DTU Management
Transport division

Electricity Price Forecasting using Probabilistic Deep Learning

Four-day-ahead forecasting of electricity spot prices

Master Thesis

Electricity Price Forecasting using Probabilistic Deep Learning
Four-day-ahead forecasting of electricity spot prices

Master Thesis
June, 2022

By
Marcos Ivorra Peleguer

Cover photo: Vibeke Hempler, 2012
Published by: DTU, Transport division, Bygningstorvet 116b, 2800 Kgs. Lyngby Denmark
www.transport.man.dtu.dk

Approval

This thesis has been prepared over six months at the Section for Machine Learning for Smart Mobility, Transport Division, DTU Management, at the Technical University of Denmark, DTU, in partial fulfilment for the degree Master of Science in Engineering, MSc Eng.

It is assumed that the reader has a basic knowledge in the areas of statistics and machine learning.

Marcos Ivorra Peleguer - s202947

....

Signature

....

Date

23/06/2022

”It is difficult to make predictions, especially about the future.” - Karl Kristian Steincke

Abstract

The explosion in recent years of the Electric Vehicle (EV onwards) market has opened possibilities for new use cases of advanced analytics and machine learning. Such is the case that there is a new wave of startups and small companies growing out of business needs or data deserts related to the Energy markets, not only in the EV domain. In this context, several companies are trying to predict the electricity spot price, or simply make use of these forecasts to optimise production, allocate resources or make better purchasing decisions. Within these companies is Monta, a danish startup in the EV chargers software sector, this thesis aims to forecast the spot price a week ahead to improve the smart charging service of the company. Smart charging opens up the possibility of charging along a time window, instead of charging immediately after connecting the EV. Therefore, the price forecasts derived from this research will augment the horizon in which to decide to charge a car from the very moment it is connected to the charging point. The forecasting is conducted with a probabilistic approach, using a Deep Learning-based algorithm. The algorithm has been iteratively improved starting from simple time-averaging baselines to ensure a positive tradeoff between complexity and results - accuracy of the model, hence revenue. The models are highly influenced by the (mal)functioning of the electricity markets and, as such, the seasonalities and main drivers of the price variations are deeply studied to better understand the series of study. In order to evaluate the models developed, a frame work is developed so that each model is tested in 20 different time windows, averaging the errors across the different windows and comparing the results to determine whether one model is better than the other. The ever-growing aspirations of the company make it a must that the model properly generalizes to all regions where they are present, as to be able to integrate the model within their daily operations and offer the smart charging seamlessly across regions. This has been taken into account when evaluating the models, as well as when implementing the designed algorithm in code, writing standalone scripts that can be easy to integrate with the optimization algorithm Monta uses for the Smart Charging. Finally the method is evaluated in terms of total cash saved. This is measured by determining the difference in savings between charging immediately and charging when the forecasts say to do so.

Index terms: Electricity price forecasting, Temporal validation, LSTM, Probabilistic LSTM.

Code: https://github.com/Marcoscos/Monta_project

Glossary

EV: Electric Vehicle

Smart Charging: Smart charging refers to a charging system where electric vehicles, charging stations and charging operators share data connections. Through smart charging, the charging stations may monitor, manage, and restrict the use of charging devices to optimize energy consumption. [1]

DL:Deep Learning. Group of techniques that mimic the structure of organic neurons to perform mathematical computations in the field of Artificial Intelligence.

SotA: Of "State of the Art", this term refers to the latest techniques that have been proven to outperform, according to the accepted benchmarking indicators, the previously discovered techniques.

CP: Charging point. Charger of the EV.

V2G: Vehicle to grid.

HPV: Hydrogen-powered vehicles.

DTU: Technical University of Denmark.

Meta-Observer: Concept coined in [2], refers to a third person that observes an act of communication in which an observer is evaluating a user. This third person then provides feedback based on the evaluation conducted by the observer.

Root Mean Squared Error (RMSE): Quantification of the error of data estimations with regards to real observations. The main benefit from using RMSE over other measures is that it is a good estimation of the standard deviation of the errors. In the context of this work, it serves as a heuristic for training models and evaluate trained models for accuracy.

Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test: Metric used for testing a null hypothesis that an observable time series is stationary around a deterministic trend (i.e. trend-stationary) against the alternative of a unit root. Note that it is against the unit root, hence if the null hypothesis is not true, the series tested is stationary.

