude that lagged variables are an exceedingly strong candidate for input variables to all the models, since the electricity price exhibits both auto-correlation and partial auto-correlation-properties. The specific values of lagged variables and seasonal components for models will be tested later in this thesis.

In Section 2 we described how electricity prices exhibit calendar effects. Therefore, we will describe how the electricity price fluctuates with respect to different groups of calendar variables. This can be insightful since the calendar variables for the future a known prior to the forecast and can improve the forecasting accuracy.

From Figure 16, one can observe that the electricity price varies throughout the year, with higher prices during the winter and lower prices in the summer months and holidays. For example, in NO3, June and July are the least expensive months, whereas November and January/February exhibit higher price levels than the rest of the year. More specifically, these observations can be quantified by looking at how much the average price of a month will differ from the average price of a year. The area price deviations for NO1, NO2, NO3, NO4, and NO5 are shown in Table 4. From these tables, it is evident that November is the month with higher prices, whereas June has the lowest prices. Demand for heating is the main driver for high prices during winter in the Nordic region, as there is min-

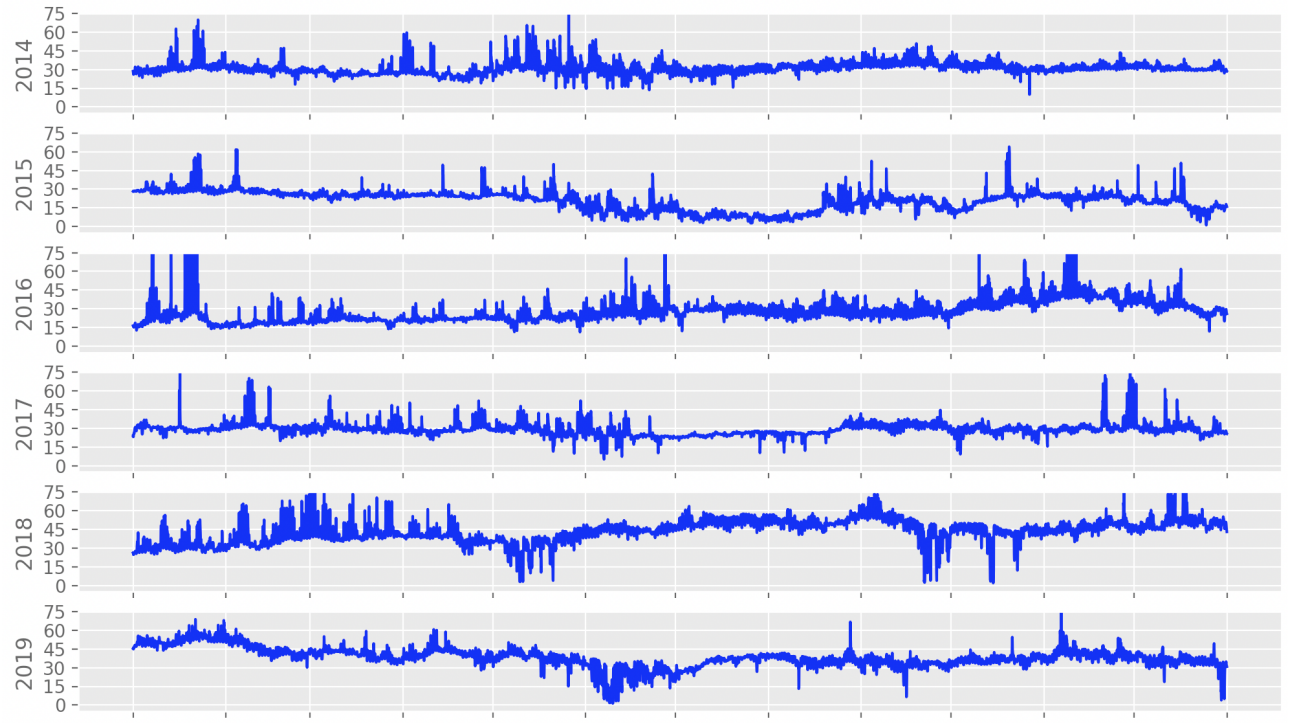


Figure 16: Hourly prices (€/MWh) in NO3 for the in-sample data (1 Jan 2014 - 31 Dec 2019)

Table 2: Descriptive statistics of hourly area prices from the NordPool area (€/MWh), (1 Jan 2014 - 31 Dec 2019).

Metric	NO1	NO2	NO3	NO4	NO5	SE1	SE2	SE3	SE4	DK1	DK2	FI
Average	27.79	27.54	29.00	27.64	27.53	29.84	29.84	31.16	32.64	31.11	33.21	35.73
Median	27.70	27.58	29.40	27.10	27.58	29.79	29.79	30.38	31.09	30.31	31.53	33.95
Standard deviation	13.46	12.92	13.17	12.88	12.90	13.37	13.37	14.14	14.79	14.58	15.46	16.12
Mean average deviation	10.03	9.81	9.88	9.71	9.86	9.90	9.90	10.18	10.73	10.79	10.17	11.68
Min	-1.73	-1.73	0.00	0.00	-0.09	-1.73	-1.73	-1.73	-1.94	-60.24	-60.26	-1.73
Max	255.00	114.70	255.02	255.02	114.70	255.02	255.02	255.02	255.02	200.04	255.02	255.02
Skew	1.47	0.60	1.35	0.47	0.54	1.41	1.41	1.52	1.51	0.39	1.22	1.98
Fisher kurtosis	13.55	1.26	15.42	16.12	1.00	13.72	13.72	13.10	10.40	2.28	9.26	15.07
Stationary (ADF)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 3: Price quantiles over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all NO-regions.

Quantile	NO1	NO2	NO3	NO4	NO5
0.01	7.23	7.62	7.64	7.71	6.77
0.05	13.58	13.51	15.91	15.4	13.48
0.1	18.08	18.03	20.57	19.7	17.94
0.15	21.15	21.11	22.83	21.37	20.77
0.5	29.45	29.37	30.96	28.99	29.36
0.85	42.41	41.99	43.16	42.33	42.00
0.9	46.25	45.78	46.88	46.15	45.7
0.95	51.12	50.77	51.52	50.41	50.66
0.99	59.95	57.75	58.74	55.79	57.31

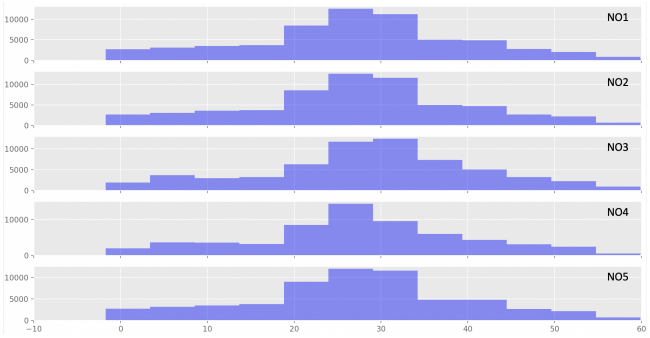


Figure 17: Price histogram over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all NO-regions. The histograms are cut off at €/MWh 60, and cover 99% of the distribution.

imal demand for cooling in warm periods Hellström et al. (2012). However, Table 4 also shows that the monthly variations are not identical across the regions. These differences come from differences in supply and demand across the regions and limited flow capacities between the areas. In Section B, similar tables for the remaining NordPool regions are presented. Looking at all the tables for the monthly coefficients, we can infer that November has the highest prices, whereas the month with the lowest prices varies across regions. An example is DK1, where March has the lowest average prices for the in-sample data. The reason for the deviation of DK1 is likely due to the energy generation mix in Denmark. It has a higher share of electricity production from wind generation than, e.g., hydro generation dominated Norway, which results in different monthly fluctuations.

The electricity price varies over a week, with weekdays being more expensive than weekends and holidays. These differences are due to the varying demand from differences

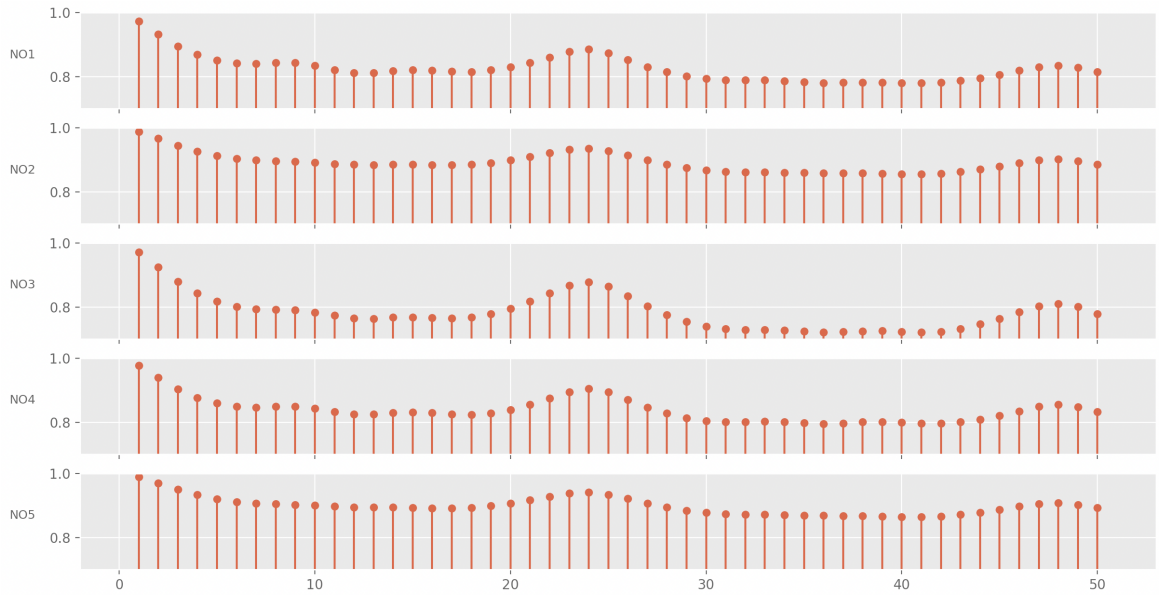


Figure 18: ACF-plot over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all NO-regions.

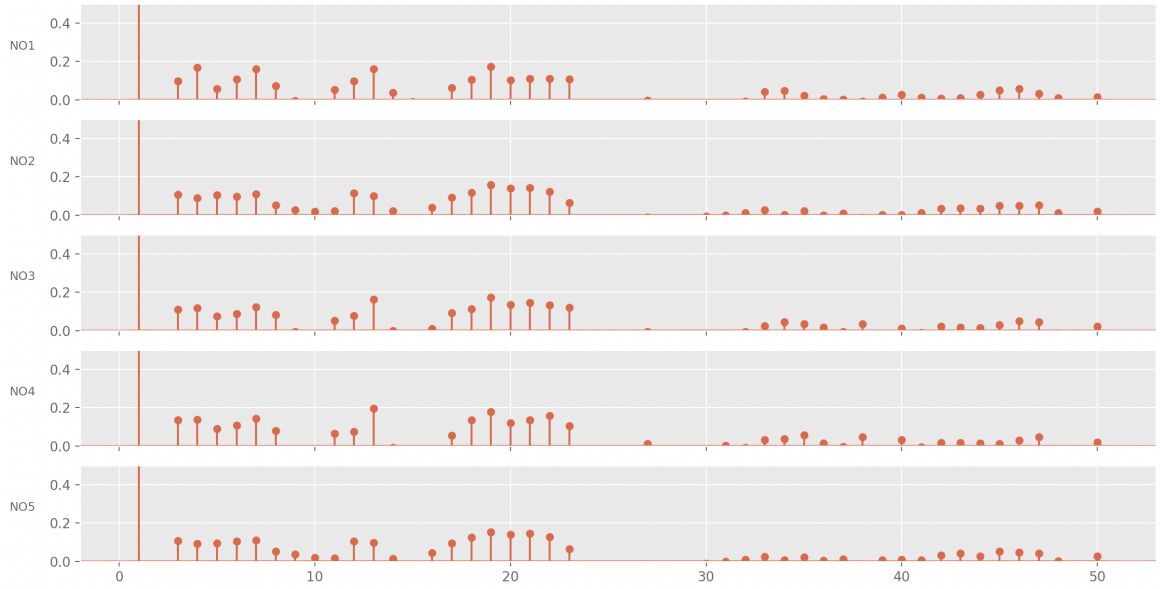


Figure 19: PACF-plot over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all NO-regions. The value at lag 1 is at approximately 0.99.

Table 4: Monthly coefficients over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all NO-regions. Bold coefficients are yearly maximums or minimums

Month	NO1	NO2	NO3	NO4	NO5
January	1.143	1.120	1.066	1.080	1.115
February	1.046	1.053	1.007	1.016	1.056
March	1.025	1.021	0.988	0.995	1.026
April	0.973	0.975	0.960	0.968	0.984
May	0.884	0.893	0.949	0.971	0.895
June	**0.835**	**0.844**	**0.889**	**0.880**	**0.844**
July	0.932	0.936	0.929	0.913	0.932
August	0.946	0.953	0.993	0.963	0.947
September	0.951	0.958	1.034	1.030	0.948
October	1.014	1.010	1.029	1.027	1.006
November	**1.156**	**1.148**	**1.127**	**1.115**	**1.154**
December	1.095	1.091	1.030	1.043	1.096

in human activity on different weekdays and holidays, as described in Section 2 and by Hellström et al. (2012). Table 5 displays the weekday and holidays coefficients for all NO-regions, while the same coefficients for the remaining NordPool areas can be found in the appendix. The price is highest above the average price on Mondays, Tuesdays, Wednesdays, and Thursdays in the Norwegian bidding areas. The prices are between 2% and 7% above the weekly average for these days. Fridays are days with lower activity, and the price falls somewhat but is still above the weekly average. However, during weekends and, especially holidays, the electricity price falls below the average. Holidays have the lowest average price compared to the two weekend days, with prices around 10% below average for the holidays. Additionally, Sundays have lower prices than Saturdays. On Sundays, the prices in the NO-regions are about $5-7\%$ below average, whereas Saturdays are about

$4-6\%$ below weekly averages.

Table 5: Weekday and holiday coefficients over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all NO-regions. Bold weekdays are either weekly maximums or minimums.

Day	NO1	NO2	NO3	NO4	NO5
Monday	1.019	1.019	1.021	1.071	1.016
Tuesday	1.026	**1.026**	**1.037**	1.025	1.022
Wednesday	1.021	1.022	1.035	1.028	1.025
Thursday	**1.031**	1.016	1.033	**1.031**	1.018
Friday	1.011	1.010	1.011	1.011	1.011
Saturday	0.956	0.964	0.942	0.953	0.965
Sunday	**0.935**	**0.943**	**0.922**	**0.939**	**0.944**
Holiday	0.908	0.916	0.889	0.930	0.920

Similar patterns to those observed concerning months and weekdays can be observed within the hourly time granularity. These electricity demand rise as human activity increases and, therefore, the electricity price. These hourly coefficients for the NO-regions are displayed in Table 6. For the remaining NordPool regions, the hourly coefficients can be found in the appendix. The NO-area has the lowest prices at 03.00 at night and the highest at 08.00 in the morning. This is understandable since demand is low during the night hours but high as the activity sharply increases in the morning. The prices fall throughout the day, except for a small increase around 17:00 and 18:00. There is a steady decline to the night prices from this local price peak.

Table 6: Hourly coefficients over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all NO-regions. Bold hours are either daily maximums or minimums.

Hour	NO1	NO2	NO3	NO4	NO5
0	0.941	0.950	0.916	0.939	0.951
1	0.915	0.924	0.889	0.915	0.925
2	0.899	0.907	0.872	0.900	0.909
3	**0.892**	**0.900**	**0.865**	**0.893**	**0.902**
4	0.898	0.905	0.874	0.899	0.908
5	0.931	0.938	0.913	0.933	0.941
6	0.976	0.981	0.976	0.984	0.983
7	1.047	1.036	1.052	1.038	1.035
8	**1.088**	**1.071**	**1.102**	**1.073**	**1.067**
9	1.077	1.065	1.095	1.069	1.063
10	1.060	1.053	1.083	1.060	1.052
11	1.044	1.041	1.066	1.048	1.041
12	1.029	1.028	1.048	1.034	1.028
13	1.018	1.018	1.034	1.026	1.019
14	1.011	1.012	1.025	1.020	1.013
15	1.014	1.012	1.026	1.021	1.013
16	1.037	1.028	1.037	1.032	1.026
17	1.066	1.053	1.064	1.051	1.049
18	1.062	1.055	1.067	1.054	1.053
19	1.039	1.040	1.051	1.041	1.039
20	1.018	1.022	1.025	1.023	1.022
21	1.004	1.009	1.006	1.009	1.009
22	0.984	0.992	0.979	0.986	0.992
23	0.951	0.960	0.934	0.953	0.960

The calendar effects analyzed in this section have provided insight into the movement of the electricity price with respect to different time frequencies. However, the month effects are distinct from the other two categories when assessing the magnitude of price increases. For example, weekday effects do not move the price more than $4-10\%$ and hours no more than 10%. On the other hand, months have calendar-effect-coefficients as high as 16% in some price areas. Consequently, it might be more relevant to include months rather than weekdays as input variables if one chooses between the calendar-effect variables. However, as stated previously, all calendar effects have a high level of significance and are available to the group. Therefore, all groups should be tested as input. Furthermore, this thesis investigates forecasts for the next seven days, with lagged values as an input variable, which means that the forecast is often done within the same month. Therefore, including month coefficients might not yield an information gain as significant as weekdays or hours. This theory has to be validated prior to model testing.

4.2 Exogenous Variables

The second group of variables included in this dataset is the exogenous variables - which are variables we believe have the potential to be a valuable input to our models without the data generating process of these variables being an electricity price zone. These variables are all listed in Table 7, except for the system price and bidding area prices. A noticeable feature of the exogenous time series is the sampling frequency, which is lower than the system price and area prices. Nevertheless, we believe these variables might hold explanatory value, so they are included in the data set. This belief is due to the exogenous variables' level and not necessarily fluctuations that can provide insight into the input factors and demand drivers for electricity prices.

Further, the exogenous variables can be divided into production, market, commodities, and weather. Production variables are the variables that either describe production, such as *wind production*, or the potential for production, such as *hydro reservoir*. Weather variables are all variables that are directly linked to weather, such as *snow mass NO*. The commodities variables are variables that represent market prices for commodities that can work as input factors for electricity generation, such as *coal*. Lastly, the market variables are used for describing the market, of which *consumption* is one. It is worth noting that we also included calendar-effect variables in our data set. These are categorical variables where there is no numeric relationship between the variables (Goodfellow et al. 2016). Consequently, they are described in Section 5.

Correlation has the potential to be a powerful indicator for how much the area prices will fluctuate with respect to other variables. Therefore, we have chose to use the Pearson correlation coefficient ρ_{xy}, described by Equation 21. $|\rho_{xy}| \leq 1.0$. Dancey and Reidy (2011) describes the following hierarchy of correlation coefficients: $|\rho_{xy}| \geq 0.6$ is considered strong, $0.3 \geq |\rho_{xy}| \leq 0.6$ is considered moderate and $|\rho_{xy}| \leq 0.3$ is considered weak. Although correlation can be a useful measure of the strength of co-movement between two variables, it should not be mistaken for causation. An example is the third variable problem, where an unknown third variable causes movements in two variables. The two variables will have a strong correlation, but no inferences can be made of causation. Furthermore, spurious correlations can also be present in data. A spurious correlation is when two unrelated variables exhibit a strong correlation due to nothing but randomness. We encourage the interested reader to visit Vigen (2021) for exemplifica-

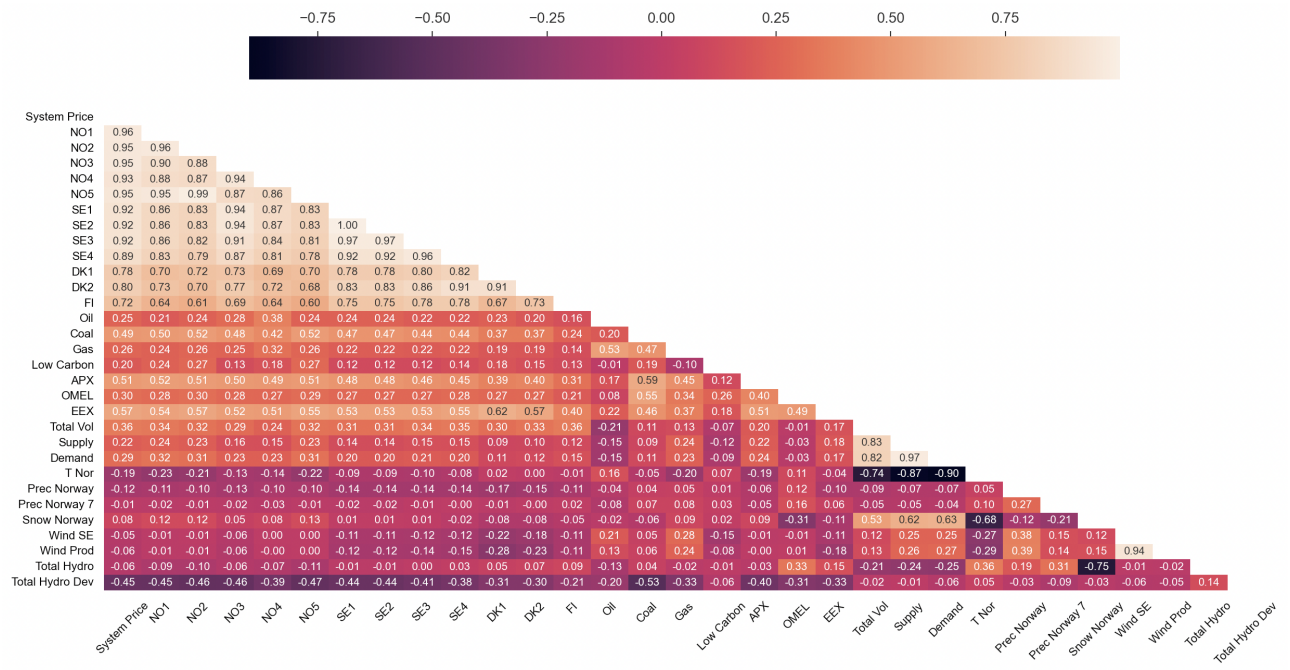


Figure 20: Correlation matrix over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all variables except categorical variables.

Table 7: Summary of time series used. All variables that are not observed on an hourly time frequency are made into hourly samples by filling forward the values.