Augmented-Dickey-Fuller (ADF) test: augmented Dickey–Fuller test (ADF) tests the null hypothesis that a unit root is present in a time series sample. The alternative hypothesis is different depending on which version of the test is used, but is usually stationarity or trend-stationarity. It is an augmented version of the Dickey–Fuller test for a larger and more complicated set of time series models. The ADF statistic is a negative number. The more negative it is, the stronger the rejection of the hypothesis that there is a unit root at some level of confidence (p-value).

Seasonal difference: Transformation applied to a time series to make it stationary. More specifically, this transformation is applied when the series has a seasonal component.

$$Diff = \Delta^d \Delta_s^D y_t \tag{1}$$

Seasonal AutoRegressive Integrated Moving Average (SARIMA): Model of the family of classical statistical models.

Acknowledgements

Marcos Ivorra Peleguer, MSc Business Analytics Engineering, DTU
Creator of the research project

Frederik Boe Hüttel, P.hD, Machine Learning for Smart Mobility, DTU
For helping me order my train of thought, have the clarity of deciding what paths to follow and motivating me along the way.

Francisco Camara Pereira, Associate Professor, P.hD., Machine Learning for Smart Mobility, DTU
For having the meta-observer perspective and swiftly directing the project.

Casper H. Rasmussen, CEO CTO, MONTA
For providing the resources, industry expertise, environment and set up for this project to take place.

Contents

List of Figures

List of Tables

1 Introduction

This document contains the research conducted with the aim of short-term forecasting of the electricity spot price of the day-ahead market. In particular, the aim is to obtain a four-day-ahead forecast of the Spot price of the Production of electricity in the day-ahead market.

Forecasting of electricity prices has been a hot topic for several decades, in the EU especially since the definition of the common energy market. There is widespread attention for the opportunities it opens to stakeholders across the whole value chain: from producers, operators, to general utilities companies, and the day-to-day user. In this context, the problem of short-term and very short-term forecasting are the ones taking the spotlight, whereas short to mid-term forecasting and long-term forecasting of electricity prices, and therefore, energy generation, has been sometimes forgotten due to the complexity and extra resources needed. In the last years, there has been a growing interest in using more complex models, generally Deep Learning techniques, which have proven to be very accurate for demand forecasting and other applications. The objective of this thesis is precisely to utlise these techniques to obtain a good forecast of the spot price, bearing in mind the trade-off between the potential extra accuracy of this models and both the extra resources needed as well as the explainability sacrificed for using them.

From a technical point of view, traditional engineering-intensive sectors, such as the Energy or Supply Chain sector have been slower to adopt the latest forecasting techniques due to the fact that these kind of sectors are well versed in classical statistical techniques and classical time series forecasting methods, which caused a resistance to adopt new methods that would slightly improve the accuracy but would highly increase the complexity of the analyses. However, even these fields are shifting towards newer forecasting techniques and, especially the energy sector [3] [4] [5], is looking to academia to answer the future they want to predict.

Regarding the modelling techniques, the resurrection of Deep Learning in the 2010s has fastened the innovations within the field of Machine Learning and more particularly within the subfield of forecasting. This has been driven particularly by the improved computing power and capabilities of computers – let alone supercomputers – which have allowed some long-invented methods to be feasible for the everyday engineer, such as the LSTM [6]. More particularly, ever since the above mentioned resurrection more and more complex models have been outperforming previous standards, such as the general RNN, the LSTM, or Tansformer-based models as of lately. The figure below represents the evolution of the size of transistors, which translates into computing power.