Data	Week	Day	Hour	From	To	Source	Description
NordPool prices			-	-	-	-	-
System price			x	01.01.2014	31.12.2020	NordPool	NordPool day-ahead system price
Bidding prices (NO, SE, DK, FI)			x	01.01.2014	31.12.2020	NordPool	NordPool day-ahead area price
Market data			-	-	-	-	-
Volume			x	01.01.2014	31.12.2020	NordPool	System market clearing quantity
Production			x	01.01.2014	31.12.2020	NordPool	Production
Consumption			x	01.01.2014	31.12.2020	NordPool	Consumption
EEX system price			x	01.01.2014	31.05.2020	Datastream	EEX - Hourly spot hour
APX system price			x	01.01.2014	31.05.2020	Datastream	APX Power UK$_{E}lec.rpd$
OMEL system price			x	01.01.2014	31.05.2020	Datastream	OMEL-Elec. Spain
Production data			-	-	-	-	-
Hydro reservoir	x			01.01.2014	31.12.2020	NordPool	Aggr. NO-SE level (MWh)
Hydro res. deviation	x			01.01.2014	31.12.2020	NordPool	MWh dev. from normal
Wind production		x		01.01.2014	31.12.2020	NordPool	DK, FI and Baltic
Wind prod. SE		x		01.01.2014	31.12.2020	NordPool	ENTSOSE - SE power statistics
Weather data			-	-	-	-	-
Temperature Norway			x	01.01.2014	31.12.2020	MET(Fros)	Mean of most populated areas
Precipitation NO		x		01.01.2014	31.12.2020	MET(Fros)	Sum of most reservoir dense areas
Prec. NO 7 days		x		01.01.2014	31.12.2020	MET(Fros)	Sum of the last 7 days
Snow mass NO		x		01.01.2014	31.12.2020	Renewables.ninja	Snow mass in NO
Commodity prices			-	-	-	-	-
Oil price		x		01.01.2014	31.12.2020	Datastream	ICE-BRENT CRUDE OIL TR1c
Gas Price		x		01.01.2014	31.12.2020	Datastream	EEX EGIX NCG Index
Coal Price		x		01.01.2014	31.12.2020	Datastream	Coal ICE API2 CIF ARA
Low carbon certificates		x		01.01.2014	31.12.2020	Datastream	Low Carbon 100 Europe Index

tions of such events. As a result, the correlation among the dependent and independent variables indicates which variables have the potential to be included as input variables in our models.

$$\rho_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{21}$$

Looking at Figure 20, it is evident that among the commodities, the coal price has the strongest correlation with the electricity prices in the Nordic region. However, none of the commodity correlation coefficients concerning area prices is considered strongly correlated, only moderately or weakly correlated. Coals correlation with the regional prices is partly due to the coal price functioning as a price setter on the continent, to which much of the Nordic price zones are connected. This market integration is described in Section 2. Hence, increasing coal prices might increase electricity prices on the continent, leading continental Europe to purchase more electricity overseas. In contrast to coal, carbon certificates seem to be linearly uncorrelated with electricity prices. An explanation can be because they constitute a neglectable cost of electricity production and are likely to be less relevant. The other power market variables included in the dataset show a moderate to weak correlation to the NordPool prices. The only exception being the EEX and DK1, likely due to their geographical proximity. Nevertheless, the added explanatory

value of other electricity market prices is unknown, and must be validated. Apart from some of the other electricity markets and the total volume, the market data are weakly or almost uncorrelated with the electricity area prices, with the total volume barely being moderately correlated with the area prices. The absence of correlation with the market group of variables can result from the hydropower generation capabilities of the Nordic regions. Hydropower installations have similar functionalities as batteries and can produce electricity when the prices are the most attractive, of which variable renewable energy sources (e.g., wind, solar) are incapable. All weather-related variables have weak correlation coefficients with the electricity prices. As a result, we will not spend more time discussing this matter Mainly, the production variables are weakly/uncorrelated with the area electricity prices, except for the hydro deviation variable. Hydro reservoir deviation is moderately correlated with all price areas, except for FI. Understandably, the hydropower generators are likely to demand higher prices for electricity production when the water levels are lower than usual. To synthesize, none of the independent exogenous variables strongly correlate with the electricity price. However, a handful of variables are moderately correlated with the electricity price. Most noticeable are the coal price and hydro level deviations. As a result, these variables should be most strongly considered as additional input variables in our models.

5 Method

In this section, the implementation of each model is documented and explained. In addition, the error metrics, statistical tests, and experimental design used are described. The current section assumes the reader has an understanding of the model concepts and terminology explained in Section 2.1. To ensure reproducibility and meaningful results, the thesis follows a list of eight electricity price forecasting best practices listed in Jedrzejewski et al. (2022).

5.1 Model Implementation

The models implemented and the used hyperparameters for all the implemented models are listed in the following subsection. All the models have been fitted or trained separately for each bidding region. However, the hyperparameters used by a specific model are identical for all bidding regions. All implementations in this project are done in the Python programming language.

The hyperparameters used were set to look at the validation loss when training, specifically at the NO1 (Oslo) and NO3 (Trondheim) regions. Many hyperparameters were also set using best practices from the field of electricity price forecasting and state-of-the-art model implementations. A common issue in the electricity price forecasting literature highlighted by Lago, Marcjasz et al. (2021) is the use of ex-post hyperparameter optimization. Examples of this include Yadav et al. (2017) and Peter and Raglend (2017), in which hyperparameters were set based on the out-of-sample performance, which gives the models an unfair and non-existent advantage. Keeping this in mind, in this study, when tuning the hyperparameters, only validation loss was used to indicate performance, to not fit the models to the out-of-sample data.

Table 8: Hyperparamters and settings used across all implemented deep learning models (DNN, ENTCN, LSTM, GRU)

Hyperparameter	Value
Optimizer	SGD
Loss function	Mean absolute error
Learning rate	0.002
Batch size	128
Epochs	30
Validation split	0.05

Although most models have differing hyperparameters, the implemented deep learning models (DNN, ENTCN, LSTM, GRU) have some shared parameters based on deep learning best practices (Goodfellow et al. 2016). These are listed in Table 8. Additionally, all these deep learning models use basic components from the *Tensorflow Keras* architecture(Fu and Aldrich 2018), which is optimised for machine learning tasks in *Python*.

5.1.1 Naive Forecasts

The naive forecasting model implemented is mainly used as a simple benchmark for other models. It is straightforward and forecasts the price at a certain time to be the same as for the same hour and day the previous week, hence 168 hours before, as seen in Equation 22. This implementation is refreed to as the seven-day (7d) naive, or simply the naive benchmark.

$$\hat{y}_t = y_{t-168} \tag{22}$$

The reason for doing this is to account for both the weekday and hourly effects of the electricity price. Using the electricity price 168 time steps previously works since the forecasting horizon is one week. Another alternative naive forecasting model, still persevering the hour effect, would be to forecast the price to be that on the same hour of the last day of the input period. However, this model would not incorporate the day of the week effect.

5.1.2 Deep Neural Network (DNN)

The deep neural network (DNN) model implemented is a simple feed-forward multi-layer perceptron (MLP) with four hidden layers, all being fully connected. The DNN model implemented is inspired by the proposed model by Lago, Marcjasz et al. (2021), which implemented a feed-forward neural network with two hidden layers. As with the implementation in Lago, Marcjasz et al. (2021), we also developed the DNN using the Keras framework in Tensorflow (Keras 2015). As the network took in panel data, the first transformation done is flatten, to ensure one dimensional input into the first hidden layer. Relevant background on the functionality of a deep neural network is provided in Section 2.1.2.

The hyperparameters are set by testing numerous configurations, using the validation loss on the Tr.heim and Oslo regions as the main indicators of model performance. The same set of hyperparameters was used for every bidding region in the thesis. Further work, with more resources, could dedicate time to investigating the use of region-specific hyperparameters to optimise model performance further. Using the ReLU activation function was based on the state-of-the-art model implementation from Lago, Ridder et al. (2018), which also inspired the use of a large number of nodes (100+) in each hidden layer. A summary of the hyperparameters used in the DNN model is presented in Table 8 and Table 9. The model is optimised using stochastic gradient descent (SGD), explained in Section 2.1.2, using mean absolute error as the loss function.

Table 9: Hyperparamters used for the DNN model. Optimizer, loss function, learing rate, batch size, epochs, and validation split are listed in Table 8

Hyperparameter	Value
Input length	3 weeks (504 hours)
# nodes, hidden layer 1	128
# nodes, hidden layer 2	128
# nodes, hidden layer 3	128
# nodes, hidden layer 4	128
hidden layer activation	ReLU
last layer activation	Sigmoid

5.1.3 Long Short-Term Memory (LSTM)

The long short-term memory models implemented in this thesis are two types, a single-layer long short-term memory model and a stacked log short-term memory model. Both

are described in Section 2.1.3. Both models consist of one or more layers with LSTM-cells, which pass their output to a fully connected network. The implementation of the models was done in Python, where we used the Keras framework in TensorFlow (Keras 2015). In order to find the optimal hyperparameters for the different models, a sensitivity search around a starting point was conducted. Then, we calculated the validation loss in NO3 for the proposed hyperparameters, and used the parameters with the lowest validation loss in the NO3 NordPool region. The hyperparameter starting point for the single-layer long short-term memory model was the hyperparameters reported by Lago, Ridder et al. (2018). The best hyperparameters are reported in Table 10. The model was optimized with the stochastic gradient descent algorithm and had a learning rate of 0.002. In addition to a regular LSTM-model, the group also implemented a stacked LSTM model (referred to as S-LSTM). The starting point was the LSTM-model proposed by Lago, Ridder et al. (2018), but with an additional layer of long short-term memory cells. This included adding additional LSTM-layers and hidden neuron layers. Table 11 shows our chosen values for hyperparameters. Optimizer, loss function, learning-rate, batch size, epochs, and validation split are listed in Table 8

Table 10: Hyperparamters used for the LSTM model. Optimizer, loss function, learning-rate, batch size, epochs, and validation split are listed in Table 8

Hyperparameter	Value
Input length	3 weeks (504 hours)
Output units in LSTM-layer	64
LSTM recurrent activation func.	tanh
LSTM activation func.	Sigmoid
# nodes, hidden layer 1	256
Hidden layer actiavtion func.	ReLu
last layer activation func.	ReLu
Dropout	no

Table 11: Hyperparamters used for the S-LSTM model. Optimizer, loss function, learning-rate, batch size, epochs, and validation split are listed in Table 8

Hyperparameter	Value
Input length	3 weeks (504 hours)
Output units S-LSTM-layer 1	128
Output units S-LSTM-layer 2	32
S-LSTM recurrent activation func.	tanh
S-LSTM activation func.	Sigmoid
# nodes, hidden layer 1	128
# nodes, hidden layer 2	64
# nodes, hidden layer 3	64
Hidden layer actiavtion func.	ReLu
last layer activation func.	ReLu
Dropout	0.4

5.1.4 Gated Recurrent Unit (GRU)

This thesis has implemented two types of gated recurrent units, having either a single layer or being stacked (referred to as S-GRU). These model architectures are described in Section 2.1.3. In order to implement the models, we used the Keras framework in TensorFlow (Keras 2015). As with the LSTM-models, a sensitivity search was conducted around initial starting values for hyperparameter tuning. In addition, the validation loss in the NO3-region was used to assess performance. In order to find the starting values for the sensitivity search for hyperparameters, we used the same methodology as with the LSTM and S-LSTM models. However, we did not use the LSTM-parameters of Lago, Ridder et al. (2018), but rather the GRU-parameters. These were also reported by the paper. A summary of the hyperparameters used in the GRU models is presented in Table 12. Optimizer, loss function, learning-rate, batch size, epochs, and validation split are listed in Table 8.

Table 12: Hyperparamters used for the GRU model. Optimizer, loss function, learing rate, batch size, epochs, and validation split are listed in Table 8

Hyperparameter	Value
Input length	3 weeks (504 hours)
Output units in GRU-layer	256
GRU recurrent activation func.	tanh
GRU activation func.	Sigmoid
# nodes, hidden layer 1	256
Hidden layer actiavtion func.	ReLu
last layer activation func.	ReLu
Dropout	no

Table 13: Hyperparamters used for the S-GRU model. Optimizer, loss function, learing rate, batch size, epochs, and validation split are listed in Table 8

Hyperparameter	Value
Input length	3 weeks (504 hours)
Output units S-GRU-layer #1	128
Output units S-GRU-layer #2	128
S-GRU recurrent activation func.	tanh
S-GRU activation func.	Sigmoid
# nodes, hidden layer 1	256
Hidden layer actiavtion func.	ReLu
last layer activation func.	ReLu
Dropout	no

5.1.5 Enhanced Naive Temporal Convolutional Network (ENTCN)

The Enhanced naive temporal convolutional network (ENTCN) model was developed by the authors of this study in the project thesis T. R. Wang et al. (2021). The model is a hybrid model combining, enhanced naive, a form of naive model adjusting for several known effects, with TCN, a temporal convolutional network, which is discussed in Section 2.1.4. In T. R. Wang et al. (2021), the hybrid model was developed to forecast a single-point forecast 14-days ahead in time on the daily NordPool system price, displaying promising results compared to implemented benchmarks. When implementing the model in the current thesis, the model had to be further developed to make hourly forecasts over a seven-day horizon for each NordPool bidding region. A summary of the ENTCN model is shown in Figure 21, in which the DNN block containing the TCN layer (referred to as the TCN model) is trained on forecasting the error of the enhanced naive model.

The enhanced naive is a type of naive forecast, but is adjusted for monthly, weekday, hour and holiday effects. The forecasted value ($\hat{y}$) at time t (t) can be expressed with

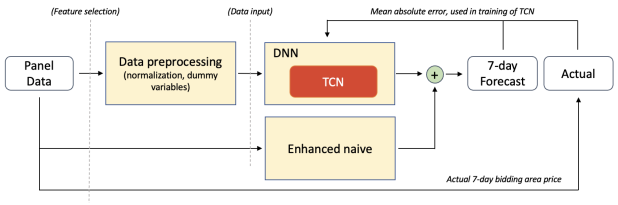


Figure 21: Summary of the ENTCN model. The DNN block (referred to as the TCN model) and the Enhanced naive model are used to make a point forecast of the system price 14 days forward in time. The TCN is optimised to forecast the error of the Enhanced naive, to improve the forecasts.

Equation 23.

$$\hat{y}_{t,r} = \frac{1}{24}\left(\sum_{i=k-24}^{k} y_i\right) * M_{t,r}^{t-k} * \frac{W_{t,r}}{W_{k,r}} * H_{t,r} * \frac{Ho_{t,r}}{Ho_{k,r}} \quad (23)$$

Here, k is the last hour of the input sequence, $M_{h,r}$, $W_{h,r}$, $H_{h,r}$, and $Ho_{h,r}$ are the monthly, weekday, hourly, and holiday effects at hour h in region r. The coefficients used for each region's effects were calculated using the relative averages in the in-sample dataset (from 2014 to 2019). The reason for dividing by the weekday and holiday effect of the last hour of the input sequence is that the average price of this day is used as the baseline for the forecast. Hence one needs to account for the price effects of this day. However, this is not necessary when looking at the hourly coefficients as there are no hourly effects when looking at a daily average. Furthermore, the monthly coefficient is calculated as the average daily price movement in a given month, looking at the relative change expected in a month (e.g., falling prices during the spring or rising prices during the autumn). The monthly coefficient $M_{m,r}$ for month m in region r is given by Equation 24.

$$M_{m,r} = \left(\frac{M_{m+1,r}}{M_{m-1,r}}\right)^{\frac{1}{24*60}} \quad (24)$$

The forecasted price of the Enhanced naive is used as a baseline for the TCN model. When training the TCN, the y values are the actual price minus the forecast of the enhanced naive, such that the TCN is trained on forecasting the error of the Enhanced naive. Consequently, when making forecasts out-of-sample, the forecasted value is the sum of the TCN and the Enhanced naive forecasts. The motivation for accounting for these effects was the significant impact they had on bidding area prices in the in-sample dataset, as discussed in Section 4.2.

The TCN model, which consists of a feed-forward deep neural network with a TCN layer followed by two hidden layers, was implemented using the widely used Tensorflow Keras architecture (Fu and Aldrich 2018). A summary of the TCN component can be seen in Figure 22 . When developing the TCN layer, code from T. R. Wang et al. (2021) and the open code TCN layer developed by Remy (2020) are used as a basis. These implementations were based on the TCN concepts presented in Bai et al. (2018), which were discussed and explained in Section 2.1.4. Notably, the padding used is causal, as to prevent any passing of data from future to past or present, which is required when doing time series forecasts. The hyperparameters used were also inspired by the ranges recommended by

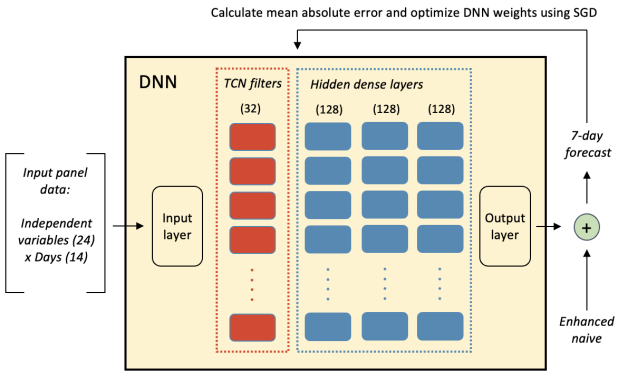


Figure 22: Illustration of the TCN model with, which consists of a TCN layer with 32 filters and three fully connected hidden layers with 128 nodes each.

Remy (2020). Furthermore, the hyperparameters were subsequently tweaked through testing several standard configurations for each parameter, using the validation loss in the NO1 and NO3 regions as a goodness of fit metric. A summary of the hyperparameters used in the TCN model is presented in Table 8 and Table 14.

Table 14: Hyperparamters used for the TCN model. Optimizer, loss function, learing rate, batch size, epochs, and validation split are listed in Table 8

Hyperparameter	Value
Input length	2 weeks (336 hours)
TCN filters	32
TCN activation	ReLU
TCN kernel size	32
TCN dialations	(1,2,4,8,16,32)
Dropout	No
# nodes, hidden layer 1	128
# nodes, hidden layer 2	128
# nodes, hidden layer 3	128
hidden layer activation	ReLU
last layer activation	Sigmoid

As seen in Table 14, the TCN layer consisted of 32 separate filters with kernel size equal to 32. Hence every TCN layer could perform pattern recognition over a 32-hour horizon. The dilutions used were 1, 2, 4, 8, 16, and 32, all factors of two, as recommended by Bai et al. (2018). Furthermore, no dropout was used. Although it often being recommended in order to avoid overfitting (Goodfellow et al. 2016), the use of dropout had little to no impact on validation loss. Finally, ReLU ($f(x) = max(0, x)$) was used as activation function in the TCN layer. However, the TCN layer only constitutes one layer in the DNN model, which took an input of 2 weeks (336) of temporal data. The final activation function used was sigmoid, widely used in DL forecasting (Goodfellow et al. 2016). As with the other deep learning models implemented, the neural network was optimised using stochastic gradient descent (SGD), explained in Section 2.1.2, using mean absolute error as the loss function. With implemented layers in the TCN model, there were a total of 475,752 trainable parameters, as seen in Table 15. Over 85% of the trainable parameters were in the TCN layer, which explains why the ENTCN model was considerably more computationally expensive than the vanilla DNN model implemented.

Table 15: Layers in TCN model, including the TCN layer, three hidden layers, and the output layer. There are a total of 475,752 trainable parameters

Layer (type)	Output shape	# param
tcn (TCN)	(None, 32)	416,832
dense (Dense)	(None, 128)	4,224
dense_1 (Dense)	(None, 128)	16,512
dense_2 (Dense)	(None, 128)	16,512
dense_3 (Dense)	(None, 168)	21,672

5.1.6 Regression

Two regression models were implemented, linear regression (Lin reg) and quadratic regression (Quad reg), using the sklearn Python framework. Both models were fitted using ordinary least squares (OLS), which is explained in Section 2.1.5. For both linear and quadratic regression, seven different regression models were developed, one for forecasting the hourly price at the same hour of the day 1, 2, 3, 4, 5, 6, and 7 days ahead. Hence, the models used the last day of the input sequence as the basis for the price, then forecasted the price at a specific hour using the price at the same hour of the last day of the input sequence and the suitable regression model (based on the number of days ahead). Furthermore, each region had its own set of regression models. The equation for the forecast ($\hat{y}_{t,r}$) of the regression models at time t in region r is given in Equation 25.

$$\hat{y}_{t,r} = R_{i,r}(x_k) \tag{25}$$

Here, $R_{i,r}$ is the regression model i days ahead in time in region r, while x_k is the independent variables at the same hour as t on the day before the forecasting period. As discussed in Section 2.1.5, the regression model $R_{i,r}$ can be expressed using Equation 26.

$$R_{i,r} = a_r + b_{1,r}x_1 + b_{2,r}x_2 + ... + b_{k,r}x_k \tag{26}$$

Here, a_r is the intercept in region r, while $b_{i,r}$ is the coefficient for independent variable i in region r. This formula holds for both the linear and quadratic regressions. However, the x values of the quadratic regression also include all independent variables multiplied with each other ($x_j * x_i$), as well as the quadratic value of each independent variable ($(x_i)^2$). Hence, when using quadratic regression, the number of independent variables is equal to $|x^2| + |x|$, which is the reason why a lower number of independent variables are included in the quadratic regression, as to avoid overfitting (Goodfellow et al. 2016).

5.1.7 The ARIMA and SARIMA Models

As a part of our statistical models, both SARIMA and ARIMA models were implemented. These models are described in Section 2.1.6. The programming language used was Python, with the statsmodels library (Seabold and Perktold 2010). This SARIMA- and ARMA-implementation is build upon the description in Durbin (2012).

Table 16: Hyperparamters used for the ARIMA and SARIMA models. * 0 by definition for the ARIMA model

Hyperparameter	Value
Estimator	AIC
Trend autoregression order (p)	2
Trend difference order (d)	1
Trend moving average order (q)	3
Seasonal autoregressive order (P)*	1
Seasonal difference order (D)*	0
Seasonal moving average order (Q)*	1
Steps (hours) per period (m)*	24

In order to find the optimal orders for the models, we conducted a grid search. A grid search means testing all combinations of variables in a given range. The information criteria chosen to evaluate the models was AIC, which is described in Section 2.1.6. The time complexity of the SARIMA models is $O(m^3 * N_{tr} * N_{te}/R)$, where m is the order of the model, N_{tr} is the number of training examples, N_{te} it the number of test examples and R is the recalibration rate (Lin et al. 2014). Consequently, it took on average 6270s to run a single model. The run time was something that we found to be exceedingly resource-consuming during the grid search. We had to be exceedingly deliberate when choosing ranges for the different variables in the grid search. A too wide search range would have resulted in a grid search that was overly resource-consuming, whereas a narrow grid search would not necessarily enable us to find the optimal hyperparameters. As a reference for ranges for the different variables with which to conduct the grid search, Engebretsen et al. (2021) was used as a basis. Additionally, we used the knowledge we obtained from our data analysis in Section 4.1. The grid search was conducted in NO3. However, the SARIMA and ARIMA models were fitted to each NordPool region individually and refitted between every forecast. A summary of the hyperparameters used in the ARIMA and SARIMA models is presented in Table 16.

5.2 Data Preprocessing and Selection

This subsection describes the data preprocessing and selection methods used in our experiments. Prior to being used as input, the data was preprocessed. The preprocessing that we have conducted can be divided into two stages. Firstly, we generated categorical variables, and secondly, the data was standardized.

The first step of the preprocessing phase was to create *dummy variables* from the categorical variables. Categorical variables are variables where there is no numerical relationship between the variables and can be helpful when transformed to dummy variables (Goodfellow et al. 2016). The calendar effect variables were categorical, and the hours, weeks, weekdays, holidays, months, and seasons were transformed into dummy variables. For a given set of n different categories for a variable, there is a need for $n-1$ new dummy variables (Brooks 2019; Goodfellow et al. 2016). A 1 represents an element belonging to a specific category, whereas a 0 implies the absence from a category. For granular categories, such as weeks, the result is a substantial amount of new variables for the data, 51 to be specific. As a result, we found it more beneficial to use less granular groups of dummy variables for calendar effects,

which can be observed in Table 17.

The second step of the preprocessing phase was *standardization*. The input data, except for the calendar-effect variables, were standardized. Goodfellow et al. (2016) highlight that adequate numerical transformations are important to enable the neural networks to learn better. There are numerous manners in which preprocessing can be conducted. We have chosen standardization, which is described in Equation 27. x_i is the data point at time t, μ is the mean, and σ is the standard deviation. Consequently, $x_{standardize} \sim \mathcal{N}(0, 1)$. Min-max normalization, principal component analysis, unit vector transformation, and $asinh$-transformation were also considered and implemented in code. Nevertheless, we used the standardization as preprocessing technique rather than the other mentioned techniques as it resulted in lower validation loss.

$$x_{standardized} = \frac{x_i - \mu}{\sigma} \tag{27}$$

The independent variables used by each model are summarised in Table 17. The process for finding the variables for the different models was conducted in the following manner. Firstly, a linear regression model was used to find the variables that showed statistical significance regarding forecasting. The variables that showed significance regarding forecasting gave us a subset of variables. Each model was validated with this variable set, and validation loss for NO1 and NO3 was calculated. Here, we removed variables individually to see which variables could be removed without creating noticeable jumps in the validation loss. This was done to prevent overfitting and spurious relationships between the input and output variables. Consequently, we believe that each model has an adequate set of input variables for forecasting the electricity price for the next seven days (168 hrs.). Naturally, the lagged values for the areas the model is forecasting are included in all models. The inclusion of this variable as input is also highly in line with the auto-correlation-plots analysed in Section 4.1. Furthermore, we also found great explanatory value for the deep learning models by including calendar effect variables, although this variable group was not as important for the linear or quadratic regression. It is noticeable that the weather and production variables proved to be of explanatory value for all model groups, except the univariate, which does not have them as input, and the quadratic regression model. The RNNs and linear regression found the market data valuable, but none of the other models used market data to this extent. The last group is the commodity variables. These were used by the RNNs, ENTCN, and linear regression models. A potential reason for their explanatory value is that the commodiy prices function as price setters for the electricity price on continental Europe, as described in Section 4.2.

5.3 Error Metrics

The performance metrics in this thesis are; *mean absolute error* (MAE), *root-mean-square error* (RMSE), *mean average percentage error* (MAPE), and *symmetric mean average percentage error* (SMAPE). The metrics are summarised in Table 18. These are all standard error metrics referenced in other prominent papers within electricity price forecasting, such as Lago, Marcjasz et al. (2021) and Weron (2014).

The error metrics most widely used in electricity price forecasting are mean absolute error (MAE), root-mean-square error (RMSE), and mean absolute percentage error (MAPE) (Lago, Marcjasz et al. 2021). MAE provides absolute errors, with symmetrical punishments of too high and too low forecasts. This can be relevant for an actor experiencing a linear loss/ gain of differences in electricity prices. However, some actors experience significant errors disproportionally more painful than small ones. In these cases, RMSE is a more relevant error metric since it increases the magnitude of the error quadratically dependent on the actual size of the error. RMSE, as with MAE, symmetrically punishes forecasting errors. A weakness with absolute errors is that they are not comparable across periods or datasets of different scales (Lago, Marcjasz et al. 2021), a solution being using relative errors.

Relative error metrics include mean average percentage error (MAPE) and symmetric mean average percentage error (SMAPE), which were both used by Engebretsen et al. (2021). Proportional error measurement can be applied for the prediction of volatile series, of which electricity prices are one (Goto and Karolyi 2004). However, as regional electricity prices can be close to zero or even negative, these metrics are unsuited as one could get errors approaching infinity. Forecasting errors approaching infinity for some forecasts will strongly disrupt the measurement and provide little additional value. To account for this, instances in which the actual price is < 1 €/MWh (representing only 0.2% of cases across regions) are ignored.

It is insightful to provide several metrics to provide a more holistic picture of model performance. Furthermore, every metric tells a different story and might be more or less relevant for different actors. Hence, including all is relevant when doing a systematic benchmarking.

5.4 Statistical Testing

To assess if differences in performance across models are statistically significant, two primary forms of statistical tests exist; the *Diebold-Mariano* (DM) and *Giacomini-White* (GW) tests. Although the importance of such tests has been downplayed in electricity price forecasting literature, most papers only presenting error metrics (Weron 2014), Lago, Marcjasz et al. (2021) highlight the importance in order to ensure statistical rigorous model comparisons.

The Diebold-Mariano (DM) test is one of the most used statistical tests within electricity price forecasting, which compares forecasts of models (Lago, Marcjasz et al. 2021). It is an asymptotic z-test, in which the hypothesis is that the mean of the loss differential series given in Equation 28 is equal to 0 (Diebold and Mariano 1995).

$$\Delta_t^{A,B} = L(\varepsilon_t^A) - L(\varepsilon_t^B) \tag{28}$$

$$\varepsilon_t^A = p_t - \hat{p_t} \tag{29}$$

$$L(\varepsilon_t^A) = |\varepsilon_t^A|^p \quad p = 1 \text{ or } 2 \tag{30}$$

Here, L is the loss function (often absolute or squared loss), while $\varepsilon_{d,h}^A$ is the prediction error for model A at time t. In

Table 17: Summary of the models implemented and the time series data types used by each. * Includes ARIMA, SARIMA, and Naive. ** Includes LSTM, Stacked LSTM, GRU, and Stacked GRU. *** Dummy variables

Data	Univariate*	DNN	RNN**	ENTCN	Lin reg	Quad reg
NordPool prices	**x**	**x**	**x**	**x**	**x**	**x**
Current bidding area price	x	x	x	x	x	x
Other bidding area prices		x	x	x	x	x
System Price			x	x	x	
Time variables		**x**	**x**	**x**	**x**	**x**
Hour***		x	x	x		
Weekday***		x	x	x	x	x
Month***				x		
Holiday		x	x	x	x	x
Commodities			**x**	**x**	**x**	
Oil			x	x	x	
Gas			x	x	x	
Coal			x	x	x	
Market data		**x**	**x**	**x**	**x**	
Total NordPool Vol		x	x	x	x	
APX price			x		x	
OMEL price			x		x	
EEX price			x		x	
Production data		**x**	**x**	**x**	**x**	
Hydro production		x	x	x	x	
Wind production		x	x	x	x	
Weather data		**x**	**x**	**x**	**x**	
Temp Norway		x	x	x	x	
Prec Norway 7 days		x	x	x	x	

Table 18: Summary of the error metrics, F_t and A_t are the forecasted and actual values at time t, n being the number of predictions. * ignores instances in which the actual price is $<$ 1 €/MWh (0.2% of cases), as these can lead to extremely high errors, or inf if the price approaches zero or negative values

Error metric	Formula
MAE	$\frac{1}{n}\sum_{n=0}^{n}\lvert F_t - A_t\rvert$
RMSE	$\sqrt{\frac{1}{n}\sum_{n=0}^{n}\lvert A_t - F_t\rvert^2}$
MAPE*	$\frac{100\%}{n}\sum_{n=0}^{n}\left\lvert\frac{A_t - F_t}{A_t}\right\rvert$
SMAPE*	$\frac{100\%}{n}\sum_{n=0}^{n}\frac{\lvert F_t - A_t\rvert}{(\lvert A_t\rvert + \lvert F_t\rvert)/2}$

this thesis, $p = 1$ is used, making the loss function equal to the mean average error. We can then calculate the Diebold-Mariano (DM) test statistic, given in equation Equation 31.

$$DM = \sqrt{N}\frac{\hat{\mu}}{\hat{\sigma}} \tag{31}$$

$\hat{\mu}$ and $\hat{\sigma}$ are the sample mean and standard deviation of $\Delta_t^{A,B}$, while N is the number of samples. These can easily be calculated if assuming covariance stationarity of $\Delta_t^{A,B}$ (Diebold and Mariano 1995). It can then be compared to the Z test statistic, following the standard normal distribution N(0,1). One can then calculate the p-value of the one-sided with the null hypothesis $H_0 : E(\Delta_t^{A,B}) \leq 0$ (or a two-sided test in which $E(\Delta_t^{A,B}) \neq 0$). Here one is trying to prove the alternative hypothesis that model A has a higher loss than model B (or they have a different loss). The lower the p-value, the more inconsistent the null hypothesis is with the observed data, one typically rejecting it at a value lower than 5% (Diebold and Mariano 1995).

An alternative to the DM test is the Giacomini-White (GW) test (Giacomini and White 2006). The GW test has replaced the DM test in some recent electricity price forecasting papers, such as in Marcjasz, Lago et al. (2020). It can be regarded as a generalization for unconditional predictive ability, with only the GW test accounting for parameter estimation uncertainty through conditioning (Lago, Marcjasz et al. 2021). Given the scope of this thesis, only the Diebold-Mariano test is implemented. However, further work might include the implementation of the alternative Giacomini-White test.

5.5 Experimental Design

Before training or fitting the implemented models, the data set was separated into two non-overlapping data sets, in-sample and out-of-sample, to ensure that the models were trained or fitted solely on the out-of-sample data. The in-sample data set consists of data from *1 January 2014 to 31 December 2019*, while the out-of-sample data set is from *1 January 2020 to 31 December 2020*. The reason for not using data from 2021 is the abnormally high prices exhibited in the NordPool market, which is not remotely reflected in the in-sample data, which might give a less meaningful indication of model performances. Furthermore, a testing environment where each model is required to make a seven-day hourly (168 points) forecast for the 12 NordPool bidding regions in Norway, Sweden, Denmark, and Finland is created. Hence, the models are trained on 2184 seven-day training cases while being tested on 339 separate seven-day test cases. As some models use up to 21 days of data as input, the first forecasted period is 00:00 22.01.2020 - 24:00 28.01.2020, with the last being 00:00 25.12.2020 - 24:00 31.12.2020. With 339 separate seven-day test cases, 56,952 distinct hourly prices are forecasted for each test run on a specific bidding area. Hence, with such a large number of forecasted data points for all 12 bidding regions, there is enough data to get statistically significant model

performance results. However, as every training and test example starts at 00:00 for a specific day, there are overlaps between different periods. For example, for two test periods with 1-day in between, there is a six-day overlap. Overlaps in test periods are not a problem but something to be conscious of as model errors for similar periods might correlate. Finally, the models are assessed across the four error metrics discussed in Section 5.3 and compared using the Diebold-Mariano test discussed in Section 5.4. In addition, the computation times of the implemented models are also compared.

6 Results

This section presents the error metrics and statistical tests (DM) of the models on the 12 NordPool bidding areas. Furthermore, the average model performances in different countries and the NordPool area is presented. Lastly, a deep dive on several high-performing models with example forecasts from the NO3 (Trondheim) region is provided.

6.1 Bidding Areas

A summary of the error metrics of the implemented models across all 12 NordPool bidding areas on the out-of-sample data is presented in Table 20. Furthermore, the highest performing model across the four error metrics in each bidding area is presented in Table 19. Looking at the results, several observations can be made:

Table 19: Best performing model across bidding regions and error metric. ARIMA is the best performing across all Norwegian regions accross all four metrics. In Sweden and Finland linear regression is the best performing, while SARIMA is the highest performing model in the Danish regions.

Area	MAE	SMAPE	RMSE	MAPE
NO1	ARIMA	ARIMA	ARIMA	ARIMA
NO2	ARIMA	ARIMA	ARIMA	ARIMA
NO3	ARIMA	ARIMA	ARIMA	ARIMA
NO4	ARIMA	ARIMA	ARIMA	ARIMA
NO5	ARIMA	ARIMA	ARIMA	ARIMA
SE1	ARIMA	ARIMA	ARIMA	SARIMA
SE2	ARIMA	ARIMA	ARIMA	SARIMA
SE3	Lin reg	Lin reg	Lin reg	Lin reg
SE4	Lin reg	Lin reg	Lin reg	Lin reg
DK1	SARIMA	SARIMA	SARIMA	Lin reg
DK2	SARIMA	SARIMA	SARIMA	Lin reg
FI	Lin reg	Lin reg	Lin reg	Naive 7d

- Across all the Norwegian bidding areas (NO1-NO5), ARIMA is the highest performing model across all error metrics. At the same time, the SARIMA and ENTCN models exhibit good performances across all error metrics.
- In SE1 and SE2, which have almost identical prices, the ARIMA and SARIMA models are the highest performing. However, in SE3 and SE4, linear regression is the highest performing across all error metrics, followed by the seven-day naive forecast.
- In the Danish regions (DK1 and DK2), the SARIMA is the highest performing model across all error metrics except MAPE (in which linear regression is the highest performing).
- In the Finish bidding area (FI), the seven-day naive forecast and the linear regression models are the highest performing.
- Across all 12 bidding regions, the deep learning[3] models implemented (DNN, LSTM, S-LSTM, GRU and S-GRU) perform poorly across all four error metrics. The LSTM model consistently performed slightly worse across most bidding areas.
- In most cases, the highest performing model across the absolute errors (MAE and RMSE) in a specific bidding region, is in almost all cases[4] also the highest performing model across the relative errors (SMAPE and MAPE)
- Although there are some regional differences, a number of models performed well across most bidding areas across most metrics; these incl. ARIMA, SARIMA, ENTCN, linear regression and the seven-day naive benchmark.

6.1.1 The Diebold-Mariano Test Across Bidding Areas

In order to compare the performance of the different models, the Diebold-Mariano (DM) test (as explained in Section 5.4) is used. The significance levels of the two-sided DM test for each bidding area are presented in Figure 23. In the figure, the blue values represent statistically significant better performance of the model on the x-axis vs the model on the y-axis 5%. Additionally, the models are sorted based on MAE in the specific region in the figure. As can be seen, almost all differences in performance between models are statistically significant. Some notable exceptions include S-GRU, S-LSTM and DNN in NO1 and ARIMA, SARIMA and Quad reg in DK2. The high level of statistical significance might be due to the very high number of test examples. Furthermore, there are also model errors which might also correlate due to overlapping test periods, somewhat invalidating the DM test assumptions that the errors follow a normal distribution (Diebold and Mariano 1995). However, the DM test still gives us a good indication of differences in model performance.

6.1.2 Computation Time

The time usage of the implemented models is summarised in Table 21. The most time-consuming model was the SARIMA, in part as this needed to be refitted for each forecast. The deep learning models were somewhat time consuming as the Tensorflow architecture needed to be initialised. Furthermore, the naive forecast and the regression models required little computation time, as limited computation was necessary. So although the SARIMA exhibited good predictive performance, it was approximately ten times slower than the second slowest model.

6.2 Countries and NordPool

To further assess the model performances, this subsection investigates the average bidding area performance across the different countries and the NordPool region. A summary of the avg. model error metrics is presented in Table 23. Furthermore, the highest performing model by

[3] Not including the ENTCN model as it is also a hybrid model
[4] Except for MAPE in DK1, DK2, and FI

Table 20: Summary of the model error metrics across NordPool bidding regions, (**bold** = lowest).

		Naive 7d	ARIMA	SARIMA	DNN	ENTCN	LSTM	S-LSTM	GRU	S-GRU	Lin reg	Quad reg
NO1	MAE	3,11	**2,21**	2,30	7,32	2,42	7,92	7,23	6,96	7,07	4,14	6,96
	SMAPE	38,55	**27,44**	34,45	144,57	31,47	186,10	128,32	125,98	106,80	57,74	71,76
	RMSE	3,97	**2,90**	2,97	7,72	3,07	8,26	8,04	7,40	8,54	4,93	12,30
	MAPE	48,60	**33,16**	36,26	79,49	40,54	97,20	104,26	74,89	163,63	128,57	205,67
NO2	MAE	3,12	**2,22**	2,28	7,32	2,41	7,92	7,22	6,95	7,06	3,87	4,88
	SMAPE	38,52	**27,34**	33,82	145,04	31,17	186,26	129,26	126,19	106,38	56,14	68,11
	RMSE	3,97	**2,90**	2,94	7,72	3,07	8,26	7,99	7,39	8,52	4,61	7,42
	MAPE	48,48	**33,13**	35,66	79,79	40,37	97,24	100,61	74,88	161,09	119,65	139,28
NO3	MAE	3,20	**2,46**	3,03	7,63	2,60	8,23	7,26	7,18	6,75	5,97	7,77
	SMAPE	40,79	**31,67**	39,34	155,09	32,73	189,57	128,93	134,86	99,57	62,15	61,29
	RMSE	3,83	**3,00**	3,53	8,00	3,09	8,55	8,00	7,60	8,14	6,83	13,44
	MAPE	50,44	**39,22**	58,61	85,22	43,43	98,01	88,04	77,74	117,75	136,23	168,70
NO4	MAE	2,64	**2,03**	2,18	7,02	2,18	7,63	6,66	6,56	6,33	4,04	6,95
	SMAPE	37,99	**28,98**	34,96	153,92	30,41	189,28	127,25	131,93	97,53	47,74	64,65
	RMSE	3,20	**2,48**	2,62	7,33	2,59	7,89	7,36	6,91	7,71	4,82	12,42
	MAPE	48,80	**35,84**	37,25	84,82	40,87	97,95	87,76	76,64	120,03	89,84	149,02
NO5	MAE	2,96	**2,06**	2,10	7,19	2,23	7,79	7,12	6,83	7,02	3,65	4,55
	SMAPE	38,15	**26,8**	33,49	144,09	30,63	185,95	129,28	125,45	107,45	55,27	65,30
	RMSE	3,55	**2,56**	2,61	7,50	2,72	8,04	7,79	7,17	8,40	4,25	6,95
	MAPE	48,03	**32,37**	34,61	79,27	39,66	97,17	101,82	74,67	167,30	116,48	133,51
SE1	MAE	6,50	**5,55**	5,65	12,96	5,94	13,57	12,55	12,51	10,62	6,69	9,93
	SMAPE	48,43	**40,98**	45,91	168,29	42,36	192,96	147,48	152,69	105,57	49,52	58,65
	RMSE	9,05	**7,44**	7,46	14,47	7,69	15,01	14,29	14,05	12,94	8,60	16,47
	MAPE	66,17	60,09	**59,94**	90,15	63,42	98,69	88,53	84,92	91,00	89,96	130,48
SE2	MAE	6,50	**5,55**	5,65	12,96	5,95	13,57	12,55	12,51	10,62	6,69	9,93
	SMAPE	48,43	**40,98**	45,91	168,29	42,33	192,96	147,48	152,69	105,57	49,54	58,48
	RMSE	9,05	**7,44**	7,46	14,47	7,69	15,01	14,29	14,05	12,94	8,61	16,48
	MAPE	66,17	60,09	**59,93**	90,15	63,58	98,69	88,53	84,92	91,00	90,05	130,58
SE3	MAE	12,44	13,07	13,38	20,25	13,03	20,86	19,83	19,79	17,41	**10,48**	14,66
	SMAPE	59,67	61,91	69,05	173,41	61,69	194,19	155,87	161,18	116,71	**53,46**	67,08
	RMSE	17,55	16,99	17,07	25,01	16,69	25,48	24,73	24,59	22,56	**14,21**	22,67
	MAPE	103,63	116,06	120,53	91,73	116,32	98,93	90,76	87,81	91,69	**81,36**	109,63
SE4	MAE	13,33	14,70	14,36	24,96	14,50	25,57	24,52	24,50	21,69	**12,76**	17,50
	SMAPE	58,47	62,51	67,00	177,80	61,82	195,18	162,78	167,68	124,87	**57,25**	71,41
	RMSE	18,10	18,46	17,87	29,77	18,13	30,26	29,43	29,34	26,87	**16,3**	25,29
	MAPE	109,66	135,35	130,44	93,11	130,86	99,12	92,64	90,03	91,31	**81,24**	103,18
DK1	MAE	13,24	12,97	**11,73**	24,54	14,04	25,10	24,21	24,18	21,50	12,48	14,87
	SMAPE	62,50	61,03	**50,08**	180,52	60,48	195,80	168,27	172,20	130,31	57,05	64,65
	RMSE	16,81	16,38	**14,4**	28,37	17,11	28,87	28,06	28,01	25,54	15,08	19,36
	MAPE	130,95	101,96	117,49	94,01	119,91	99,22	93,98	91,69	89,59	**82,11**	97,87
DK2	MAE	13,41	14,22	**11,91**	27,76	14,75	28,34	27,54	27,35	23,96	13,15	17,15
	SMAPE	54,11	58,96	**45,62**	183,16	56,26	196,39	173,93	175,14	129,53	53,69	60,53
	RMSE	18,00	18,42	**15,39**	32,06	18,31	32,57	31,87	31,66	28,75	16,60	24,52
	MAPE	103,28	90,92	102,09	94,96	112,28	99,35	94,92	92,65	87,01	**69,67**	88,14
FI	MAE	12,85	16,80	12,78	27,76	15,64	28,00	26,67	26,92	23,77	**12,09**	16,04
	SMAPE	52,87	65,82	50,90	183,16	61,10	195,81	160,54	171,11	127,25	**49,29**	52,59
	RMSE	17,92	21,23	16,69	32,06	19,65	33,49	32,48	32,55	29,86	**15,79**	23,59
	MAPE	**82,38**	129,86	102,06	94,96	120,96	99,25	92,12	91,26	87,81	89,05	111,15

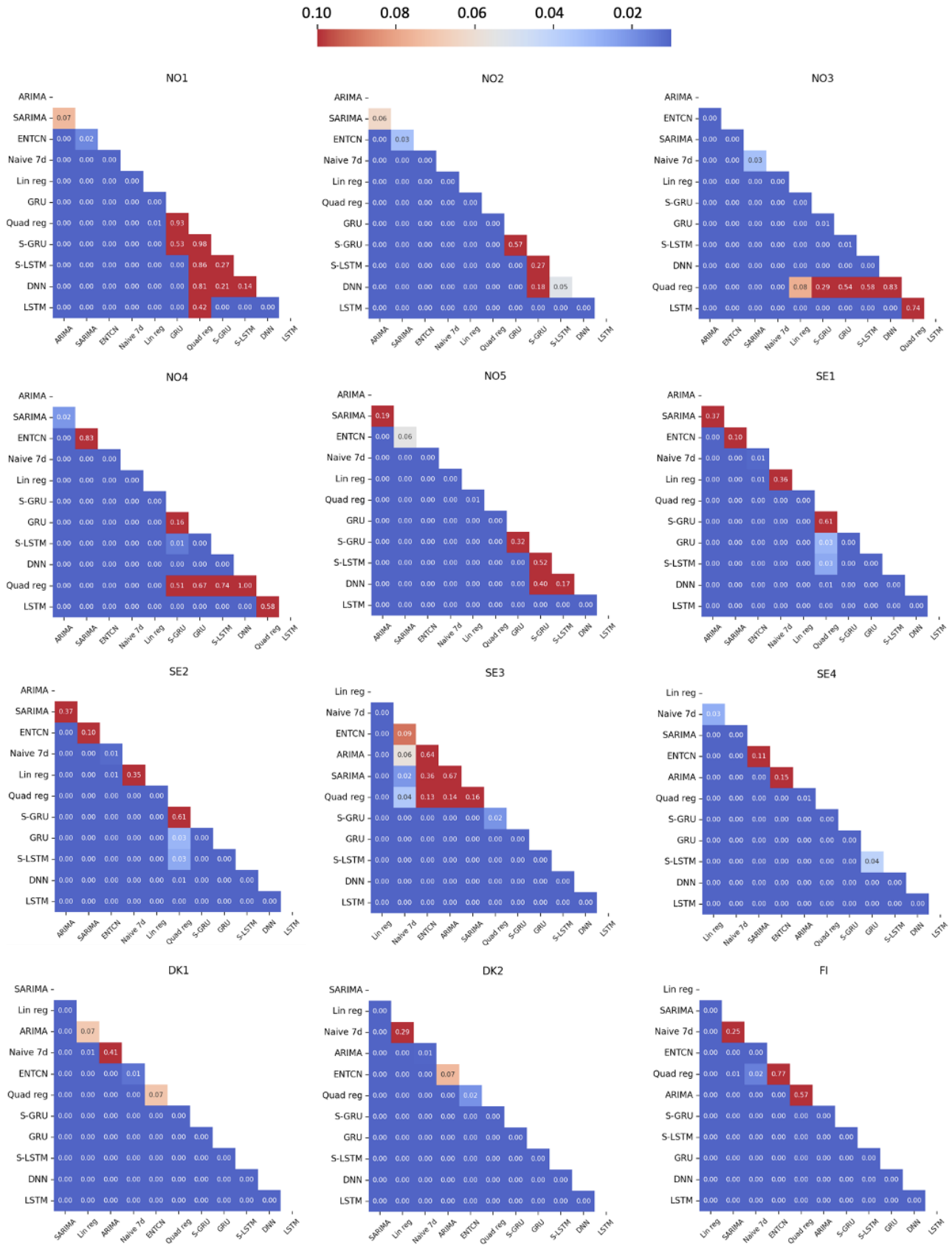


Figure 23: Significance level (p) of the two-sided DM test for each bidding area, with blue values showing a statistically significant better performance of the model on x-axis vs. the model on y-axis. The models sorted based on MAE model performance.

Table 21: The computation time (running on the Apple M1-processor) of fitting/ training and running the implemented models over the whole out-of-sample dataset on a single bidding area.

Model	Computation time
Naive 7d	$\leq$ 1 sec
ARIMA	2-3 min
SARIMA	50-70 min
DNN	50-80 sec
ENTCN	3-5 min
LSTM	3-4 min
S-LSTM	5-7 min
GRU	2-3 min
S-GRU	3-4 min
Lin reg	3-5 sec
Quad reg	6-8 sec

country for each error metric is presented in Table 22. Looking at the results, a number of observations can be made:

Table 22: Best performing model over countries and NordPool (avg. of bidding regions) across the four error metrics. SARIMA and ARIMA are the best performing models on avg. across all NordPool bidding areas, with linear regression performing well in both Sweden and Finland.