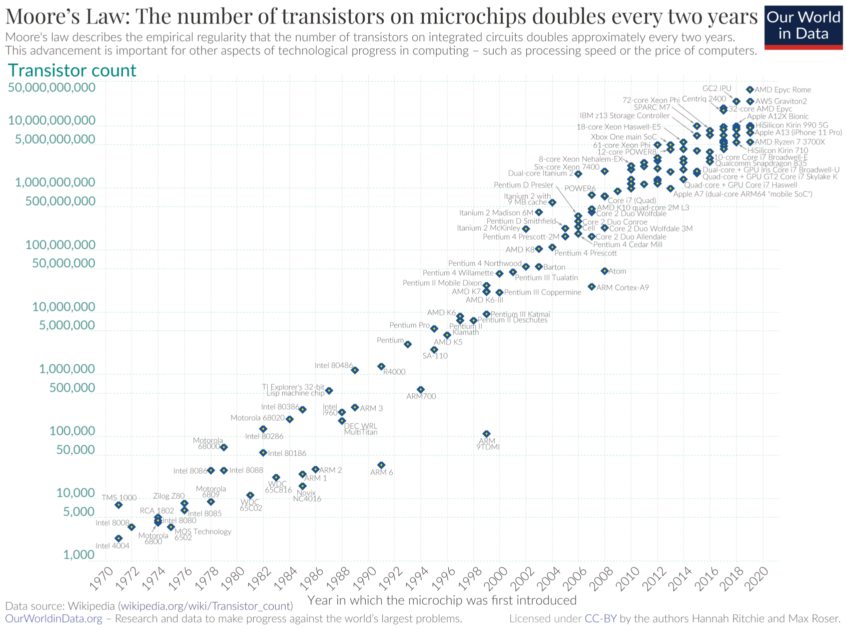


Figure 1.1: Moore's Law of Transistor integrated counts. Source: Our world in data.

The research project has been conducted as part of a collaboration between Monta and the Machine Learning for Smart Mobility section of the Technical University of Denmark.

1.1 Motivation

Having a longer reliable window of known prices is essential to be able to offer Monta's charging system users the lowest price for the period during which their car is connected to the charging point (CP). Monta is not an electricity provider, but an intermediary between end-users and utilities companies as well as providers and as such it seeks to charge the customer the lowest possible price to fully charge the car. As depicted in figure 1.2, users will connect their cars to the CPs and perhaps go to work to their offices, where their interest is not to charge immediately but to charge preferably at the lowest price, and finished the charge when they finish the work day and they need to leave. In reality, what Monta sees in Denmark and other Scandinavian countries is that users do not need to charge every time since they do short distances, or perhaps leave the car for several days connected to their home charger. It is in this situations when smart charging is most interesting and where the four-day ahead forecasts enter.

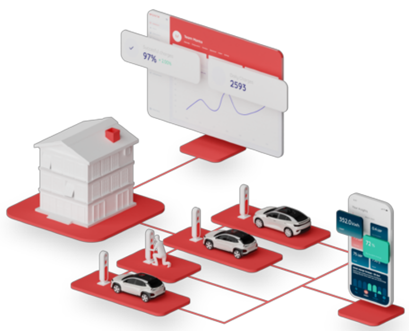


Figure 1.2: Smart charging in Monta's ecosystem. Source: Monta.

Because the forecast of the price is the going to be used for the Vehicle Charging problem, it is interesting to have an uncertainty estimation of the prediction, hence the probabilistic approach. The mean of the prediction, equivalent to the point prediction of regular forecasting techniques, will be the input of the optimization algorithm which will evaluate when is the optimal moment to charge. Coming back to why use a probabilistic model, the percentiles of the forecast will be an estimation of the uncertainty of the price, and hence can be used to further improve the optimization algorithm and drive it upwards or downwards. Moreover, day-ahead electricity spot prices are set up based on a bidding system which is general to a specific region, setting a unitary price for the whole region. This bidding is based on the costs of producing the energy, thus it is really dependant on the energy source. Because of this, there is not a linear relationship between energy production of the different sources and the final spot price, but what is more is the price series does not have constant variance, which is something that classical statistical models, and more broadly point forecasts assume. In this context probability forecasts make a lot of sense because it allows to assume a varying variance, this is, to account for the heteroscedasticity of the series.

On a different note, this is an interesting opportunity for someone who has learnt about Machine Learning in a purely academic environment, since the collaboration with the company makes it necessary to have best practices in place and aim for reproducibility and scalability of the code.

As per what is the prospect of this project within future research, this work can of course be used to reproduce the experiment and validate the general conclusions obtained, but it can also serve as the basis for researching the best probabilistic algorithm of the price, not to mention that future researchers can benefit from the frameworks developed in this project to properly train their models. It also hopefully spreads curiosity about forecasting in general and electricity price foecasting more particularly, as it is a topic that can be beneficial for research institutions and public organizations.