Region	MAE	SMAPE	RMSE	MAPE
NordPool	SARIMA	ARIMA	SARIMA	ARIMA
Norway	ARIMA	ARIMA	ARIMA	ARIMA
Sweden	Lin reg	ARIMA	Lin reg	Lin reg
Denmark	SARIMA	SARIMA	SARIMA	Lin reg
Finland	Lin reg	Lin reg	Lin reg	Naive 7d

- On average, across all NordPool bidding regions, SARIMA is the highest performing across the absolute errors (MAE and RMSE). At the same time, ARIMA is the highest performing across relative errors (MAPE and SMAPE).
- There are big differences in model performances between countries. ARIMA is the highest performing in Norway, while SARIMA is the highest performing in Denmark. However, linear regression is generally the highest performing model in Sweden and Finland.
- As seen with the bidding areas, almost all models perform significantly better in Norway than in Sweden, Denmark and Finland across all four error metrics. This is partly due to lower prices and less electricity price volatility in the Norwegian region, partly due to a more readily available hydropower generation.

6.2.1 The Diebold-Mariano Test Across Countries

As with the individual bidding areas, the performance of the different models across the countries is compared using the Diebold-Mariano (DM) test (as explained in Section 5.4). The significance levels of the two-sided DM test for each bidding area are presented in Figure 24. The blue values represent statistically significant better performance of the model on the x-axis vs the model on the y-axis 5%. As can be seen, almost all differences in performance between models across all countries are statistically significant. Notable exceptions include the SARIMA and the ENTCN in Norway, ARIMA and seven-day naive in NordPool, and ENTCN and Lin reg in NordPool.

6.3 Comparison of High Performing Models

As discussed in Section 6.1, and shown in Table 20, a number of models performed consistently well across all bidding areas and error metrics. These models are further analysed in this subsection and include; ARIMA, SARIMA, ENTCN, linear regression and the seven-day naive benchmark.

Looking at the descriptive summary of MAE (incl. mean, median, std, min, and max) provided in Table 24, one can make several further inferences about model performance:

- Across most bidding areas, the ENTCN model exhibits the highest maximum MAE and standard deviation, partly due to making forecasts more deviating from the current price. This is due to the TCN component trying to model complex relationships in the data, while the simple multivariate models to a higher degree bases their forecast on the most recent price points.
- Across all models and bidding areas, the median MAE is consistently lower than the mean. This is a symptom of a small number of large absolute errors driving up the average value.
- For all models, the standard deviation is the highest in the SE3 region, with the mean MAE also being relatively high. This stands in contrast to the Norwegian bidding areas in which the models exhibit low standard deviations, also having low mean MAE
- The linear regression model is consistently outperformed by the other high performing models in the Norwegian bidding areas but still has low maximum MAEs. However, the model also exhibits high minimum MAEs.
- In Southern Norway, NO1(Oslo), NO2 (Kr.sand) and NO5 (Bergen), the ARIMA and SARIMA models exhibit extremely low minimum MAEs (≤ 0.44). This is due to extremely low and consistent price levels in the Southern region over a one week period. On the flip side, the minimum MAE values in Finland are never below 4.5 for any of the models, indicating that neither of them does a good job of forecasting the price dynamics in the country. This can be a symptom of the models being fitted/ trained using the validation error in the NO1 and NO3 regions as a goodness of fit indicator. While these regions might have similar dynamics as other NO regions and even SE and DK regions, they are far apart from the FI region. These large regional price differences between regions can be seen in Figure 35.

Table 23: Summary of the avg. bidding area model error metrics by country and across NordPool, (**bold** = lowest).

		Naive 7d	ARIMA	SARIMA	DNN	ENTCN	LSTM	S-LSTM	GRU	S-GRU	Lin reg	Quad reg
NordPool	MAE	7,77	7,82	**7,28**	15,64	7,97	16,21	15,28	15,19	13,65	8,00	10,93
	SMAPE	48,21	**44,53**	45,88	164,78	45,20	191,70	146,62	149,76	113,13	54,07	63,71
	RMSE	10,42	10,02	**9,25**	17,87	9,99	18,47	17,86	17,56	16,73	10,05	16,74
	MAPE	75,55	**72,34**	74,57	88,14	77,68	98,40	93,67	83,51	113,27	97,85	130,60
Norway	MAE	3,00	**2,19**	2,38	7,29	2,37	7,90	7,10	6,90	6,85	4,33	6,22
	SMAPE	38,80	**28,45**	35,21	148,54	31,28	187,43	128,61	128,88	103,55	55,81	66,22
	RMSE	3,70	**2,77**	2,93	7,65	2,91	8,20	7,84	7,29	8,26	5,09	10,51
	MAPE	48,87	**34,74**	40,48	81,72	40,97	97,51	96,50	75,76	145,96	118,15	159,24
Sweden	MAE	9,69	9,72	9,76	17,78	9,86	18,39	17,36	17,33	15,08	**9,16**	13,01
	SMAPE	53,75	**51,6**	56,97	171,94	52,05	193,82	153,40	158,56	113,18	52,44	63,90
	RMSE	13,44	12,58	12,46	20,93	12,55	21,44	20,68	20,51	18,83	**11,93**	20,23
	MAPE	86,40	92,90	92,71	91,28	93,54	98,86	90,12	86,92	91,25	**85,66**	118,47
Denmark	MAE	13,32	13,60	**11,82**	26,15	14,39	26,72	25,88	25,76	22,73	12,81	16,01
	SMAPE	58,30	59,99	**47,85**	181,84	58,37	196,10	171,10	173,67	129,92	55,37	62,59
	RMSE	17,41	17,40	**14,9**	30,21	17,71	30,72	29,97	29,84	27,15	15,84	21,94
	MAPE	117,11	96,44	109,79	94,48	116,09	99,29	94,45	92,17	88,30	**75,89**	93,00
Finland	MAE	12,85	16,80	12,78	27,76	15,64	28,00	26,67	26,92	23,77	**12,09**	16,04
	SMAPE	52,87	65,82	50,90	183,16	61,10	195,81	160,54	171,11	127,25	**49,29**	52,59
	RMSE	17,92	21,23	16,69	32,06	19,65	33,49	32,48	32,55	29,86	**15,79**	23,59
	MAPE	**82,38**	129,86	102,06	94,96	120,96	99,25	92,12	91,26	87,81	89,05	111,15

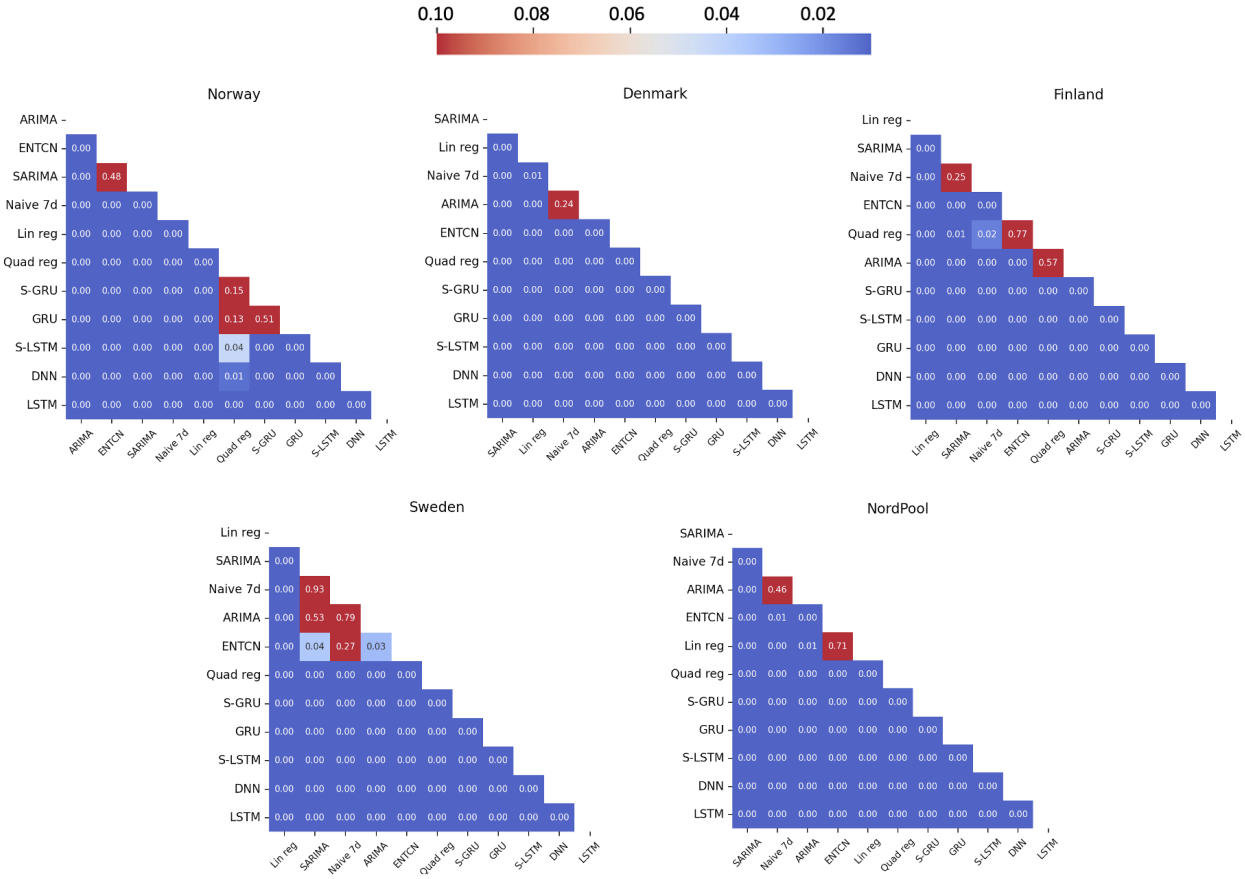


Figure 24: Significance level (p) of the two-sided DM test for each country and NordPool, with blue values showing a statistically significant better performance of model on x-axis vs. model on y-axis. Sorted based on MAE model performance.

Table 24: Descriptive summary of the MAE of high performing models across bidding areas

Area	Model	Mean	Median	Std	Min	Max
NO1	SARIMA	2.304	1.356	2.415	0.043	17.599
	ARIMA	2.213	1.441	2.217	0.022	11.576
	ENTCN	2.419	1.539	2.334	0.414	19.721
	Lin reg	4.139	3.959	1.698	0.794	11.844
NO2	SARIMA	2.28	1.323	2.396	0.044	17.636
	ARIMA	2.216	1.418	2.212	0.022	11.615
	ENTCN	2.411	1.502	2.321	0.413	19.096
	Lin reg	3.865	3.71	1.579	0.849	10.702
NO3	SARIMA	3.03	2.515	2.1	0.313	9.786
	ARIMA	2.459	1.945	1.997	0.259	19.021
	ENTCN	2.603	1.912	2.019	0.49	16.885
	Lin reg	5.973	5.584	2.689	1.075	16.057
NO4	SARIMA	2.18	1.63	1.793	0.238	9.671
	ARIMA	2.028	1.561	1.735	0.25	18.671
	ENTCN	2.179	1.657	1.771	0.456	17.446
	Lin reg	4.041	3.377	2.346	1.071	14.283
NO5	SARIMA	2.1	1.313	2.126	0.044	14.502
	ARIMA	2.056	1.438	2.015	0.022	9.455
	ENTCN	2.227	1.447	1.957	0.404	10.027
	Lin reg	3.653	3.586	1.469	0.853	8.49
SE1	SARIMA	5.649	4.425	4.415	0.438	32.388
	ARIMA	5.552	4.405	4.469	0.397	25.144
	ENTCN	5.944	4.598	5.521	0.671	42.451
	Lin reg	6.688	6.319	3.148	1.47	17.989
SE2	SARIMA	5.649	4.423	4.414	0.437	32.407
	ARIMA	5.552	4.405	4.468	0.397	25.117
	ENTCN	5.948	4.616	5.521	0.667	42.452
	Lin reg	6.691	6.324	3.149	1.472	17.986
SE3	SARIMA	13.375	10.558	9.451	2.045	55.93
	ARIMA	13.07	11.082	9.315	1.206	60.356
	ENTCN	13.03	10.984	9.282	1.564	60.836
	Lin reg	10.482	8.484	6.465	1.511	42.025
SE4	SARIMA	14.357	12.333	7.823	4.296	51.876
	ARIMA	14.704	13.316	6.902	3.392	39.527
	ENTCN	14.501	13.1	7.744	2.565	55.351
	Lin reg	12.763	11.127	6.332	3.267	41.593
DK1	SARIMA	11.727	10.122	4.922	3.472	37.747
	ARIMA	12.971	10.95	6.597	3.049	42.44
	ENTCN	14.035	12.151	7.17	3.343	39.833
	Lin reg	12.484	11.493	4.895	3.865	34.895
DK2	SARIMA	11.906	10.776	4.858	5.229	31.97
	ARIMA	14.219	12.344	7.16	3.547	46.369
	ENTCN	14.749	13.039	7.424	3.348	47.127
	Lin reg	13.145	11.566	5.635	3.761	39.584
FI	SARIMA	12.775	11.417	5.37	5.754	36.529
	ARIMA	16.795	15.099	7.581	5.49	68.725
	ENTCN	15.641	13.686	8.009	4.536	75.934
	Lin reg	12.085	10.864	4.756	5.463	33.429

6.3.1 NordPool

One can draw some interesting inferences by making a deep dive into the high performing models' average performance across the NordPool bidding regions. Looking at Figure 25, showing the avg. MAE by the hour of the day, one can see significant variations across the day. While forecasting errors are relatively low during the night for all models, they reach almost $2x$ as high values during the morning while also spiking during the evening hours. This also corresponds to the movement of the electricity price during the day. Looking at Figure 26, showing the avg. MAE as a function of the hour into the forecasting horizon, one can see that these daily fluctuations are highly apparent, regardless of model. Furthermore, one can see that the MAE increases further into the forecasting horizon for all models except the naive benchmark.

In Table 25 one can see the average forecasted price across all NordPool bidding areas and the share of forecasts over the actual. One can see that the naive benchmark, the ARIMA, and the SARIMA models forecast too high and too low approximately at the same rate. In contrast, the ENTCN and linear regression models forecast too high values 59% and 67% of hours, respectively. Interestingly, their average forecasts are still very similar to the actual, with the linear regression even forecasting too low price on average. One reason for this might be the price differences between regions, resulting in the model making substantial underestimates in specific regions while consistently over forecasting by a small amount in others. From Table 25, it is interesting to note that, on average, the forecast of the SARIMA model is the closest to that of the actual price.

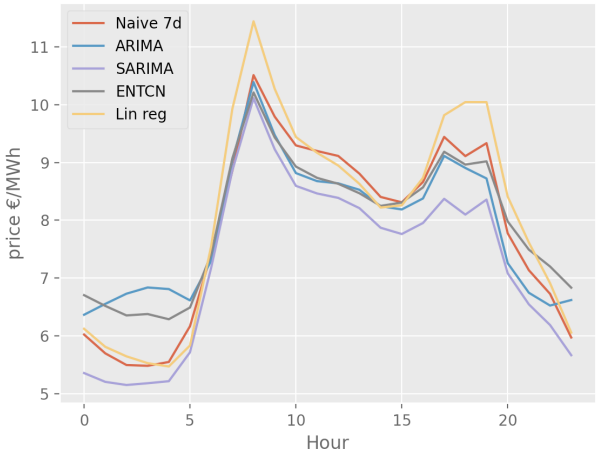


Figure 25: Average MAE across all NordPool bidding regions as a function of hour of the day. One can observe that the highest errors occur during the morning and afternoon

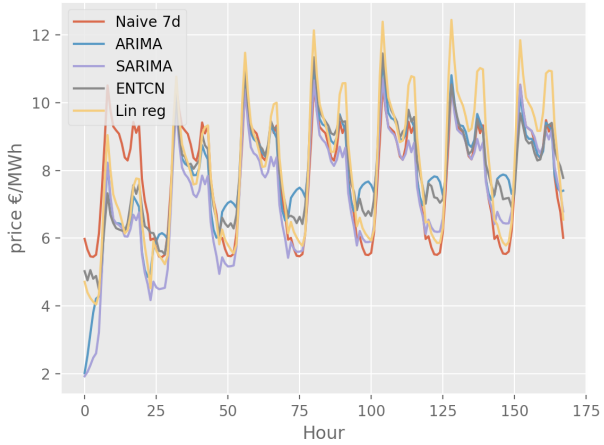


Figure 26: Average MAE across all NordPool bidding regions as a function of hour into the forecasting horizon. One can observe that the error increases further into the forecasting horizon for all models except the naive benchmark, while also exhibiting daily fluctuations

Table 25: Average price and share of forecasts above actual value for high performing model on NordPool bidding areas

	Naive 7d	ARIMA	SARIMA	ENTCN	Lin reg	Actual
Avg. (€/MWh)	16.359	15.317	16.190	16.887	15.868	16.260
Share, above actual	53.6%	50.9%	49.1%	58.6%	66.5%	-

6.3.2 Deep Dive on the NO3 (Trondheim) Bidding Area

To better understand model performance, it is interesting to take a deep dive into their performance in the NO3 (Trondheim) bidding area, the home of the authors of this thesis. Looking at Figure 27, showing the avg. MAE by the hour of the day, one can see significant variations across the day. Most notably is the performance of the linear regression model, which is much worse than that of the other models. Looking at Figure 28, showing the avg. MAE as a function of the hour into the forecasting horizon, one can see that these daily fluctuations are highly apparent, regardless of model. Furthermore, one can see that the MAE increases further into the forecasting horizon for all models except the naive benchmark. The ARIMA, SARIMA and ENTCN models start the forecasting horizon with almost no error, followed by a gradual increase throughout the forecasting horizon.

In Table 26 one can see the average forecasted price in the NO3 (Trondheim) bidding area and the share of forecasts over the actual. One can see that the naive benchmark and the ARIMA models forecast too high and too low approximately at the same rate. In contrast, the SARIMA and ENTCN models forecast too high values 55% and 59% of hours, respectively. However, one can also observe that the linear regression model forecasts too high values in over 92% of cases, which explains the poor performance of this model on the NO3 region. In contrast to the other models, linear regression is an additive model, meaning the effect of the independent variables is not dependent on the price levels. Hence, as NO3 exhibited low prices in the out-of-sample data, the linear model has consistently forecasted too high prices.

Interestingly, when zooming in on one region, one can analyse the individual forecasts made by the models. For the

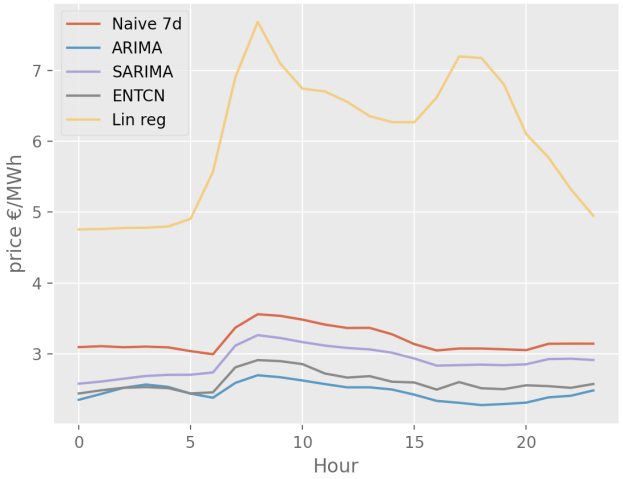


Figure 27: Average MAE in NO3 (Trondheim) as a function of hour of the day. One can observe that the highest errors occur during the morning and afternoon

NO3 region, four forecasts in which the ENTCN have performed particularly well (Figure 29 and Figure 30) or particularly poor (Figure 31 and Figure 32) have been highlighted. While it is easy to understand the dynamics of the Enhanced naive component, the TCN component is a relatively black box, making it hard to draw causal relationships between independent and dependent variables.

In the NO3 forecasting period from 00:00 15.08.2020 to 23:00 21.08.2020, seen in Figure 29, the ENTCN models perform well as the actual price holds a stable price level while following expected daily fluctuations. The ARIMA and SARIMA models, which often forecast similarly to the ENTCN, also exhibit good performances during the period. On the other hand, the naive benchmark consistently fore-

Table 26: Average price and share of forecasts above actual value for high performing model on NO3 (Trondheim)

	Naive 7d	ARIMA	SARIMA	ENTCN	Lin reg	Actual
Avg. (€/MWh)	8.500	8.353	8.553	8.889	14.082	8.330
Share, above actual	51.8%	49.2%	55.3%	59.1%	93.2%	-

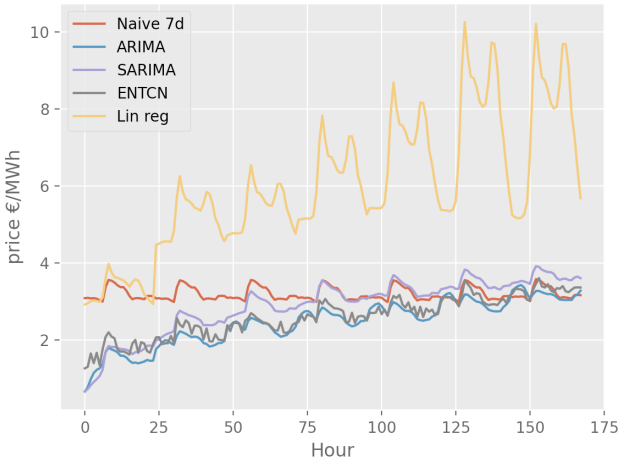


Figure 28: Average MAE in NO3 (Trondheim) as a function of hour into the forecasting horizon. One can observe that the error increases further into the forecasting horizon for all models except the naive benchmark, while also exhibiting daily fluctuations

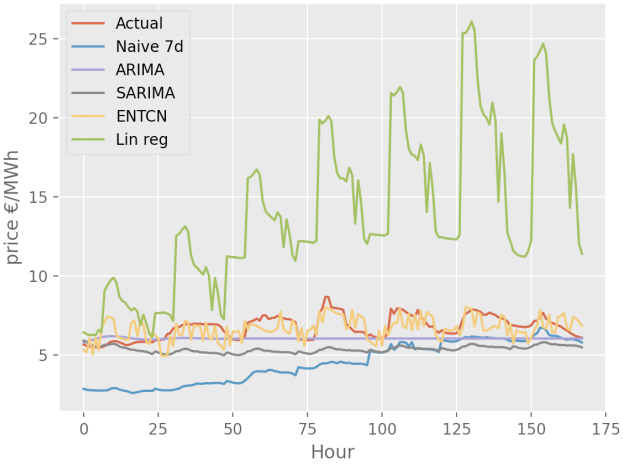


Figure 29: Forecast of high performing models of the NO3 bidding price in the period from 00:00 15.08.2020 to 23:00 21.08.2020, in which the ENTCN model performed well.

casts too low values due to values the previous week, while the additive linear regression model expects rising NO3 prices. Additionally one can analyse the NO3 forecasting period from 00:00 25.01.2020 to 23:00 31.01.2020, seen in Figure 29. Here the ENTCN, naive benchmark and the ARIMA perform well, as the price follows a similar pattern as the week before. However, both the SARIMA and linear regression models perform poorly, as they underestimate and overestimate the price movement throughout the week.

In the NO3 forecasting period from 00:00 20.05.2020 to 23:00 26.05.2020, seen in Figure 31, all the models perform very poorly. The reason is the extreme spike in price at the beginning of the forecasting horizon, which neither of the models can forecast. Furthermore, the price falls very low

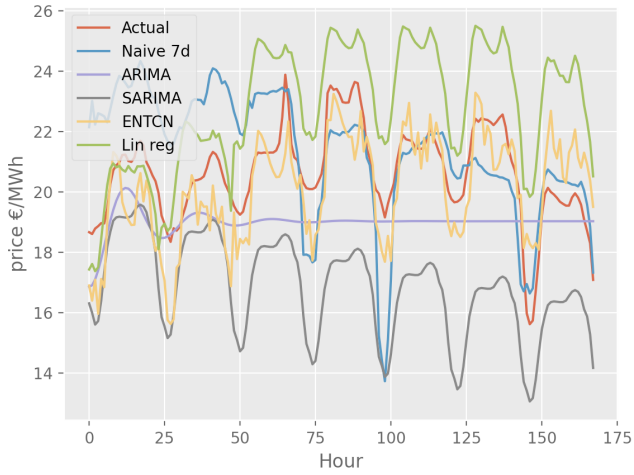


Figure 30: Forecast of high performing models of the NO3 bidding price in the period from 00:00 25.01.2020 to 23:00 31.01.2020, in which the ENTCN model performed well.

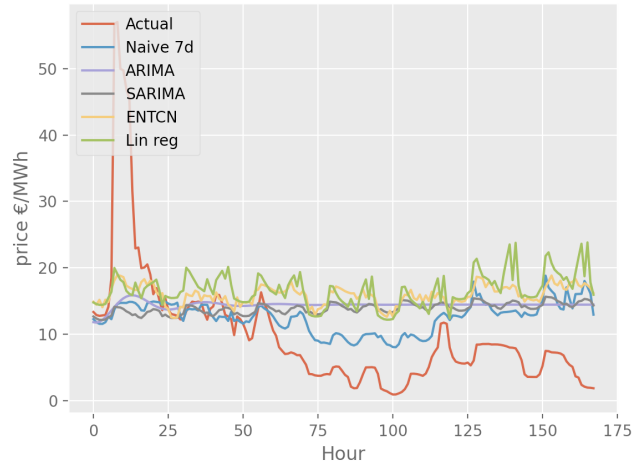


Figure 31: Forecast of high performing models of the NO3 bidding price in the period from 00:00 20.05.2020 to 23:00 26.05.2020, in which the ENTCN model performed poorly.

throughout the forecasting horizon. The only model able to somewhat forecast these movements is the seven-day naive benchmark, which mimics the similar price movements exhibited the week before. Similarly, in the period from 00:00 08.10.2020 to 23:00 14.10.2020, seen in Figure 32, all the models except the linear regression model perform poorly, as the price increases gradually throughout the week. A problem here for the ENTCN, ARIMA and SARIMA models is the low NO3 prices in the input data, making relatively small absolute changes significant in relative terms and subsequently hard to forecast. This is, however, not necessarily a problem when using linear regression as this is an additive model, which forecasts changes in prices in absolute terms.

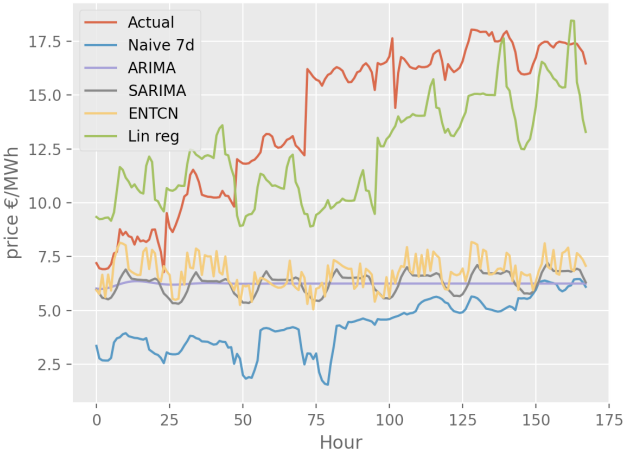


Figure 32: Forecast of high performing models of the NO3 bidding price in the period from 00:00 08.10.2020 to 23:00 14.10.2020, in which the ENTCN model performed poorly.

7 Discussion and Conclusion

In this section, we aim to discuss mainly two topics, our results and benchmark process. We believe that this will enable the reader and us better understand our benchmark results.

7.1 Performance of Statistical Models

The statistical models were the most accurate forecasting models in this benchmark. This subsection aims to understand why the statistical models had the lowest forecasting error. Our experiment's most accurate forecasting models were the SARIMA, ARIMA, linear regression, and seven-day naive. Apart from the linear regression model, all of the mentioned statistical models are univariate. Furthermore, the linear regression also has a linear and highly explainable relationship between the exogenous input variables and the output variable. Consequently, all the best-performing models use simple relationships between the input and output variables. These simple linear relationships contrast with the non-linear transformations the deep learning models perform when processing the input data to output. We have identified reasons for the simple statistical models to perform superior to more complex models such as deep learning. The first feature of the electricity price beneficial to the simpler statistical models is autocorrelation and partial autocorrelation properties. From Section 4.1, the autocorrelation and partial autocorrelation effects are clear. Consequently, one of the most valuable input variables a model can have is the lagged value of the electricity prices. In addition to autocorrelation, the electricity prices in the NordPool area are stable due to the high degree of hydropower-based electricity generation (Statista 2019). Since the electricity production from a hydro reservoir is more flexible than variable renewable energy sources, such as a solar power farm, it will lead to more stable prices. This is due to the fact that a hydro reservoir functions as a massive battery. This property of hydro reservoirs enables the producers to increase supply when demand is higher and lower supply when demand is lower, resulting in more stable prices. Consequently, there is less need to model complex price relationships. The more stable prices are beneficial to the simpler statistical models, as these models do not estimate substantial changes in the electricity price from past values. The third feature making the simple models perform well is the absence of exogenous variables in the input. From looking at the correlation analysis in Section 4.2, it is likely not much explanatory power that can be added by including numerous exogenous variables. Therefore, models such as SARIMA and ARIMA make conservative estimates with input variables that do not cause them to forecast deviations in either direction by omitting exogenous variables. The fourth feature that makes the statistical models accurate is taking calendar effects into their forecast. In Section 4 we also highlighted the calendar effects that electricity prices often exhibit. The calendar effects have been implemented in statistical models in manners that do not overly complicate the model. The simplest form is the naive seven days, which uses the price one week prior (168 hours) to the forecast time. The SARIMA and ARIMA models have somewhat more complicated methods for weighting lagged values. Nevertheless, the parameter estimation ensures that the models extract the autocorrelation features in the most suitable manners (Brooks 2019; Durbin 2012).

We believe some properties of the electricity price have enabled the statistical models to perform superior in our experiment. First, the electricity price exhibits strong autocorrelation properties. Hence, using lagged values as input makes a good base for forecasting. Second, the NordPool markets have relatively stable prices due to the high hydropower electricity generation share. Stable prices are beneficial for statistical models that do conservative forecasts. Thirdly, there is not necessarily substantial additional explanatory value in exogenous variables, which makes the models avoid wrongful relationships between the input and output by omitting the exogenous variables from the input. Lastly, the statistical models adjust for the calendar effects the electricity price exhibits; the simple statistical models performed the best in our benchmark.

7.2 Performance of Deep Learning Models

Although deep learning models have been found to perform strongly for electricity price forecasting tasks (Kuo and Huang 2018; Lago, Ridder et al. 2018; Wan et al. 2019), we found the opposite to be the case. This observation is interesting since the deep learning models should be capable of modeling highly complex relationships between the input and output variables. This observation is also highly exemplified in Figure 33 and Figure 34. These figures show the average error across every 24 hours of the day and a week, respectively. The deep learning models have substantially higher forecasting errors in all hours of both the day and week compared to the statistical models and ENTCN.

From Figure 33, it seems that the forecasting error of the deep learning models is higher as the electricity price increases. This can be observed as the lower error at night and higher error in the morning (relative to other hours of the day). The same observation can be made when looking at the average forecasting error across a week. Although the error is expected to increase as the forecasting horizon increases, the deep learning models still seem to have a forecasting error pattern that resembles the electricity price.

Overfitting is a potential hazard when looking at the complexity of the models and the errors. However, in Table 8, it is described that the deep learning models are trained for a total of 30 epochs. After 30 epochs, we experienced small reductions in error after a previously steep decline. Consequently, overfitting is likely not the explanation for the inaccuracy of the deep learning models compared to the simpler statistical models (Goodfellow et al. 2016).

A possible reason for the poorer performance of the deep learning models is that the models are stuck in a local minimum, where there is too much explanatory power in exogenous variables. This theory is interesting and can hold some explanatory power. However, the group has taken two measures to avoid a situation where the models are in a local minimum with excessive explanatory power given to exogenous variables. First, a wrapper method, described in Section 5, was implemented and used for feature selection. The wrapper should ensure that the models omit variables that have little explanatory power. How-

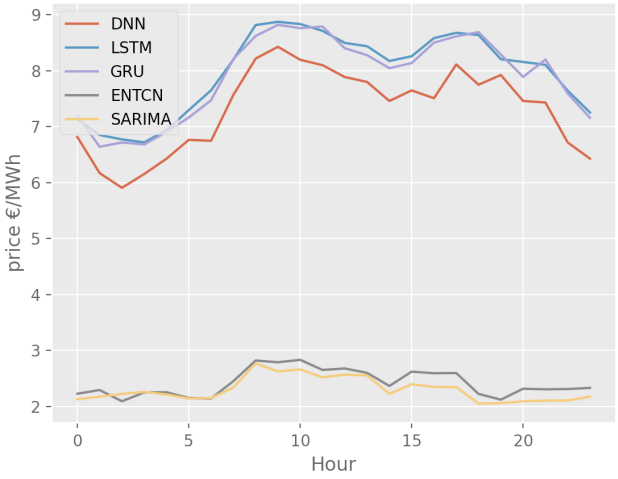


Figure 33: Average MAE in NO3 (Trondheim) as a function of hour of the day for selected models. One can see the deep learning (DNN, LSTM, and GRU) models have significantly higher errors than the SARIMA and ENTCN for all hours of the day

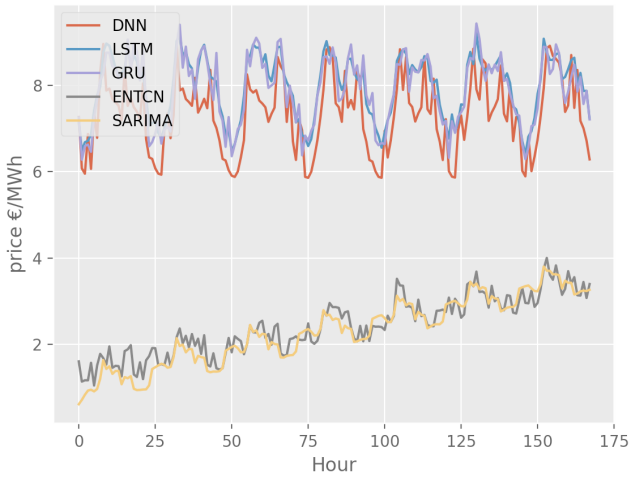


Figure 34: Average MAE in NO3 (Trondheim) as a function of hour into the forecasting horizon for selected models. One can see the deep learning (DNN, LSTM, and GRU) models have significantly high errors throughout the whole forecasting horizon, while the SARIMA and ENTCN have gradually increasing error throughout the forecasting horizon

ever, from Table 17, it is evident that the deep learning models have many features as input variables, which increases the risk of giving explanatory power to insufficiently relevant variables. This issue can be handled by increasing the level of significance required for exogenous variables to be tested in the model. The second means used to avoid a local minimum was the learning rate. A higher learning rate might have yielded a model with different and more desirable parameters. Nevertheless, we tested for multiple values for the learning rate and found the learning rate reported in Table 8 to be the best in the validation data. As a result of our two means taken to avoid such a situation, it is not very likely that this is the case for the models.

To find starting points for our hyperparameters, we used the literature described in Section 5. As stated by Jedrzejewski et al. (2022), 90% of the literature written about electricity price forecasting is about the day-ahead market. As a result, the initial values around which we conducted the sensitivity search for hyperparameters were reported when forecasting the day-ahead price. Consequently, the hyperparameter values can be vastly different from values that would have given us a superior forecast, assuming they exist. In order to combat such an event, where the initial values for the model hyperparameters are vastly different from what is required, we had broad searches for different values. However, a multi-dimensional grid search was not conducted since it would be exceedingly resource-consuming and not common to do (Goodfellow et al. 2016). As a result, we cannot rule out that a linear combination of hyperparameter values found to be suboptimal will not create models that forecast better.

An explanation for the larger errors for the deep learning models is that they are not provided with the correct data to model the future electricity price appropriately. In Section 3 and Table 1 it is evident that numerous models use features such as load forecast as input. Including forecasts of other relevant exogenous variables can increase the forecasting accuracy of the deep learning models.

An alternative to including more data sources is to include data from further in the past. This will give us more training data and enable the model to train on a broader range of data. We did not have a vast amount of training sample available, only around 2000. Goodfellow et al. (2016) highlight that substantial amounts of data is required for making strong deep learning models. Consequently, adding older data to our data set can improve the training of the models. However, there are also downsides to adding data from further back. Since the electricity markets are continuously evolving, new grid interconnections, fuel sources, and other external factors result in structural breaks in the electricity prices. Consequently, including older data samples might mean the inclusion of irrelevant data, which means that the market has evolved to such an extent that the oldest training samples no longer represent the different variable relationships present in the market. Naturally, a solution to this problem is to update the data set with the most recent data. In order to be in line with the best practices of electricity price forecasting, the test period must be shifted if the most recent data is added.

The deep neural network displays a lower forecasting error than the LSTM and GRU. The difference in error is observable in Figure 33 and Figure 34 and verified by the Diebold-Mariano test. A possible explanation for this result is that the deep neural network has the flexibility to weight input steps differently. The RNNs are not capable of this weighting due to parameter sharing, explained in Section 2.1.3. The ability to weigh inputs differently has proven to improve the forecasts of other electricity price forecasting models. Examples are Wan et al. (2019) and Lago, Ridder et al. (2018). As a result, the flexibility of the deep neural network is a possible reason for the observation that the DNN model performs superior to the RNNs.

7.3 Discussing the ENTCN

As a part of our contribution to the literature, we benchmarked the ENTCN-model against numerous other electricity price forecasting models, some of which were described as state-of-the-art by other authors. We will, in this sub-

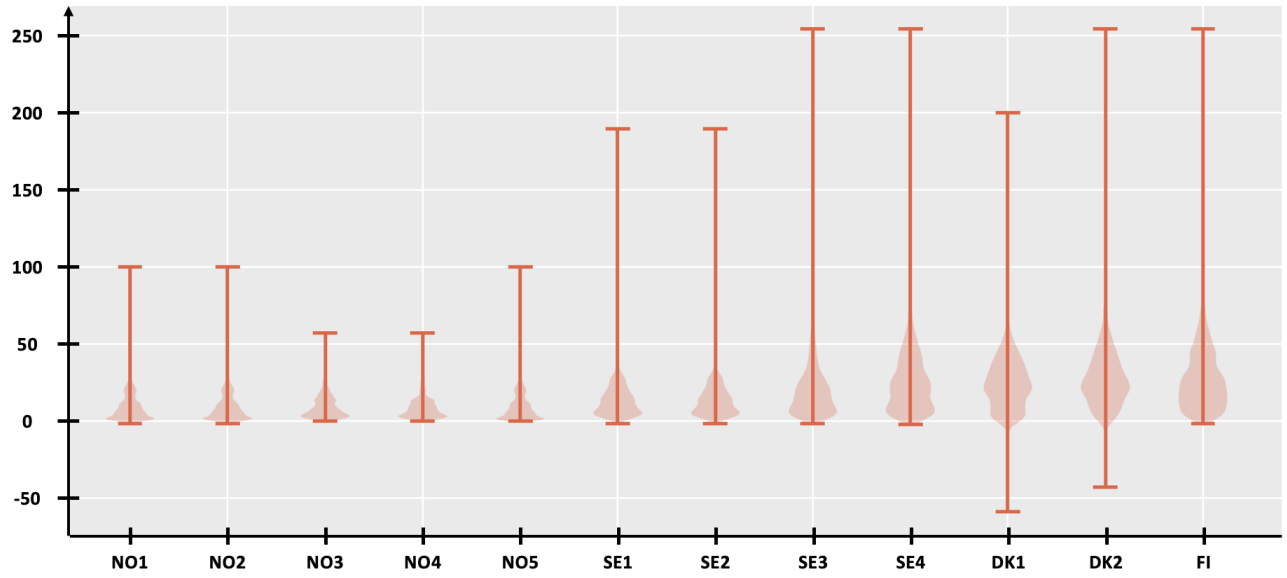


Figure 35: Violin plots showing the hourly electricity price (€/ MWh) distribution for each bidding area in the out-of-sample dataset. There are clear differences in mean, standard deviation, and skew between the different regions.

section, discuss the results of the ENTCN-model. Our proposed model turned out to be fairly accurate. The model consists of two parts, described in Figure 21. Since the ENTCN had a feature selection and a hyperparameter tuning process similar to the other deep learning models previously described, it might suffer from some of the potential pitfalls of the deep learning models. These shortcomings are described in Section 7.2. The temporal convolutional neural network did not add an exceedingly more accurate forecast. The changes the convolutional layers made to the enhanced naive forecasts were small. Consequently, the TCN-layers did not extract increased explanatory information from the input data. As a result, it is hard to conclude that the ENTCN-model we have proposed is very successful for electricity price forecasting.

7.4 Generalizability

For research to be valid, the results should be generalizable. A piece of research is generalizable, meaning that it can be used in other similar problems.

Looking at the generalizability first, the proposed model and all other models made in this thesis were tested in twelve different regions. Testing in different regions is good for generalizable results. Furthermore, Figure 35 showing the actual price distribution in the out-of-sample data for each region, shows that there are also large differences across regions, further emphasising the generalizability of the benchmark. However, we have solely used the NordPool area, where there is a similarity between the markets. The similarity comes from the grid interconnectivity described in Section 2. For example, forecasting the price in NO1 compared to NO2 is not that different. Furthermore, the two regions have a linear correlation coefficient of 0.96, which is observable in Figure 20. A feature of our study which makes it more generalizable is the length of the test data. As highlighted in Section 3 and by the papers Croonenbroeck and Stadtmann (2019) and Jedrzejewski et al. (2022), a test period for a year is required for generalizable results. Our test period was the entire year of 2020, which is on par with the best practices. Naturally, 2020 proved to be a year different from many of the preceding years, which can have affected the electricity price in this period. Nevertheless, it was the most recent year in our data set, so we used it as the test period. Even though the models have been tested across twelve different bidding regions and had a test period across a year, we can conclude that more work can be done to investigate our models' generalizability. These improvements are further discussed in Section 8.

7.5 Forecasting Spikes in the Electricity Price

Electricity price forecasting models that are capable of forecasting price spikes are powerful. Hellström et al. (2012) highlight that price spikes are often caused by exogenous factors that strongly shift demand or supply in either direction. An example can be windy weather conditions which causes substantial amounts of wind energy to be produced. Consequently, creating a negative price spike. This price spike is predictable if weather forecasts are included as an input variable. Furthermore, Jedrzejewski et al. (2022) highlight that the electricity price and grid load exhibit a non-linear relationship. From the two observations by Hellström et al. (2012) and Jedrzejewski et al. (2022), a deep learning model with reliable forecasts and abilities to model non-linear relationships will therefore be capable of predicting price spikes, naturally assuming proper training and architecture. However, the deep learning models in this experiment did not have forecasts as input variables. Consequently, we were not capable of capture the non-linear forecasting capabilities of the deep learning models. This can also explain why the deep learning models did not perform as well as the statistical models.

7.6 Electricity Price Forecasting Best Practices

To ensure reproducibility and meaningful results, the thesis follows a list of eight electricity price forecasting best practices listed in Jedrzejewski et al. (2022) and (Lago, Marcjasz et al. 2021), as discussed in Section 5:

1. The out-of-sample data is selected as the last section of the data, with 2014-2019 used for training while 2020 is used for testing. Furthermore, the hyperparameters are set using a validation dataset from the in-sample data, consisting of 5% of the training data.
2. All new models are tested against well-known state-of-the-art models, with open-source libraries such as Tensorflow/ Keras, statsmodels, and sklearn. Furthermore, all the data is sourced from open-access datasets such as NordPool and the Norwegian Meteorological Institute.
3. Several error metrics, both absolute and relative, are used to evaluate the models, including MAE, MAPE, SMAPE, and RMSE.
4. The statistical test Diebold-Mariano is used to assess the statistical significance of differences in predictive performance between the models.
5. The split and dates of the dataset are explicitly stated, as discussed Section 5.5.
6. All the inputs of the models are explicitly stated in Table 17.
7. The computational costs of the methods are evaluated and compared with the running time reported. Here one could also compare the big O time complexities of the models.
8. A number of forecasting models are recalibrated daily, incl., SARIMA and ARIMA. Given better processing power or more time available, one could also use to recalibrate the deep learning models more often, as these are only calibrated at the beginning of the test period.

7.7 Conclusion

To conclude the thesis, the benchmark showed that the statistical models performed comparably better than the deep learning ones across all bidding areas and error metrics. In addition, the hybrid ENTCN model performed significantly better than the deep learning models but was often outperformed by simpler statistical models. Furthermore, there was a little statistically significant improvement in adding the TCN component to the enhanced naive, somewhat invalidating the attractiveness of the ENTCN model. The enhanced naive model even performed better than the proposed ENTCN model in some bidding regions. There were differences in which models performed best across different bidding areas and countries. The ARIMA model performed best across all error metrics in Norway, while the SARIMA model was the highest performing in Denmark. However, the linear regression model performed best in both Sweden and Finland. On average, across the 12 NordPool bidding areas, the SARIMA performed best on the absolute error metrics while the ARIMA did best on the relative error metrics. These results were obtained using data from 2014-2019 for training and data from 2020 for testing. However, these results might differ if the models were tested or trained on other time periods and might not even be generalizable to more volatile and higher NordPool prices, as seen in 2021 and 2022. Furthermore, these results can not be used to make any inferences on performance in other electricity markets. However, the benchmark indicates their performance on the respective NordPool bidding areas in the Nordics, providing a new perspective to the field of electricity price forecasting. In summary thesis contributes to the literature and the field of electricity price forecasting in two main ways. Firstly, it provides an up to date systematic benchmarking of multiple state-of-the-art methods across multiple NordPool bidding areas in accordance with electricity price forecasting best practices (Jedrzejewski et al. 2022). Secondly, the thesis provides development and a state-of-the-art benchmark of a hybrid ENTCN model, which was developed in a project thesis by the authors of this thesis (T. R. Wang et al. 2021).

8 Further Work

In this section, we have identified several areas of potential further work on the benchmark in this thesis. The potential areas of further work include developing ensemble models, tuning hyperparameters, testing on other power markets, and using new independent variables.

8.0.1 The Development of Ensemble Models

Many other benchmarks have shown ensemble models, which are models combining two or more models, to exhibit improved predictive accuracy. Ensembling of models is a common way in which to improve performance (Goodfellow et al. 2016), an example including Lago, Marcjasz et al. (2021) which showed that ensemble DNN models generally outperformed single models. Hence, it would be exciting to benchmark different ensemble permutations of the models implemented in this thesis. However, one concern regarding the use of ensemble models is the high correlation in forecasting between the high-performing models in the current thesis (as can be seen Figure 31), as most work somewhat similarly. Therefore, there is no guarantee that the ensemble models would see the same improvement as reported in other cases.

8.0.2 Tuning of Hyperparameters and Feature Selection

Although much time was used on the tuning of hyperparameters and selecting input variables in this thesis (as described in Section 5.2 and Section 5.1), this is still an area of further work. Investigating other ways of tuning hyperparameters and inputs could shed further light on the potential model performances. A potential technique would be to implement a tree-structured Parzen estimator (Bergstra et al. 2011), a Bayesian optimization algorithm based on sequential model-based optimization, implemented in both (Lago, Marcjasz et al. 2021) and (Lago, Ridder et al. 2018). Here, the optimal combination of hyperparameters and features is optimized, with the features being represented as binary hyperparameters. The challenge with doing this, and the main reason for not doing it in this thesis, is the computational requirements of doing it for a large number of models across a large number of bidding areas. However, if implementing a specific model for a certain bidding region, using more time optimizing could potentially improve performance. This thesis used the same hyperparameters and features for each bidding area. Further work should also include tuning these for each region as significant differences in price movement dynamics across regions.

8.0.3 Testing of Other Power Markets

As discussed in Jedrzejewski et al. (2022), generalizable electricity price forecasting benchmarks should aim to model several markets. Although the current thesis has benchmarked the model on 12 NordPool bidding regions, it would be highly relevant to test the models on other power markets, which might be less correlated. Examples of this could be PJM, EPEX-BE, EPEX-FR, EPEX-DE, or OMEL which all exhibit different price dynamics, vs. NordPool, with different energy generation mixes and transmission capacities. However, there is a particular focus on how the model would perform in the Nordic region in this paper. Furthermore, one could also try to benchmark the models on newer testing periods. However, the main challenge with testing different time periods is that the out-of-sample data always has to be at the end of the dataset (Jedrzejewski et al. 2022) to not test the models on data prior to the training examples.