1.2 Aim of the Thesis

The ultimate goal is to forecast the spot price of electricity production four days in advance in order to have a longer window to charge a car when smart charging. The goal is to develop a probabilistic deep learning model that accounts for the complexity - aleatoric uncertainty - of electricity prices and effectively depicts the behaviour of the day-ahead electricity markets.

1.3 Situation

In the fiercely competitive Automotive sector, innovations seem to come from big technological disruptions, such as LiDAR, V2G, EVs, and HPVs. This has been the case especially due to the attracting force that media attaches to these disruptions. One accurate example is Apple Inc.'s development of the autonomous vehicle which was all over the news, even though Volvo has been silently developing it for years under the radar thanks to, now yes, LiDAR, a disruptive new technology. Although it is big disruptions that seem to fill the headlines, these disruptions are slowly improved and merged with other smaller innovations and it is then when the end user truly experiences the change. Again, it is the small, incremental improvements that really change the user experience and make the sector have this feeling of constant renewal, even reinvention. Far from revolutionary, this belief is present across many deep learning material given the similarities it has had with regards to its adoption. To cite François Chollet in [7] "The technological revolution that's currently unfolding didn't start with any single breakthrough invention. Rather, like any other revolution, it's the product of a vast accumulation of enabling factors – gradual at first, and then sudden.".

This is all the more true if we look at a specific technology, EVs. Electric vehicles where introduced just over a century ago, though their limited range put them out of the market. Now with, incrementally more energy-efficient batteries, EVs are not only a reality but have even outsold combustion-engine vehicles for the first time in Norway. But even when reintroduced to the day-to-day user, the adoption has been slow and it is now with attractive vehicles, long battery ranges and better charging experiences that adoption is rapidly growing. There are many pain points for the everyday user of EVs, such as the charging experience. That is where Monta originated from - tiredness of the simplistic and penurious experience of charging an EV filled with cutting edge technology. The company was born in late 2020, and has skyrocketed to be the de facto EV charging app used in Denmark and Scandinavian countries. They provide a new layer of technology that makes it easy, reliable and pleasant to charge an EV thanks to its functionalities. With a headcount just short of 120 employees, the company has a focus on sustainability and open-source and collaborates with other sustainability-focused start-ups to develop their core products helping each other. It currently operates in Denmark, Sweden, Norway, Germany, Austria, the UK and Ireland, providing service for businesses, residentials, CP Operators and Installers as well as EV drivers. Monta enables a Lead generation portal for its partners, it also bids for locations of CPs in collaboration with partners, it helps EV drivers in fulfilling the tax refund in Denmark, and it collects money for each connected charger and each transactions. Hence, Monta benefits from presence in the whole value chain of EV charging - from the hardware manufacturing to the Real estate and the usage.

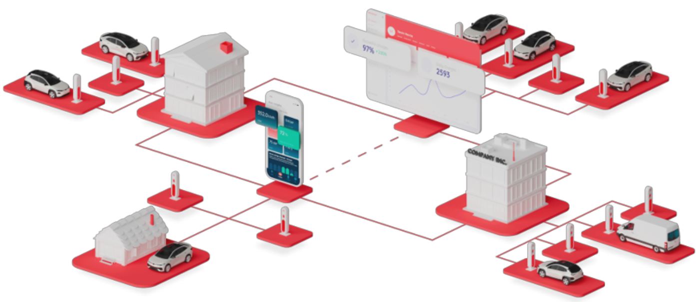


Figure 1.3: Monta's presence in the EV charging value chain. Source: Monta.

1.4 Structure of the document

The document covers the work done as part of the research to have the 4-day-ahead forecast of the price. First, the topic is introduced, making mention of the mathematical techniques, the situational context of the company and the electricity markets as well as the motivation behind the project. After this, there is a Literature review to ensure the work done is meaningful and follows the current studies elsewhere.

Following, the work is shown following the structure of a Data Science project, from idea to evaluation of results. Getting more into detail, the project's work consists of:

2 Literature Review

This section is the connection between academic research and the present work. It addresses the theoretical concepts that have shaped this research project. It is important to highlight those that have, with their scientific contributions, enabled this research project and it is also important to document how they have influenced this work for future researchers to profit from it.