8.0.4 Gathering of Relevant Data

When developing multivariate time series forecasting models, the features used are essential for model performance (Goodfellow et al. 2016). A challenge in the current thesis, and an area of future work, is the gathering and utilizing more (and better) data. Although there was a lot of available weather data for Norway, having the same type of data for the other Nordic countries could be highly beneficial when forecasting the SE, DK, and FI regions. Furthermore, what is lacking is historic weather forecasts, such as temperature, precipitation, sun, or sunlight, relevant for future production of variable renewable energy sources such as wind and solar. Other interesting datasources include, load forecasts, wind foreacasts, forward prices, or grid capacities between regions (which is available from NordPool for the most recent years). The difficulty here is that these might be hard to find. Interestingly, if grid capacities between regions were to be included it might counter the effects of structural breaks due to new capacities, as these changes then would be incorporated into the models. The utilization of such data could enhance the performance of the implemented multivariate model.

Bibliography

Afrasiabi, Mousa et al. (2019). 'Multi-agent microgrid energy management based on deep learning forecaster'. In: *Energy* 186.

Aineto, Diego et al. (2019). 'On the Influence of Renewable Energy Sources in Electricity Price Forecasting in the Iberian Market'. In: *Energies* 12.

Akaike, H. (1974). 'A new look at the statistical model identification'. In: *IEEE Transactions on Automatic Control* 19.6, pp. 716–723.

Andina, Diego et al. (2007). *Neural Networks Historical Review*.

Atef, Sara and Amr B. Eltawil (2019). 'A Comparative Study Using Deep Learning and Support Vector Regression for Electricity Price Forecasting in Smart Grids'. In: *2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)*, pp. 603–607.

Bai, Shaojie, Zico Kolter and Vladlen Koltun (2018). *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*.

Bento, P.M.R. et al. (2018). 'A bat optimized neural network and wavelet transform approach for short-term price forecasting'. In: *Applied Energy* 210, pp. 88–97.

Bergstra, James et al. (2011). 'Algorithms for Hyper-Parameter Optimization'. In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor et al. Vol. 24. Curran Associates, Inc.

Bhandare, Ashwin et al. (2016). 'Applications of Convolutional Neural Networks'. In: *International Journal of Computer Science and Information Technologies* 7.

Blazquez, Jorge et al. (2018). 'The renewable energy policy Paradox'. In: *Renewable and Sustainable Energy Reviews* 82, pp. 1–5.

Brancucci Martinez-Anido, Carlo, Greg Brinkman and Bri-Mathias Hodge (2016). 'The impact of wind power on electricity prices'. In: *Renewable Energy* 94, pp. 474–487.

Brooks, Chris (2019). *Introduction to Econometrics for Finance*. 4th. Cambridge University Press.

Bunn, Derek (2004). *Modelling Prices in Competitive Electricity Markets*. Wiley.

Burke, Paul and Ashani Abayasekara (2017). 'The price elasticity of electricity demand in the United States: A three-dimensional analysis'. In: *SSRN Electronic Journal*.

Bye, Torstein and Einar Hope (2005). 'Deregulation of electricity markets—The Norwegian experience'. In: *Economic and Political Weekly* 40.

Chen, Christie and Yang-yu Wang (3rd Oct. 2021). 'Full power ahead for UK to Norway under-sea power cable'. In: *BBC News*. URL: https://www.bbc.com/news/uk-england-tyne-58772572 (visited on 20th Nov. 2021).

Cho, KyungHyun et al. (2014). 'On the Properties of Neural Machine Translation: Encoder-Decoder Approaches'. In: *CoRR*.

Croonenbroeck, Carsten and Georg Stadtmann (2019). 'Renewable generation forecast studies – Review and good practice guidance'. In: *Renewable and Sustainable Energy Reviews* 108, pp. 312–322.

Cruz, Alberto et al. (2011). 'The effect of wind generation and weekday on Spanish electricity spot price forecasting'. In: *Electric Power Systems Research* 81, pp. 1924–1935.

Dancey, Christine and John Reidy (2011). *Statistics Without Maths for Psychology*. 5th. Prentice Hall.

Darudi, Ali, Masoud Bashari and Mohammad Hossein Javidi (2015). 'Electricity price forecasting using a new data fusion algorithm'. In: *IET Generation Transmission Distribution* 9, pp. 1382–1390.

Diebold, Francis and Roberto Mariano (1995). 'Comparing Predictive Accuracy'. In: *Journal of Business and Economic Statistics* 13, pp. 253–265.

Durbin, James (2012). *Time Series Analysis by State Space Methods*. 2nd ed. Vol. 38. Oxford: Oxford University Press.

Eichler, Michael et al. (2013). 'Models for short-term forecasting of spike occurrences in Australian electricity markets: A comparative study'. In: *The Journal of Energy Markets* 7.

Engebretsen, Jakob et al. (2020). *A Systematic Literature Review of Electricity Price Forecasting*.

— (2021). *Mid-Term Electricity Price Forecasting using Auction Data to Construct Supply and Demand Curves*.

Eydeland, Alexander and Krzysztof Wolyniec (n.d.). *Energy and Power Risk Management*. 1st.

Fleten, Stein-Erik and Trine Krogh (2008). 'Short-term hydropower production planning by stochastic programming'. In: *Computers and Operations Research* 35.

Forecasting, Aleasoft Energy (2019). *European electricity markets panorama: Nordic countries*.

Fu, Yihao and Chris Aldrich (2018). 'Using Convolutional Neural Networks to Develop State-of-the-Art Flotation Froth Image Sensors'. In: *IFAC-PapersOnLine* 51, pp. 152–157.

García-Ascanio, Carolina and Carlos Maté (2010). 'Electric power demand forecasting using interval time series: A comparison between VAR and iMLP'. In: *Energy Policy* 38, pp. 715–725.

Ghayekhloo, M. et al. (2019). 'A combination approach based on a novel data clustering method and Bayesian recurrent neural network for day-ahead price forecasting of electricity markets'. In: *Electric Power Systems Research* 168, pp. 184–199.

Giacomini, Raffaella and Halbert White (2006). 'Tests of Conditional Predictive Ability'. In: *Econometrica* 74, pp. 1545–1578.

Goodfellow, Ian, Yoshua Bengio and Aaron Courville (2016). *Deep Learning*. MIT Press.

Goto, Mika and Andrew Karolyi (2004). 'Understanding Electricity Price Volatility Within and Across Markets'. In: *SSRN Electronic Journal*.

Härdle, Wolfgang Karl Karl and Stefan Trueck (2010). 'The dynamics of hourly electricity prices'. In: *SFB 649 Discussion paper 2010-013*.

Hauser, Christine and Edgar Sandoval (2021). 'Death Toll From Texas Winter Storm Continues to Rise'. In: *New York Times*. URL: https://www.nytimes.com/2021/07/14/us/texas-winter-storm-deaths.html (visited on 7th Oct. 2021).

He, Kaiming et al. (2015). *Deep Residual Learning for Image Recognition*.

Hellström, Jörgen, Jens Lundgren and Haishan Yu (2012). 'Why do electricity prices jump? Empirical evidence from the Nordic electricity market'. In: *Energy Economics* 34 (6).

Hochreiter, Sepp and Jürgen Schmidhuber (Dec. 1997). 'Long Short-term Memory'. In: *Neural computation* 9, pp. 1735–80.

Infrastructure, Ministry of (2022). 'The situation regarding Sweden's energy supply in light of Russia's invasion of Ukraine'. In: *Goverment Offices of Sweden*. URL: https://www.government.se/articles/2022/03/the-situation-regarding-swedens-energy-supply-in-light-of-russias-invasion-of-ukraine/ (visited on 2nd Apr. 2022).

Ismail, Mohammed (1989). 'Analog VLSI Implementation of Neural Systems'. In: *The Kluwer International Series in Engineering and Computer Science* 80.

Jedrzejewski, Arkadiusz et al. (2022). 'Electricity Price Forecasting: The Dawn of Machine Learning'. In: *IEEE Power and Energy Magazine* 20, pp. 24–31.

Joskow, Paul (2001). 'California's electricity crisis'. In: *Oxford Review of Economic Policy* 17, pp. 365–388.

Kaminski, Vincent (2013). *Energy Markets*. Risk Books.

Keras, Chollet (2015). *Temporal Convolutional Networks for Keras*. https://github.com/keras-team/keras.

Khan, Gul Muhammad, Rabia Arshad and Nadia Masood Khan (2017). 'Efficient Prediction of Dynamic Tariff in Smart Grid Using CGP Evolved Artificial Neural Networks'. In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 493–498.

Khare, Vikas, Savita Nema and Prashant Baredar (2016). 'Solar–wind hybrid renewable energy system: A review'. In: *Renewable and Sustainable Energy Reviews* 58, pp. 23–33.

Kontogiannis, Dimitrios et al. (2022). 'Error Compensation Enhanced Day-Ahead Electricity Price Forecasting'. In: *Energies* 15.

Kragelund, Martin et al. (2010). *Optimal Production Planning of a Power Plant*.

Kuo, Ping-Huan and Chiou-Jye Huang (2018). 'An Electricity Price Forecasting Model by Hybrid Structured Deep Neural Networks'. In: *Sustainability* 10.4.

Lago, Jesus, Grzegorz Marcjasz et al. (2021). 'Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark'. In: *Applied Energy* 293.

Lago, Jesus, Fjo De Ridder and Bart De Schutter (2018). 'Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms'. In: *Applied Energy* 221, pp. 386–405.

Lea, Colin et al. (2016). *Temporal Convolutional Networks: A Unified Approach to Action Segmentation*.

Lee, Chien and Jun Lee (2009). 'Energy prices, multiple structural breaks, and efficient market hypothesis'. In: *Applied Energy* 86, pp. 466–479.

Legendre, Adrien-Marie (1805). 'Sur la Méthode des moindres quarrés'. In: *Nouvelles méthodes pour la détermination des orbites des comètes*.

Li, Wei and Denis Mike Becker (2021). 'Day-ahead electricity price prediction applying hybrid models of LSTM-based deep learning methods and feature selection algorithms under consideration of market coupling'. In: *Energy* 237.

Lin, Lei et al. (2014). 'On-line prediction of border crossing traffic using an enhanced Spinning Network method'. In: *Transportation Research Part C: Emerging Technologies* 43, pp. 158–173.

Long, Jonathan, Evan Shelhamer and Trevor Darrell (2015). *Fully Convolutional Networks for Semantic Segmentation*.

Marcjasz, Grzegorz, Jesus Lago and Rafał Weron (2020). *Neural networks in day-ahead electricity price forecasting: Single vs. multiple outputs*.

Marcjasz, Grzegorz, Bartosz Uniejewski and Rafał Weron (2019). 'On the importance of the long-term seasonal component in day-ahead electricity price forecasting with NARX neural networks'. In: *International Journal of Forecasting* 35.4, pp. 1520–1532.

McCulloch, Warren and Walter Pitts (1943). 'A logical calculus of the ideas immanent in nervous activity'. In: *The bulletin of mathematical biophysics* 5, pp. 115–133.

Meier, Jan-Hendrik et al. (2019). 'ANN-Based Electricity Price Forecasting Under Special Consideration of Time Series Properties'. In: ed. by Vadim Ermolayev et al., pp. 262–275.

MIT (Mar. 2020a). *Lecture notes in MIT 6.036, Chaper 8, Neural Networks*.

— (Mar. 2020b). *Lecture notes in MIT 6.036, Chapter 7, Regression*.

NordPool (2021). *NordPool About Us*. URL: https://www.nordpoolgroup.com/About-us/ (visited on 18th Oct. 2021).

— (2022). *NordPool: History*. URL: https://www.nordpoolgroup.com/en/About-us/History/.

Oord, Aaron van den et al. (2016). *WaveNet: A Generative Model for Raw Audio*.

Pechman, Carl and Elliott Nethercutt (2021). 'Regulatory Questions Engendered by the Texas Energy Crisis of 2021'. In: *NRRI insights*.

Peter, Smitha and Jacob Raglend (2017). 'Sequential wavelet-ANN with embedded ANN-PSO hybrid electricity price forecasting model for Indian energy exchange'. In: *Neural Computing and Applications* 28, pp. 2277–2292.

Rantonen, Mika and Joni Korpihalkola (2020). 'Prediction of Spot Prices in Nord Pool's Day-Ahead Market Using Machine Learning and Deep Learning'. In: *Machine Learning, Optimization, and Data Science*. Springer International Publishing, pp. 676–687.

Remy, Philippe (2020). *Temporal Convolutional Networks for Keras*. https://github.com/philipperemy/keras-tcn. (Visited on 10th Sept. 2021).

Rintamäki, Tuomas, Afzal Siddiqui and Ahti Salo (2017). 'Does renewable energy generation decrease the volatility of electricity prices? An analysis of Denmark and Germany'. In: *Energy Economics* 62, pp. 270–282.

Roldan-Fernandez, Juan-Manuel et al. (2016). 'The Merit-Order Effect of Energy Efficiency'. In: *Energy Procedia* 106, pp. 175–184.

Schnürch, Simon and Andreas Wagner (2020). *Electricity Price Forecasting with Neural Networks on EPEX Order Books*. Vol. 27. 3. Informa UK Limited, pp. 189–206.

Schweppe, Fred et al. (1988). *Spot Pricing of Electricity*. Springer.

Seabold, Skipper and Josef Perktold (2010). 'Statsmodels: Econometric and statistical modeling with python'. In: *9th Python in Science Conference*.

Statista (2019). *Distribution of electricity production in Norway in 2019, by source*. URL: https://www.statista.com/statistics/1025497/distribution-of-electricity-

production-in-norway-by-source/ (visited on 18th Nov. 2021).

Talari, Saber et al. (2017). 'Price Forecasting of Electricity Markets in the Presence of a High Penetration of Wind Power Generators'. In: *Sustainability* 9.

Tong, Yuanren et al. (2020). 'Can natural language processing help differentiate inflammatory intestinal diseases in China? Models applying random forest and convolutional neural network approaches'. In: *BMC Neducak Informatics and Decision Making* 20.

Ugurlu, Umut, Ilkay Oksuz and Oktay Tas (2018). 'Electricity Price Forecasting Using Recurrent Neural Networks'. In: *Energies* 11.

Ugurlu, Umut, Oktay Tas et al. (2018). 'The Financial Effect of the Electricity Price Forecasts' Inaccuracy on a Hydro-Based Generation Company'. In: *Energies* 11.

Ventosa, Mariano et al. (2007). *What Is Computational Intelligence and Where Is It Going?* Springer, pp. 1–13.

Vigen, Tyler (2021). *Spurious Correlations*. URL: https://tylervigen.com/spurious-correlations (visited on 16th Apr. 2022).

Wan, Renzhuo et al. (2019). 'Multivariate Temporal Convolutional Network: A Deep Neural Networks Approach for Multivariate Time Series Forecasting'. In: *Electronics* 8.8.

Wang, Tarje Rusten et al. (2021). *Application of Temporal Convolutional Network for Mid-term Electricity Price Forecasting*. URL: https://drive.google.com/file/d/1zISCZimNj72o9LM_uYL8saSv-pF400eO/view?usp=sharing.

Weron, Rafal (2014). 'Electricity price forecasting: A review of the state-of-the-art with a look into the future'. In: *International Journal of Forecasting* 30, pp. 1030–1081.

Westfall, Peter (2014). 'Kurtosis as Peakedness, 1905 - 2014. R.I.P.' In: *The American statistician* 68.3, pp. 191–195.

Windler, Torben, Jan Busse and Julia Rieck (2019). 'One month-ahead electricity price forecasting in the context of production planning'. In: *Journal of Cleaner Production* 238.

Yadav, Anamika, Rajagopal Peesapati and Niranjan Kumar (2017). 'Electricity Price Forecasting and Classification Through Wavelet–Dynamic Weighted PSO–FFNN Approach'. In: *IEEE Systems Journal*, pp. 1–10.

Yan, Jining et al. (2020). 'Temporal Convolutional Networks for the Advance Prediction of ENSO'. In: *Scientific Reports* 10.

Yang, Haolin and Kristen R. Schell (2020). 'HFNet: Forecasting Real-Time Electricity Price via Novel GRU Architectures'. In: *IEEE*, pp. 1–6.

— (2022). 'GHTnet: Tri-Branch deep learning network for real-time electricity price forecasting'. In: *Energy* 238.

Zedda, Stefano and Giovanni Masala (2019). 'Price spikes in the electricity markets: how and why'. In: *HAEE Annual Conference: Energy Transition: European and Global Perspectives*.

Zhang, Fan, Hasan Fleyeh and Chris Bales (2022). 'A hybrid model based on bidirectional long short-term memory neural network and Catboost for short-term electricity spot price forecasting'. In: *Journal of the Operational Research Society* 73.2, pp. 301–325.

Zhang, Wenjie, Farwa Cheema and Dipti Srinivasan (2018). 'Forecasting of Electricity Prices Using Deep Learning Networks'. In: *2018 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pp. 451–456.

Zhang, Yuanyuan et al. (2022). 'Research on credit rating and risk measurement of electricity retailers based on Bayesian Best Worst Method-Cloud Model and improved Credit Metrics model in China's power market'. In: *Energy*.

Ziel, Florian and Rick Steinert (2018). 'Probabilistic Mid- and Long-Term Electricity Price Forecasting'. In: *Renewable and Sustainable Energy Reviews* 94, pp. 251–266.

Zihan, Changa, Zhangb Yang and Chena Wenbo (2019). 'Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform'. In: *Energy* 187, p. 115804.

Appendix

A Abbrevations

The used abbreviations and terminology is listed below:

- ACF: Complete auto-correlation function
- ADF: Augmented Dickey-Fuller test
- AI: Artificial intelligence
- AIC: Akaike information criterion
- APX: Energy market operating in the Netherlands, the United Kingdom, and Belgium
- AR: Autoregressive
- ARIMA: Autoregressive integrated moving average
- ARMA: Autoregressive (AR) with moving average (MA)
- CNN: Convolutional neural network
- CI: Computational intelligence
- DL: Deep-learning
- DM: Diebold-Mariano (also referees to the Diebold-Mariano test statistic)
- DNN: Deep neural network
- EEX: European Energy Exchange
- EN: Enhanced naive
- ENTCN: Enhance naive temporal convolutional network, the hybrid model combining the developed Enhanced naive model and the TCN model
- EPF: Electricity price forecasting
- GRU: Gated recurrent unit
- GW: Giacomini-White
- LSTM: Long-short term memory
- MA: Moving average
- MAE: Mean average error
- MAPE: Mean average percentage error
- ML: Machine learning
- MET: Norwegian Meteorological Institute
- MLP: Multi-layer perceptron
- MWh: Megawatt hour
- NordPool: Pan-European power exchange, with main operations in the Nordics. In this thesis it often refers to the NordPool operations in Norway, Sweden, Denmark and Finland
- OLS: Ordinary least squares
- OMEL/ OMIE: Energy market operating in Spain/ Portugal
- PACF: Partial auto-correlation function
- PJM (Pennsylvania Jersey Maryland): Regional transmission organization in the US
- ReLU: Rectified Linear Unit ($f(x) = max(0, x)$)
- RMSE: Root-mean-square deviation
- RNN: Recurrent neural network

- SMAPE: Symmetric mean average percentage error
- SGD: Stochastic gradient descent
- S-GRU: Stacked gated recurrent unit
- S-LSTM: Stacked long-short term memory
- TCN: Temporal convolutional network, a form of CNN used on time series data
- VRE: Variable renewable energy (e.g., wind and solar)

B Further Data Analysis

This section presents the values for calendar effects, auto-correlation plots, and partial autocorrelation plots for the remaining price areas in the NordPool area.

B.1 Calendar effect variables

Table 27: Hourly coefficients over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all non-NO-regions.

Hour	SE1	SE2	SE3	SE4	DK1	DK2	FI
0	0.886	0.886	0.875	0.851	0.828	0.817	0.782
1	0.852	0.852	0.841	0.818	0.791	0.772	0.749
2	0.831	0.831	0.820	0.797	0.768	0.746	0.730
3	0.823	0.823	0.813	0.790	0.758	0.737	0.724
4	0.838	0.837	0.827	0.804	0.770	0.753	0.743
5	0.891	0.891	0.880	0.856	0.822	0.814	0.850
6	0.972	0.972	0.961	0.950	0.966	0.946	1.010
7	1.073	1.073	1.082	1.101	1.123	1.124	1.164
8	1.129	1.129	1.155	1.180	1.191	1.205	1.241
9	1.121	1.121	1.139	1.159	1.164	1.181	1.220
10	1.106	1.106	1.119	1.131	1.127	1.147	1.184
11	1.088	1.088	1.096	1.106	1.096	1.115	1.160
12	1.063	1.063	1.065	1.066	1.046	1.068	1.134
13	1.045	1.045	1.044	1.044	1.017	1.041	1.101
14	1.031	1.031	1.029	1.025	0.997	1.020	1.069
15	1.031	1.031	1.031	1.026	1.002	1.023	1.074
16	1.044	1.044	1.051	1.054	1.034	1.058	1.101
17	1.084	1.084	1.103	1.126	1.137	1.154	1.136
18	1.091	1.091	1.106	1.139	1.177	1.185	1.159
19	1.072	1.072	1.073	1.110	1.165	1.159	1.079
20	1.035	1.035	1.027	1.046	1.106	1.087	0.959
21	1.011	1.010	0.999	0.995	1.042	1.020	0.935
22	0.974	0.974	0.962	0.946	0.988	0.961	0.885
23	0.912	0.912	0.901	0.880	0.885	0.867	0.811

Table 28: Weekday and holiday coefficients over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all non-NO-regions.

Type of day	SE1	SE2	SE3	SE4	DK1	DK2	FI
Monday	1.030	1.030	1.036	1.041	1.034	1.035	1.074
Tuesday	1.047	1.047	1.052	1.062	1.070	1.072	1.069
Wednesday	1.044	1.044	1.046	1.058	1.075	1.069	1.068
Thursday	1.042	1.042	1.049	1.053	1.060	1.062	1.066
Friday	1.012	1.011	1.013	1.019	1.044	1.036	1.049
Saturday	0.927	0.927	0.916	0.898	0.894	0.887	0.859
Sunday	0.899	0.899	0.889	0.868	0.823	0.838	0.815
Holiday	0.824	0.823	0.813	0.798	0.726	0.721	0.796

B.2 ACF- and PACF-PLOTS

Table 29: Monthly coefficients over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all non-NO-regions.

Month	SE1	SE2	SE3	SE4	DK1	DK2	FI
January	1.071	1.071	1.083	1.068	1.002	1.019	1.071
Februar	1.001	1.001	1.001	1.001	0.967	0.963	0.982
March	0.966	0.966	0.958	0.943	0.903	0.905	0.918
April	0.935	0.935	0.925	0.904	0.923	0.893	0.909
May	0.916	0.916	0.906	0.906	0.935	0.926	0.899
June	0.903	0.903	0.899	0.931	0.961	0.977	0.904
July	0.955	0.955	0.945	0.935	1.006	0.971	1.033
August	1.045	1.045	1.050	1.037	1.077	1.076	1.087
September	1.052	1.053	1.053	1.046	1.066	1.087	1.079
October	1.024	1.024	1.033	1.077	1.052	1.078	1.053
November	1.111	1.111	1.122	1.129	1.150	1.136	1.070
December	1.020	1.020	1.027	1.023	0.958	0.969	0.992

Table 30: Monthly average prices over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all non-NO-regions.

Month	SE1	SE2	SE3	SE4	DK1	DK2	FI
January	34.738	34.738	35.542	36.091	32.194	34.637	39.668
February	32.463	32.463	32.865	33.801	31.079	32.751	36.374
March	31.312	31.312	31.436	31.870	29.040	30.760	33.992
April	30.319	30.319	30.368	30.538	29.664	30.383	33.654
May	29.700	29.700	29.736	30.604	30.064	31.488	33.283
June	29.290	29.290	29.501	31.435	30.902	33.227	33.478
July	30.954	30.954	31.014	31.583	32.323	33.033	38.241
August	33.894	33.894	34.484	35.035	34.612	36.590	40.237
September	34.117	34.141	34.570	35.350	34.272	36.951	39.936
October	33.191	33.191	33.907	36.371	33.818	36.645	38.993
November	36.037	36.037	36.835	38.142	36.959	38.632	39.603
December	33.065	33.065	33.713	34.548	30.795	32.963	36.723

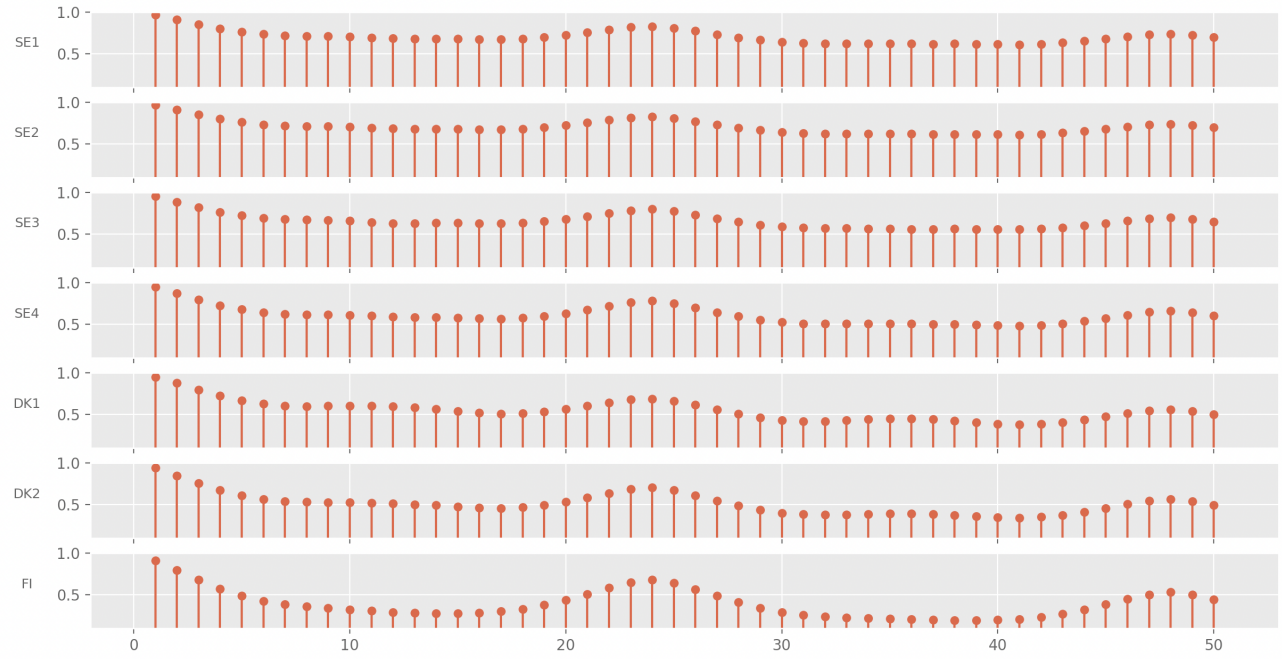


Figure 36: ACF-plot over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all non-NO-regions.

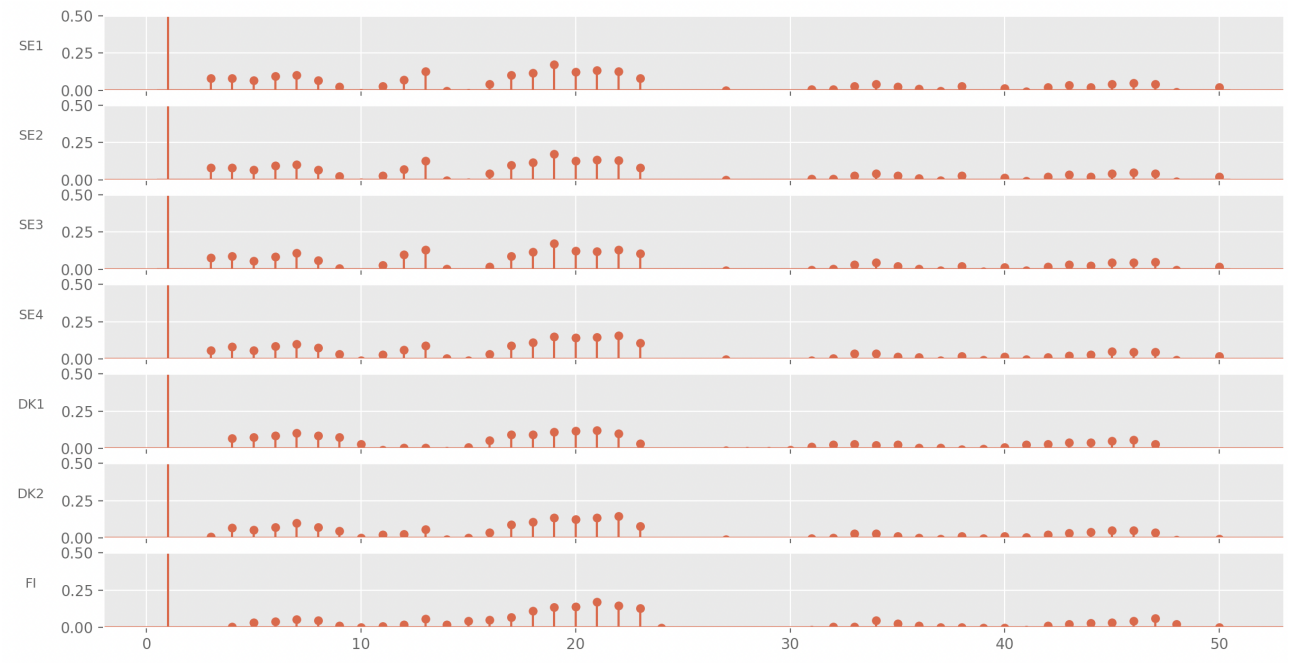


Figure 37: PACF-plot over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all non-NO-regions. The value at lag 1 is at approximately 0.99.

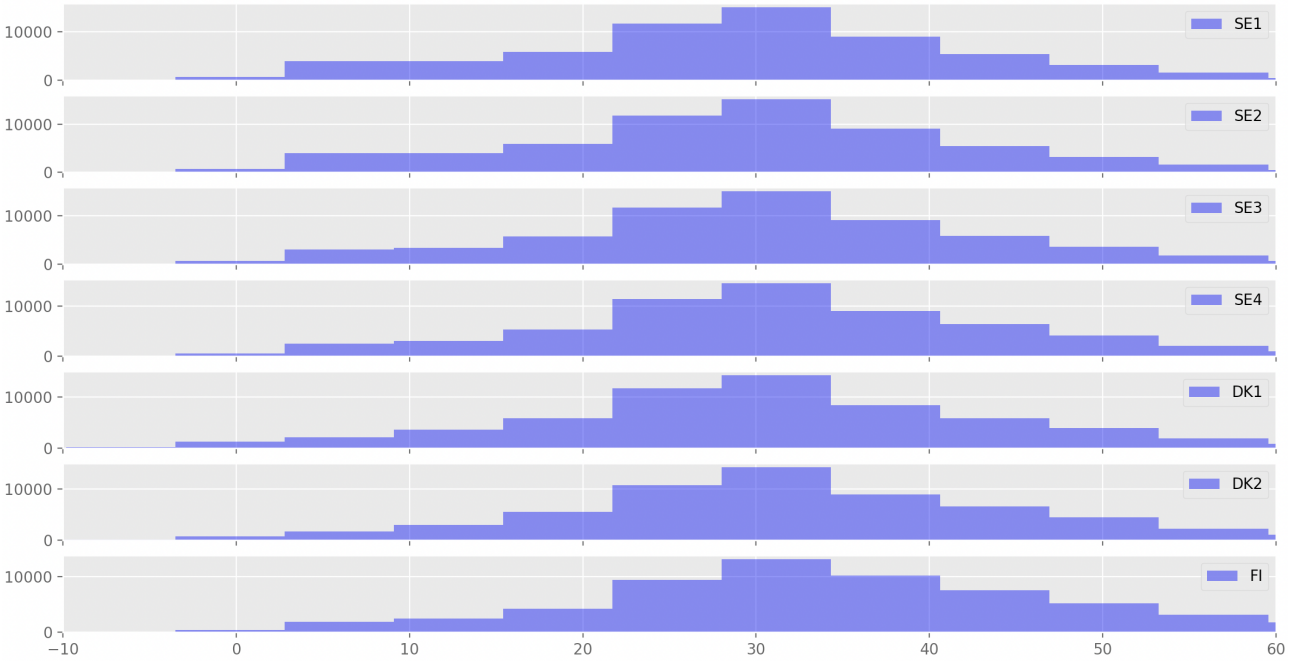


Figure 38: Price histogram over the in-sample data period (1 Jan 2014 - 31 Dec 2019) for all non-NO-regions. The histograms are cut off at €/MWh 60, and cover 99% of the distribution.

C Detailed Model Performances

This section provides the mean, median, standard deviation, min and max for all error metrics across all bidding regions for each implemented model. The results for the different models can be found in the following tables:

- Naive 7-day: Table 31
- ARIMA: Table 32
- SARIMA: Table 33
- DNN: Table 34
- ENTCN: Table 35
- LSTM: Table 36
- Stacked LSTM: Table 37
- GRU: Table 38
- Stacked GRU: Table 39
- Linear regression: Table 40
- Quadratic regression: Table 41

Table 31: Summary of Naive 7d model performance across all bidding areas and error metrics

Area	Metric	Mean	Median	Std	Min	Max
NO1	MAE	3.114	2.19	2.906	0.039	12.135
	SMAPE	38.553	27.697	31.529	2.602	147.676
	RMSE	3.967	2.539	4.048	0.054	18.104
	MAPE	48.597	28.095	60.29	2.675	353.977
NO2	MAE	3.118	2.203	2.899	0.039	12.135
	SMAPE	38.518	27.526	31.487	2.602	147.676
	RMSE	3.97	2.54	4.044	0.054	18.104
	MAPE	48.475	28.402	59.743	2.675	346.318
NO3	MAE	3.195	2.862	2.166	0.622	10.406
	SMAPE	40.789	34.604	21.574	8.384	104.891
	RMSE	3.832	3.373	2.594	0.712	12.033
	MAPE	50.437	36.457	44.412	8.588	247.859
NO4	MAE	2.642	2.182	1.698	0.484	8.605
	SMAPE	37.991	31.984	22.468	8.384	124.927
	RMSE	3.203	2.524	2.142	0.574	11.569
	MAPE	48.801	32.17	54.2	8.588	387.875
NO5	MAE	2.955	2.189	2.737	0.039	11.864
	SMAPE	38.153	26.157	31.646	2.602	147.676
	RMSE	3.547	2.539	3.238	0.054	13.335
	MAPE	48.032	26.182	60.275	2.675	353.977
SE1	MAE	6.503	5.576	4.233	0.75	19.404
	SMAPE	48.432	44.862	22.178	8.384	115.631
	RMSE	9.045	7.496	6.817	1.099	31.174
	MAPE	66.165	47.467	54.179	8.588	284.248
SE2	MAE	6.503	5.576	4.233	0.75	19.404
	SMAPE	48.432	44.888	22.178	8.384	115.631
	RMSE	9.045	7.496	6.817	1.099	31.174
	MAPE	66.165	47.467	54.179	8.588	284.248
SE3	MAE	12.439	10.245	8.429	1.981	44.716
	SMAPE	59.665	53.93	28.433	9.312	145.938
	RMSE	17.55	14.411	11.563	2.538	57.333
	MAPE	103.628	64.841	133.915	9.486	925.69
SE4	MAE	13.329	11.552	6.835	4.644	38.217
	SMAPE	58.472	56.049	23.109	16.745	131.327
	RMSE	18.103	15.603	9.003	6.88	46.143
	MAPE	109.658	68.783	123.306	16.72	764.836
DK1	MAE	13.236	11.741	6.597	3.626	44.358
	SMAPE	62.503	59.494	29.204	9.662	161.197
	RMSE	16.807	15.305	7.531	5.271	48.37
	MAPE	130.95	74.191	227.862	9.034	1921.01
DK2	MAE	13.406	11.559	6.67	4.888	38.357
	SMAPE	54.105	51.598	22.019	18.019	133.233
	RMSE	18.004	15.854	8.837	7.676	46.219
	MAPE	103.277	60.178	125.914	15.682	743.989
FI	MAE	12.845	10.463	7.049	4.152	37.972
	SMAPE	52.869	50.374	23.882	13.97	129.503
	RMSE	17.919	14.449	9.646	5.676	47.278
	MAPE	82.381	63.065	82.648	14.451	536.253

Table 32: Summary of ARIMA model performance across all bidding areas and error metrics

Area	Metric	Mean	Median	Std	Min	Max
NO1	MAE	2.213	1.441	2.217	0.022	11.576
	SMAPE	27.438	18.557	23.412	1.403	124.511
	RMSE	2.9	1.791	3.208	0.028	17.876
	MAPE	33.161	19.616	41.89	1.406	271.697
NO2	MAE	2.216	1.418	2.212	0.022	11.615
	SMAPE	27.335	18.807	23.151	1.385	124.389
	RMSE	2.899	1.76	3.205	0.028	17.87
	MAPE	33.134	20.801	41.756	1.389	271.187
NO3	MAE	2.459	1.945	1.997	0.259	19.021
	SMAPE	31.672	27.328	19.833	5.115	120.896
	RMSE	3.0	2.451	2.324	0.332	19.49
	MAPE	39.223	27.386	42.075	5.341	437.315
NO4	MAE	2.028	1.561	1.735	0.25	18.671
	SMAPE	28.977	23.96	20.552	4.3	116.295
	RMSE	2.482	1.828	2.044	0.293	19.096
	MAPE	35.838	24.84	41.849	4.228	372.067
NO5	MAE	2.056	1.438	2.015	0.022	9.455
	SMAPE	26.804	17.891	23.351	1.4	125.601
	RMSE	2.563	1.758	2.538	0.028	11.787
	MAPE	32.366	19.051	41.378	1.403	271.485
SE1	MAE	5.552	4.405	4.469	0.397	25.144
	SMAPE	40.983	35.687	24.192	5.048	149.107
	RMSE	7.437	5.592	6.469	0.499	29.718
	MAPE	60.089	37.229	60.51	5.188	373.792
SE2	MAE	5.552	4.405	4.468	0.397	25.117
	SMAPE	40.982	35.69	24.192	5.048	149.11
	RMSE	7.437	5.592	6.468	0.499	29.718
	MAPE	60.086	37.229	60.507	5.188	373.779
SE3	MAE	13.07	11.082	9.315	1.206	60.356
	SMAPE	61.913	57.251	29.696	5.626	200.0
	RMSE	16.994	13.649	11.488	1.953	62.317
	MAPE	116.064	72.479	151.682	5.43	1277.305
SE4	MAE	14.704	13.316	6.902	3.392	39.527
	SMAPE	62.508	58.135	25.337	17.427	168.495
	RMSE	18.462	16.069	8.807	4.987	47.127
	MAPE	135.347	80.449	144.574	15.854	741.256
DK1	MAE	12.971	10.95	6.597	3.049	42.44
	SMAPE	61.034	50.395	36.508	12.089	200.0
	RMSE	16.383	14.511	7.657	4.522	46.95
	MAPE	101.956	67.239	133.146	11.229	1340.57
DK2	MAE	14.219	12.344	7.16	3.547	46.369
	SMAPE	58.955	49.856	33.031	13.929	200.0
	RMSE	18.417	15.495	9.12	5.319	56.692
	MAPE	90.919	56.227	104.37	14.455	800.679
FI	MAE	16.795	15.099	7.581	5.49	68.725
	SMAPE	65.818	60.619	27.611	19.949	199.327
	RMSE	21.227	18.292	9.47	7.136	72.618
	MAPE	129.86	81.693	139.2	17.164	918.253

Table 33: Summary of SARIMA model performance across all bidding areas and error metrics

Area	Metric	Mean	Median	Std	Min	Max
NO1	MAE	2.304	1.356	2.415	0.043	17.599
	SMAPE	34.454	22.584	32.187	2.879	149.212
	RMSE	2.971	1.606	3.297	0.056	17.888
	MAPE	36.262	23.0	41.875	2.927	295.817
NO2	MAE	2.28	1.323	2.396	0.044	17.636
	SMAPE	33.82	22.512	31.898	2.71	148.657
	RMSE	2.937	1.575	3.272	0.057	17.926
	MAPE	35.655	22.613	41.549	2.718	295.496
NO3	MAE	3.03	2.515	2.1	0.313	9.786
	SMAPE	39.342	30.517	25.884	6.206	134.357
	RMSE	3.528	3.074	2.369	0.467	10.747
	MAPE	58.608	30.755	59.904	5.921	286.005
NO4	MAE	2.18	1.63	1.793	0.238	9.671
	SMAPE	34.961	25.075	31.034	3.254	163.597
	RMSE	2.622	2.044	2.104	0.293	10.134
	MAPE	37.251	24.729	38.719	3.29	332.585
NO5	MAE	2.1	1.313	2.126	0.044	14.502
	SMAPE	33.491	20.612	33.22	2.707	168.209
	RMSE	2.614	1.569	2.602	0.057	15.092
	MAPE	34.612	20.999	41.492	2.717	295.628
SE1	MAE	5.649	4.425	4.415	0.438	32.388
	SMAPE	45.912	34.861	33.546	6.164	185.478
	RMSE	7.463	5.672	6.216	0.521	33.295
	MAPE	59.936	40.771	58.843	6.079	336.477
SE2	MAE	5.649	4.423	4.414	0.437	32.407
	SMAPE	45.908	34.862	33.539	6.166	185.548
	RMSE	7.462	5.667	6.216	0.52	33.315
	MAPE	59.925	40.72	58.838	6.082	336.468
SE3	MAE	13.375	10.558	9.451	2.045	55.93
	SMAPE	69.051	57.712	35.893	10.504	178.069
	RMSE	17.065	13.216	11.306	2.919	61.815
	MAPE	120.536	74.531	158.376	9.659	1241.857
SE4	MAE	14.357	12.333	7.823	4.296	51.876
	SMAPE	67.004	58.816	35.241	16.62	179.332
	RMSE	17.867	15.127	9.238	5.891	55.289
	MAPE	130.442	75.102	141.962	16.636	850.137
DK1	MAE	11.727	10.122	4.922	3.472	37.747
	SMAPE	50.083	46.949	22.761	14.265	141.48
	RMSE	14.401	12.777	5.78	4.469	41.617
	MAPE	117.489	63.549	178.555	14.524	1384.826
DK2	MAE	11.906	10.776	4.858	5.229	31.97
	SMAPE	45.623	43.075	17.091	17.041	110.292
	RMSE	15.393	13.399	6.729	6.28	41.858
	MAPE	102.092	54.875	129.853	18.641	839.895
FI	MAE	12.775	11.417	5.37	5.754	36.529
	SMAPE	50.897	46.743	19.921	17.311	119.164
	RMSE	16.689	14.465	7.583	7.353	44.256
	MAPE	102.055	68.13	96.754	17.953	545.008

Table 34: Summary of DNN model performance across all bidding areas and error metrics

Area	Metric	Mean	Median	Std	Min	Max
NO1	MAE	7.315	5.43	6.301	0.478	25.589
	SMAPE	144.571	157.084	38.212	69.033	188.451
	RMSE	7.722	5.734	6.668	0.655	29.937
	MAPE	79.494	87.127	17.704	40.03	96.984
NO2	MAE	7.317	5.408	6.281	0.478	25.589
	SMAPE	145.037	157.084	37.728	69.033	188.451
	RMSE	7.72	5.734	6.654	0.655	29.937
	MAPE	79.785	87.113	17.421	40.03	96.984
NO3	MAE	7.625	6.061	5.124	1.131	20.005
	SMAPE	155.093	161.115	25.198	96.647	186.86
	RMSE	7.997	6.221	5.2	1.285	20.062
	MAPE	85.221	88.623	10.158	58.8	96.554
NO4	MAE	7.024	5.791	4.569	1.077	20.005
	SMAPE	153.917	160.77	25.094	94.465	186.86
	RMSE	7.334	6.058	4.621	1.285	20.062
	MAPE	84.815	88.474	10.263	57.34	96.554
NO5	MAE	7.188	5.338	6.134	0.478	23.348
	SMAPE	144.092	157.084	38.51	69.033	188.156
	RMSE	7.495	5.714	6.268	0.655	25.267
	MAPE	79.269	87.113	17.853	40.03	96.907
SE1	MAE	12.963	11.055	8.421	2.104	35.106
	SMAPE	168.285	172.862	17.538	119.89	191.982
	RMSE	14.473	11.847	9.618	2.429	39.579
	MAPE	90.149	92.051	6.447	70.323	97.934
SE2	MAE	12.963	11.055	8.421	2.104	35.106
	SMAPE	168.285	172.862	17.538	119.89	191.982
	RMSE	14.474	11.847	9.618	2.429	39.579
	MAPE	90.149	92.051	6.447	70.323	97.934
SE3	MAE	20.251	16.642	11.414	3.172	54.008
	SMAPE	173.41	176.491	14.939	125.757	192.596
	RMSE	25.006	20.67	13.546	3.967	66.395
	MAPE	91.727	93.184	5.575	72.597	98.095
SE4	MAE	24.962	22.405	11.762	5.811	61.236
	SMAPE	177.798	179.664	12.171	141.412	193.66
	RMSE	29.772	26.825	12.946	8.088	70.166
	MAPE	93.108	94.228	4.576	77.933	98.328
DK1	MAE	24.541	23.018	9.635	6.512	52.535
	SMAPE	180.52	182.371	12.024	108.651	193.16
	RMSE	28.365	26.846	9.724	13.097	56.643
	MAPE	94.007	94.961	4.923	61.017	98.244
DK2	MAE	27.76	25.23	10.769	11.571	61.483
	SMAPE	183.158	184.716	8.456	153.881	193.744
	RMSE	32.055	28.544	11.989	16.111	70.334
	MAPE	94.959	95.75	3.246	81.584	98.351
FI	MAE	27.317	24.167	10.249	8.157	61.483
	SMAPE	179.495	181.026	10.419	143.865	193.442
	RMSE	32.856	29.942	11.557	12.536	71.877
	MAPE	93.788	94.687	3.898	80.621	98.311

Table 35: Summary of ENTCN model performance across all bidding areas and error metrics

Area	Metric	Mean	Median	Std	Min	Max
NO1	MAE	2.419	1.539	2.334	0.414	19.721
	SMAPE	31.466	26.524	20.805	4.398	115.462
	RMSE	3.071	1.835	3.167	0.503	19.829
	MAPE	40.539	31.945	43.748	4.444	290.535
NO2	MAE	2.411	1.502	2.321	0.413	19.096
	SMAPE	31.17	26.32	20.65	3.941	115.271
	RMSE	3.071	1.799	3.168	0.504	19.177
	MAPE	40.369	31.32	43.798	4.003	290.233
NO3	MAE	2.603	1.912	2.019	0.49	16.885
	SMAPE	32.73	27.856	19.193	5.451	113.545
	RMSE	3.091	2.418	2.289	0.559	17.445
	MAPE	43.43	30.318	44.636	5.653	388.231
NO4	MAE	2.179	1.657	1.771	0.456	17.446
	SMAPE	30.406	25.098	20.204	4.367	122.213
	RMSE	2.593	1.926	2.021	0.503	17.855
	MAPE	40.87	26.575	47.121	4.236	366.573
NO5	MAE	2.227	1.447	1.957	0.404	10.027
	SMAPE	30.633	25.608	20.721	3.989	115.57
	RMSE	2.724	1.76	2.413	0.495	11.199
	MAPE	39.664	31.109	43.361	4.047	292.169
SE1	MAE	5.944	4.598	5.521	0.671	42.451
	SMAPE	42.364	35.814	24.984	6.88	143.379
	RMSE	7.688	5.869	6.995	0.833	43.574
	MAPE	63.418	38.647	67.593	6.676	449.436
SE2	MAE	5.948	4.616	5.521	0.667	42.452
	SMAPE	42.329	35.659	24.916	6.849	142.043
	RMSE	7.692	5.894	6.993	0.844	43.581
	MAPE	63.58	38.508	67.761	6.648	448.904
SE3	MAE	13.03	10.984	9.282	1.564	60.836
	SMAPE	61.691	57.269	28.935	7.707	143.754
	RMSE	16.687	13.565	11.24	1.953	63.269
	MAPE	116.319	72.736	145.781	7.537	1039.517
SE4	MAE	14.501	13.1	7.744	2.565	55.351
	SMAPE	61.822	58.638	28.008	11.636	156.009
	RMSE	18.134	15.577	9.171	3.664	57.501
	MAPE	130.856	73.613	147.684	11.026	869.975
DK1	MAE	14.035	12.151	7.17	3.343	39.833
	SMAPE	60.479	52.403	34.185	11.269	200.0
	RMSE	17.109	15.38	7.994	4.492	45.114
	MAPE	119.908	70.586	173.023	11.591	1477.436
DK2	MAE	14.749	13.039	7.424	3.348	47.127
	SMAPE	56.256	50.997	28.375	14.161	200.0
	RMSE	18.312	16.172	8.645	4.774	51.655
	MAPE	112.276	62.436	132.475	14.126	858.815
FI	MAE	15.641	13.686	8.009	4.536	75.934
	SMAPE	61.102	55.999	25.41	15.98	143.538
	RMSE	19.649	16.887	9.5	5.962	77.299
	MAPE	120.962	73.901	129.246	17.412	817.257