Taking this into account, the key influences can be grouped into 4 categories, namely:

- Model Selection and Evaluation methods
- Design of Recurrent Neural Networks
- Smart Charging and Electricity Markets
- Probabilistic Deep Learning

These broad topics have shaped the modelling approach not only by determining the evaluation methods, but also in making a framework that is as least biased as possible, not to mention the design of model architectures. The most influential being the papers related to the first topic, it is important to define a framework to effectively compare model performance and improve forecasts without data leakage, in a sound, analytic way.

2.1 Model selection and Evaluation methods

Fata et al. [3], depict a clear and extensive comparison of different families of models and their performance. Reading this paper contributed to understanding the rolling origin evaluation and that common used evaluation metrics are MAE, MSE and MAPE, together of course with their relative or root counterparts. It also brought light to what to expect from electricity price series, showing that the first, second, third and 24th lags are critical for the prediction. This can be seen in the seasonal differencing of the series, in subsection 3.2.1. The same paper also depicts issues experienced with ARIMA models, in particular the lack of performance of the model with regards to peaks of the series, in some cases even leading to overfitting. As a last insight, they considered regression tree models the best at the time of publication, 2019, achieving a 90% reduction of test set error without exorbitantly increasing model complexity. This is interesting because future benchmark papers consider Deep Learning approaches much worth as compared to classical statistical models or tree-based models, considering the Deep Learning approaches already as a compromise of accuracy and model complexity when compared to, for instance, Temporal Fusion Transformers, which are the utmost SotA. Continuing on the same topic, [5] comprises arguably the most complete benchmark of electricity price forecasting to date. This paper has provided context with regards to making an unbiased modelling framework: in comparing different regions, in avoiding leakage in temporal features, clearly stating the train-test splits, comparing different models and also what metrics to use for comparison. It also brought light into what to expect with regards to the performance improvement that can be achieved with Deep neural networks as compared to classical models. However, the biggest influence this paper had is in terms of Reproducibility. It explains why it is important to clearly state the data source, the inputs of the model and the selection of hyperparameters. In the study they conclude that Lasso-estimated Autoregressive models which were once SotA for electricity price forecasting, have been displaced by LSTMs and furthermore by probabilistic LSTMs and more complex ensemble models, though the last ones suppose an extra level of complexity, unbalancing the tradeoff between accuracy and complexity. Moreover, Rob Hyndman in the iconoclastic [4] clearly explains the different cross-validation frameworks that can be used for

Time Series data, which was useful to define the desired framework without temporal leakage and without overlapping of training, evaluation and testing data sets. This was also seen in and [8]. Going back to Hyndman's book, it represents to date the best resource to get started in the field of forecasting with very good examples in R that allow one to understand foundational concepts.

2.2 Design of Recurrent Neural Networks

Deep Learning supposed a revolution in the early 2010s due to successful applications of such models for difficult problems, such as the AlexNet [9]. More recently, recurrent neural networks took the lead with big advancements especially in the fields of Natural Language processing [10], and three of the most prominent researchers, Yoshua Bengio, Geoffrey Hinton and Yann LeCunn received the Turing award, acknowledging the impact that deep learning is and will have in information and society. Going back to recurrent neural networks, the original conceptualization of the LSTM [6] was useful in understanding the need and benefits from such model, especially how the gates allow to keep long-term ifnormation and help solve the vanishing gradients problem. The learning material from New York University, designed by LeCunn et al. [11] was very illustrative of the functioning of the recurrent gates, especially with the comparison with the electronic logical gates. Though this materials set the foundations to apply the LSTM, it is [12] and [13] that contributed to implementing the neural network model and helped apply it to the temporal cross validation framework. Further developments in the design of the model, as well as important considerations such as correctly defining the window size, the output window size and the batch size were thanks majorily to [14]. This article also brought light to questions such as: Is it better to preprocess the data - as in time series models - or input raw data? Is it better to follow a time series approach w.r.t. data or use them as independent features, like in a supervised learning problem? Is the neural network really better with more nodes and more training? If so, when to stop? After reading it is clear that generally, for the special case of forecasting it is not necessary to extensively preprocess the time series, that there needs to be a compromise not to overfit the models, which varies depending on each model, and that it is better to treat data as a time series, rather than feeding the models the lags to predict the future prices, as in a supervised problem. Finally, [15] outlines a comparison of single output and multioutput models, which has been useful in the final stages of the project in deciding how to achieve better generalizability of the model and improve accuracy.