Table 36: Summary of LSTM model performance across all bidding areas and error metrics

Area	Metric	Mean	Median	Std	Min	Max
NO1	MAE	7.916	6.001	6.31	1.02	26.2
	SMAPE	186.101	190.527	11.21	161.439	197.68
	RMSE	8.262	6.276	6.7	1.068	30.465
	MAPE	97.204	98.284	2.451	91.107	99.593
NO2	MAE	7.919	6.001	6.29	1.02	26.2
	SMAPE	186.261	190.527	11.055	162.056	197.68
	RMSE	8.262	6.275	6.684	1.068	30.465
	MAPE	97.243	98.282	2.412	91.163	99.593
NO3	MAE	8.233	6.673	5.127	1.743	20.617
	SMAPE	189.569	191.601	6.691	173.129	197.363
	RMSE	8.555	6.819	5.223	1.798	20.664
	MAPE	98.011	98.485	1.367	94.521	99.535
NO4	MAE	7.633	6.402	4.572	1.689	20.617
	SMAPE	189.282	191.442	6.725	172.338	197.363
	RMSE	7.893	6.643	4.644	1.765	20.664
	MAPE	97.953	98.421	1.382	94.288	99.535
NO5	MAE	7.79	5.948	6.143	1.02	23.96
	SMAPE	185.947	190.527	11.311	161.439	197.635
	RMSE	8.039	6.276	6.306	1.068	25.825
	MAPE	97.168	98.265	2.475	91.107	99.586
SE1	MAE	13.567	11.666	8.426	2.654	35.718
	SMAPE	192.96	194.265	4.337	180.18	198.4
	RMSE	15.013	12.446	9.613	2.932	40.104
	MAPE	98.691	98.989	0.868	95.941	99.724
SE2	MAE	13.568	11.666	8.426	2.654	35.718
	SMAPE	192.96	194.265	4.337	180.18	198.4
	RMSE	15.013	12.446	9.613	2.932	40.104
	MAPE	98.691	98.989	0.868	95.941	99.724
SE3	MAE	20.855	17.232	11.42	3.781	54.62
	SMAPE	194.187	195.116	3.663	182.084	198.572
	RMSE	25.482	21.182	13.551	4.466	66.891
	MAPE	98.931	99.128	0.737	96.266	99.754
SE4	MAE	25.566	23.016	11.765	6.422	61.847
	SMAPE	195.178	195.776	2.968	185.61	198.731
	RMSE	30.26	27.308	12.964	8.463	70.694
	MAPE	99.118	99.258	0.6	96.925	99.83
DK1	MAE	25.105	23.574	9.669	7.016	53.154
	SMAPE	195.799	196.418	3.293	172.534	198.733
	RMSE	28.873	27.356	9.751	13.549	57.19
	MAPE	99.222	99.376	0.726	93.099	99.922
DK2	MAE	28.343	25.837	10.786	12.108	62.089
	SMAPE	196.394	196.852	2.065	187.822	198.78
	RMSE	32.566	29.096	11.996	16.594	70.847
	MAPE	99.346	99.453	0.429	97.142	99.942
FI	MAE	28.005	24.642	10.356	8.718	62.136
	SMAPE	195.81	196.311	2.398	188.031	198.767
	RMSE	33.492	30.523	11.656	13.076	72.414
	MAPE	99.252	99.368	0.476	97.454	99.795

Table 37: Summary of Stacked LSTM model performance across all bidding areas and error metrics

Area	Metric	Mean	Median	Std	Min	Max
NO1	MAE	7.232	4.786	5.575	2.273	24.679
	SMAPE	128.32	119.78	19.626	103.894	168.927
	RMSE	8.04	5.611	5.923	2.826	29.453
	MAPE	104.258	88.733	33.801	73.071	203.846
NO2	MAE	7.222	4.786	5.653	2.131	24.789
	SMAPE	129.261	121.203	20.822	103.545	170.733
	RMSE	7.989	5.602	5.999	2.631	29.516
	MAPE	100.606	88.708	28.376	73.041	181.87
NO3	MAE	7.255	5.405	4.714	2.152	19.205
	SMAPE	128.932	124.335	20.995	101.048	167.619
	RMSE	8.0	6.104	4.731	2.671	19.442
	MAPE	88.039	87.241	10.718	73.129	126.967
NO4	MAE	6.665	5.198	4.146	2.146	19.204
	SMAPE	127.253	123.531	19.673	100.98	167.619
	RMSE	7.361	6.013	4.152	2.671	19.441
	MAPE	87.755	86.184	11.446	73.121	131.496
NO5	MAE	7.119	4.776	5.473	2.128	22.548
	SMAPE	129.281	120.511	20.743	102.003	170.196
	RMSE	7.786	5.582	5.564	2.622	24.698
	MAPE	101.825	89.367	29.241	73.042	189.325
SE1	MAE	12.554	10.537	8.304	2.403	34.553
	SMAPE	147.478	150.41	20.806	107.237	181.995
	RMSE	14.29	11.613	9.453	2.978	39.246
	MAPE	88.534	89.597	5.934	74.597	106.805
SE2	MAE	12.554	10.537	8.305	2.403	34.553
	SMAPE	147.478	150.41	20.806	107.237	181.995
	RMSE	14.291	11.613	9.454	2.978	39.246
	MAPE	88.534	89.597	5.934	74.597	106.805
SE3	MAE	19.826	16.441	11.329	3.233	53.455
	SMAPE	155.868	157.536	17.358	112.433	183.887
	RMSE	24.725	20.629	13.491	4.225	66.029
	MAPE	90.762	91.281	4.891	76.315	106.848
SE4	MAE	24.521	22.082	11.71	5.407	60.682
	SMAPE	162.781	163.011	14.748	120.254	186.25
	RMSE	29.432	26.484	12.909	7.898	69.7
	MAPE	92.64	93.266	4.699	76.368	111.59
DK1	MAE	24.211	22.747	9.545	7.264	52.078
	SMAPE	168.266	169.351	12.141	125.125	186.245
	RMSE	28.06	26.531	9.687	13.001	56.224
	MAPE	93.98	94.517	4.498	82.113	123.484
DK2	MAE	27.542	25.063	10.743	11.548	61.211
	SMAPE	173.932	174.379	9.314	144.919	189.027
	RMSE	31.871	28.297	11.986	16.027	70.067
	MAPE	94.924	95.273	2.409	88.305	105.436
FI	MAE	26.667	23.377	10.257	8.311	60.643
	SMAPE	160.543	159.462	12.183	125.406	183.51
	RMSE	32.478	29.636	11.609	12.291	71.339
	MAPE	92.122	92.203	4.289	79.976	115.036

Table 38: Summary of GRU model performance across all bidding areas and error metrics

Area	Metric	Mean	Median	Std	Min	Max
NO1	MAE	6.957	4.966	6.16	0.688	25.088
	SMAPE	125.979	134.301	40.964	60.998	180.861
	RMSE	7.397	5.308	6.529	0.869	29.466
	MAPE	74.89	78.326	15.925	47.623	94.949
NO2	MAE	6.953	4.966	6.146	0.688	25.088
	SMAPE	126.188	134.301	40.699	60.998	180.861
	RMSE	7.391	5.308	6.518	0.87	29.466
	MAPE	74.884	78.326	15.931	47.623	94.949
NO3	MAE	7.184	5.58	5.082	0.935	19.524
	SMAPE	134.864	140.727	31.854	71.7	178.397
	RMSE	7.603	5.793	5.14	1.139	19.587
	MAPE	77.74	81.31	13.393	49.166	94.263
NO4	MAE	6.556	5.331	4.52	0.896	19.491
	SMAPE	131.927	137.788	31.619	70.099	177.884
	RMSE	6.915	5.653	4.556	1.09	19.555
	MAPE	76.64	80.28	13.431	48.202	94.102
NO5	MAE	6.834	4.853	5.988	0.688	22.848
	SMAPE	125.448	134.301	41.234	60.998	180.44
	RMSE	7.174	5.282	6.117	0.868	24.742
	MAPE	74.675	78.326	16.056	47.623	94.837
SE1	MAE	12.506	10.576	8.403	1.796	34.627
	SMAPE	152.689	158.14	23.895	93.406	186.706
	RMSE	14.053	11.422	9.598	2.151	39.182
	MAPE	84.925	87.579	9.132	59.931	96.592
SE2	MAE	12.506	10.576	8.404	1.796	34.627
	SMAPE	152.689	158.14	23.896	93.406	186.706
	RMSE	14.053	11.422	9.598	2.151	39.182
	MAPE	84.925	87.579	9.132	59.931	96.592
SE3	MAE	19.793	16.163	11.399	2.837	53.529
	SMAPE	161.176	165.067	19.647	105.108	188.143
	RMSE	24.586	20.405	13.533	3.647	65.936
	MAPE	87.807	89.439	7.326	64.601	96.964
SE4	MAE	24.504	21.93	11.753	5.334	60.756
	SMAPE	167.679	168.92	15.911	126.08	190.046
	RMSE	29.34	26.374	12.925	7.612	69.689
	MAPE	90.027	90.969	5.79	73.29	97.444
DK1	MAE	24.179	22.711	9.6	6.485	52.121
	SMAPE	172.198	174.534	14.565	96.95	189.834
	RMSE	28.005	26.396	9.696	12.873	56.165
	MAPE	91.689	92.768	5.184	64.411	97.397
DK2	MAE	27.348	24.801	10.753	11.176	61.045
	SMAPE	175.138	176.457	10.798	139.587	190.534
	RMSE	31.66	28.135	11.97	15.785	69.889
	MAPE	92.651	93.383	3.878	79.349	97.554
FI	MAE	26.919	23.534	10.34	7.814	61.028
	SMAPE	171.114	172.184	12.821	128.517	190.287
	RMSE	32.548	29.644	11.639	12.132	71.431
	MAPE	91.26	92.11	4.707	75.036	97.527

Table 39: Summary of Stacked GRU model performance across all bidding areas and error metrics

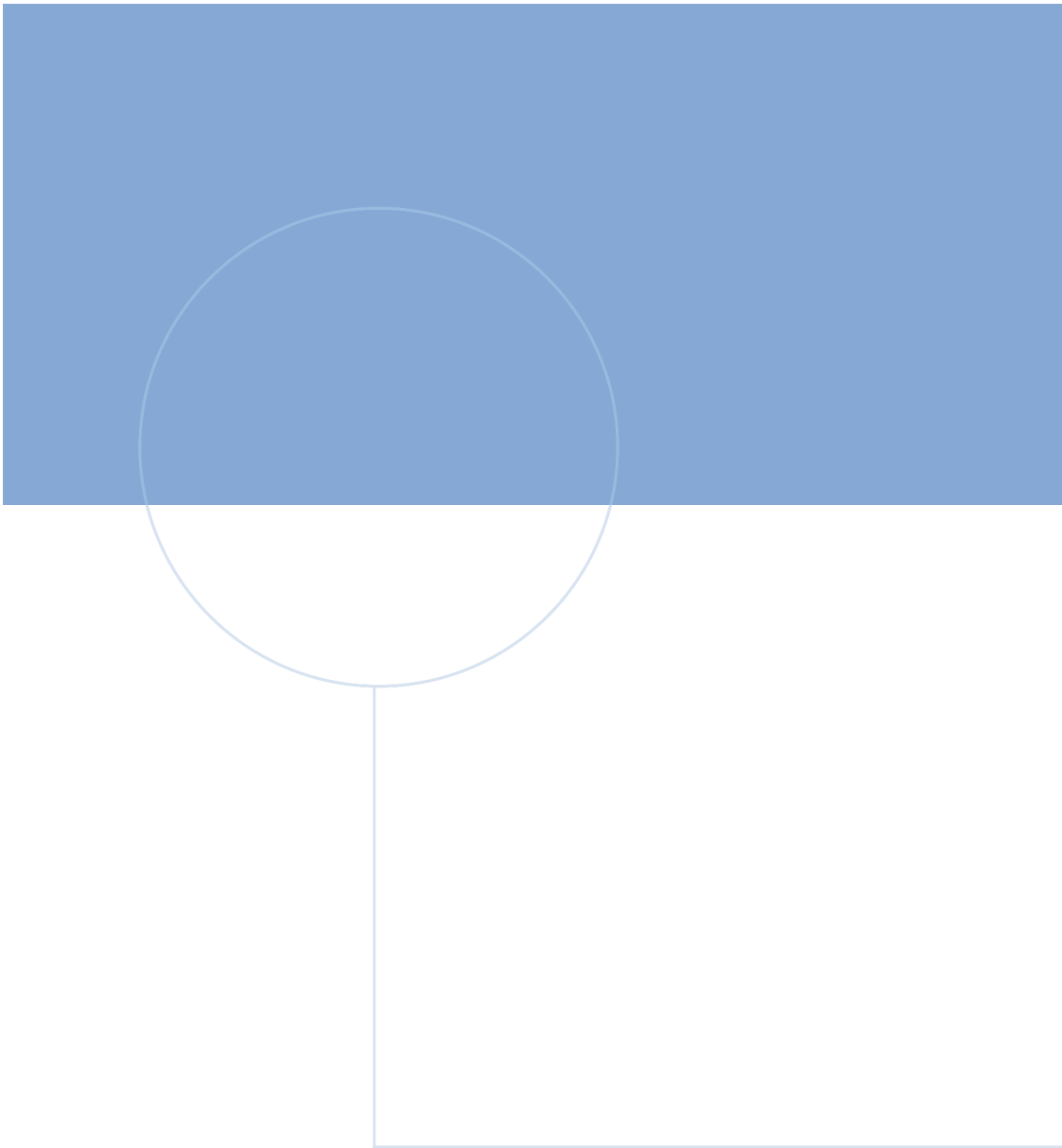
Area	Metric	Mean	Median	Std	Min	Max
NO1	MAE	7.073	5.544	3.445	4.509	20.863
	SMAPE	106.796	101.869	17.649	83.207	139.699
	RMSE	8.54	6.916	3.983	5.564	26.779
	MAPE	163.634	87.073	132.119	59.388	479.089
NO2	MAE	7.058	5.517	3.451	4.513	20.86
	SMAPE	106.381	101.123	17.444	83.229	139.421
	RMSE	8.524	6.897	3.989	5.568	26.776
	MAPE	161.088	87.154	129.831	58.967	465.626
NO3	MAE	6.745	5.454	2.434	4.63	15.041
	SMAPE	99.572	98.512	10.979	82.299	125.772
	RMSE	8.143	6.813	2.623	5.723	16.115
	MAPE	117.752	79.09	70.627	59.101	324.485
NO4	MAE	6.329	5.513	1.951	4.722	14.95
	SMAPE	97.533	94.901	10.97	82.396	123.959
	RMSE	7.714	6.891	2.128	5.829	16.061
	MAPE	120.035	80.765	74.568	58.619	330.43
NO5	MAE	7.021	5.618	3.173	4.562	18.569
	SMAPE	107.445	102.078	18.082	83.352	141.448
	RMSE	8.397	7.016	3.367	5.641	21.693
	MAPE	167.303	89.902	135.463	59.682	486.835
SE1	MAE	10.62	7.857	6.775	4.469	30.399
	SMAPE	105.567	101.291	15.484	83.415	146.495
	RMSE	12.937	9.569	8.164	5.539	35.948
	MAPE	90.998	77.358	32.929	60.806	207.493
SE2	MAE	10.621	7.857	6.776	4.469	30.399
	SMAPE	105.568	101.291	15.486	83.415	146.495
	RMSE	12.938	9.569	8.164	5.539	35.948
	MAPE	90.998	77.358	32.929	60.806	207.493
SE3	MAE	17.409	14.347	10.26	4.882	49.551
	SMAPE	116.708	116.213	16.565	90.304	151.782
	RMSE	22.561	18.445	12.797	6.158	62.702
	MAPE	91.687	81.703	28.408	62.728	205.171
SE4	MAE	21.69	18.76	10.879	5.965	56.647
	SMAPE	124.872	124.369	17.251	93.131	161.766
	RMSE	26.865	23.836	12.361	8.036	66.163
	MAPE	91.312	83.718	24.896	64.454	208.534
DK1	MAE	21.504	20.137	8.788	9.018	48.282
	SMAPE	130.308	131.287	14.689	100.341	159.775
	RMSE	25.543	24.226	9.217	12.262	52.713
	MAPE	89.586	83.835	27.237	66.137	288.888
DK2	MAE	23.961	21.211	10.186	9.957	56.757
	SMAPE	129.53	129.557	14.69	98.419	161.367
	RMSE	28.755	24.889	11.615	13.848	66.211
	MAPE	87.009	81.567	19.927	67.753	197.99
FI	MAE	23.77	20.574	9.693	8.53	56.779
	SMAPE	127.248	125.35	13.887	100.064	159.115
	RMSE	29.862	27.239	11.32	10.949	68.214
	MAPE	87.812	82.661	18.555	66.831	166.979

Table 40: Summary of Linear regression model performance across all bidding areas and error metrics

Area	Metric	Mean	Median	Std	Min	Max
NO1	MAE	4.139	3.959	1.698	0.794	11.844
	SMAPE	57.737	47.451	38.453	4.919	147.719
	RMSE	4.934	4.496	2.511	1.015	23.324
	MAPE	128.571	69.145	132.698	5.044	617.129
NO2	MAE	3.865	3.71	1.579	0.849	10.702
	SMAPE	56.143	44.436	38.351	5.258	145.421
	RMSE	4.607	4.165	2.503	1.057	22.958
	MAPE	119.646	62.936	124.297	5.209	592.803
NO3	MAE	5.973	5.584	2.689	1.075	16.057
	SMAPE	62.147	57.683	34.457	7.106	138.161
	RMSE	6.827	6.258	3.166	1.467	22.096
	MAPE	136.233	94.72	124.004	6.765	737.252
NO4	MAE	4.041	3.377	2.346	1.071	14.283
	SMAPE	47.738	40.718	28.274	5.981	124.792
	RMSE	4.819	3.989	2.909	1.231	18.844
	MAPE	89.838	56.947	89.112	6.201	581.89
NO5	MAE	3.653	3.586	1.469	0.853	8.49
	SMAPE	55.265	43.166	37.728	5.657	141.521
	RMSE	4.247	4.009	1.933	1.004	19.463
	MAPE	116.481	60.208	120.058	5.517	555.119
SE1	MAE	6.688	6.319	3.148	1.47	17.989
	SMAPE	49.524	44.21	24.531	7.936	128.397
	RMSE	8.602	7.459	5.113	1.966	27.917
	MAPE	89.964	63.602	78.778	8.192	495.27
SE2	MAE	6.691	6.324	3.149	1.472	17.986
	SMAPE	49.537	44.248	24.554	7.941	128.475
	RMSE	8.606	7.459	5.113	1.967	27.911
	MAPE	90.052	63.627	78.897	8.197	496.112
SE3	MAE	10.482	8.484	6.465	1.511	42.025
	SMAPE	53.457	50.34	22.071	8.815	109.267
	RMSE	14.207	10.851	9.64	2.09	55.637
	MAPE	81.361	56.379	64.842	9.094	359.483
SE4	MAE	12.763	11.127	6.332	3.267	41.593
	SMAPE	57.248	53.737	20.57	15.118	109.204
	RMSE	16.296	13.877	8.615	4.273	51.24
	MAPE	81.243	57.203	62.497	15.958	344.282
DK1	MAE	12.484	11.493	4.895	3.865	34.895
	SMAPE	57.054	52.09	21.274	14.183	134.063
	RMSE	15.084	13.796	5.816	5.347	38.388
	MAPE	82.113	54.445	94.995	14.704	764.671
DK2	MAE	13.145	11.566	5.635	3.761	39.584
	SMAPE	53.691	50.695	18.377	15.533	107.778
	RMSE	16.595	13.897	7.718	5.026	49.21
	MAPE	69.67	47.837	58.152	16.264	327.819
FI	MAE	12.085	10.864	4.756	5.463	33.429
	SMAPE	49.294	46.407	18.57	18.159	115.258
	RMSE	15.789	13.47	7.375	6.884	46.456
	MAPE	89.048	60.167	78.247	17.325	440.213

Table 41: Summary of Quadratic regression model performance across all bidding areas and error metrics

Area	Metric	Mean	Median	Std	Min	Max
NO1	MAE	6.959	3.565	18.694	0.8	259.126
	SMAPE	71.76	52.957	53.451	4.929	196.295
	RMSE	12.295	4.474	43.213	1.029	515.2
	MAPE	205.67	57.508	612.187	4.958	6050.477
NO2	MAE	4.883	3.278	6.501	0.594	74.977
	SMAPE	68.114	48.06	52.811	4.932	200.0
	RMSE	7.42	4.067	14.261	0.71	158.401
	MAPE	139.283	51.32	283.971	4.939	3392.794
NO3	MAE	7.771	4.182	19.007	0.726	256.118
	SMAPE	61.289	52.83	38.788	5.331	182.861
	RMSE	13.437	5.192	45.054	0.912	532.948
	MAPE	168.701	65.396	527.232	5.435	6766.744
NO4	MAE	6.946	3.397	19.151	0.85	260.805
	SMAPE	64.652	47.991	46.556	5.592	193.655
	RMSE	12.423	4.399	45.689	1.057	541.913
	MAPE	149.024	56.102	481.121	5.49	5903.999
NO5	MAE	4.551	2.97	6.27	0.585	70.068
	SMAPE	65.304	45.988	51.029	4.911	200.0
	RMSE	6.951	3.638	14.172	0.706	156.149
	MAPE	133.509	50.414	288.33	4.893	3516.649
SE1	MAE	9.931	6.187	18.965	1.191	256.401
	SMAPE	58.647	52.784	32.084	5.824	168.496
	RMSE	16.465	7.903	44.66	1.451	530.899
	MAPE	130.484	61.342	465.353	5.911	6862.598
SE2	MAE	9.932	6.206	18.968	1.19	256.39
	SMAPE	58.48	52.798	31.891	5.822	168.451
	RMSE	16.478	7.9	44.667	1.45	530.926
	MAPE	130.581	61.326	465.395	5.908	6862.228
SE3	MAE	14.657	9.457	20.357	1.416	266.673
	SMAPE	67.08	60.95	32.235	6.777	165.409
	RMSE	22.672	12.199	44.768	2.023	532.086
	MAPE	109.629	66.175	319.459	6.923	4554.73
SE4	MAE	17.504	12.342	20.069	2.372	263.782
	SMAPE	71.405	66.434	31.537	10.085	157.01
	RMSE	25.286	16.453	43.885	3.756	526.593
	MAPE	103.176	68.976	261.65	10.022	3458.0
DK1	MAE	14.865	12.936	8.436	2.651	72.881
	SMAPE	64.649	60.399	26.75	10.997	147.932
	RMSE	19.361	16.062	15.954	4.653	170.689
	MAPE	97.869	64.42	127.761	10.092	1171.522
DK2	MAE	17.149	12.599	19.487	3.075	263.166
	SMAPE	60.532	55.704	24.717	12.724	147.014
	RMSE	24.519	16.11	42.796	4.811	513.676
	MAPE	88.137	56.547	121.062	12.759	1591.188
FI	MAE	16.037	11.294	24.65	4.641	344.638
	SMAPE	52.593	50.446	21.885	16.393	161.1
	RMSE	23.587	14.749	47.847	6.259	576.733
	MAPE	111.149	70.825	145.201	16.464	1666.854



NTNU

Kunnskap for en bedre verden