2.3 Smart Charging

In line with a more sustainable mobility across society, be it for goods or for people, especially for EVs, there needs to be a more sustainable organization of their charging. As discussed in [16], Prioritisation of immediate charging incurs in greater CO2 emissions ($+15\%$) and costs ($+20\%$) as compared to delaying charging until electricity prices are lower during the day. To fully understand what Smart Charging is and its implications, [17] provide an exhaustive overview, with a focus on how it affects the electric installations. They conclude that smart charging allows to fill demand valleys and reduces the maximum power demand by approximately $+20\%$. Furthermore, Monta's own resources on Smart Charging [1] clearly cover the basics of Smart charging and how it affects their users, which has influenced how to evaluate the forecasts of the models. Following the same topic, and though a bit dated, [18] explains the complexity of the vehicle charging problem, especially related to potential use cases of Smart Charging such as Vehicle to Grid services to buy and sell electricity. This paper is referred to as "dated", because it outlines problems and important considerations to take into account when solving an optimization problem such as the vehicle charging problem - i.e: when to charge your car minimizing costs in a specific horizon - though in Monta's case simple heuristics implemented in robust programming languages have proven very successful for that matter.

2.4 Probabilistic Deep Learning

Probabilistic deep learning is the concept of learning probability distributions and uncertainties over the weights of the networks and the target variable [19], as opposed to normal deep learning, where each weight and output, is considered to be a single value [20]. The uncertainties can be broken down as either *epistemic* or *aleatoric* uncertainty, where epistemic is on the model parameters itself, and the aleatoric is from the signal [21]. The aleatoric uncertainty can be modelled by modifying the models to get a distribution conditioned on the output values of the model [22] or by using methods such as the proposed by Hüllermeier et al. [23]. The epistemic uncertainty can be modelled by learning a distribution over the weights using [24], which uses the techniques described in [25] or it can be done using the methods described in [26]. Regarding the project, major influences have been [27] and [28]. The first for introducing me to the topic of probabilistic modelling and setting the foundations to the approach. The second, on the other hand on the application of Deep Learning and the transition from point estimates to distribution forecasts. Moreover, [29] provides a down to earth explanation of the problem set up and the interest of distribution forecasts rather than point forecasts. It is as well helpful in understanding shortcomings of point prediction methods in capturing cross-correlation. In particular, it describes that the common approach of modelling the individual series and even the somewhat more complex approach of modelling the conditional probability of a high-dimensional series are computationally expensive as the number of parameters grow quadratically, whereas using an encoder-decoder architecture allows to significantly reduce the learned parameters by encoding the signal to a lower dimensional-space.

3 Methods and Materials

3.1 Introduction

The following chapter addresses the design and implementation of the project. This section covers everything the reader needs to know in order to conduct the experiments, this is, information about the data set, the experimental set up and the prediction algorithms used. All code in this project is published in Github [30].

3.2 Data set

The data used in the project comes from NordPool, the company that operates the day-ahead electricity markets and acts as the physical commodity exchange (energy exchange) of the Nordic and Baltic regions. The historical data for electricity prices is public, though it cannot be easily collected since it is only the latest prices that are publicly disclosed. Thankfully, NordPool provided access to a server that contains all data necessary for making the project a reality.

To begin with, it is necessary to have an overview of how the European electricity markets work to properly follow the project, although it is outside of the scope to give a detailed explanation of its mechanisms. The electric grids and the electricity market in general are composed of several layers that define prices for production and consumption for the next day, for the next five years, and in some cases for the next 30 years. The latter lays the grounds from which prices are defined in the shorter horizons, whereas the former define the price of closer time ranges. These shorter time ranges, the day-ahead market in particular, is a bidding commodity-like system for defining the final price. Producers and points of demand bid for supply and demand and the price is set where they both meet. Though this is the ideal theoretical functioning, reality differs from this perfect supply-demand mechanism and resembles imperfect information zero-sum games [31]. Firstly, The bidding starts with the offered supply of the cheapest energy sources, and is then complemented with traditional energy sources which are more expensive and are the ones that end up being the driver of rising prices. This behaviour is reflected in figure 3.1. On top of that, the structure is effectively very complex since there are a small number of producers, where entry barriers are very big and regulators are sometimes influenced by decisions from these corporations [32]. Large producers also generally have stakes in competing corporations in other countries, further influencing and setting a conflict of interests in the exchange of electricity between regions [33].

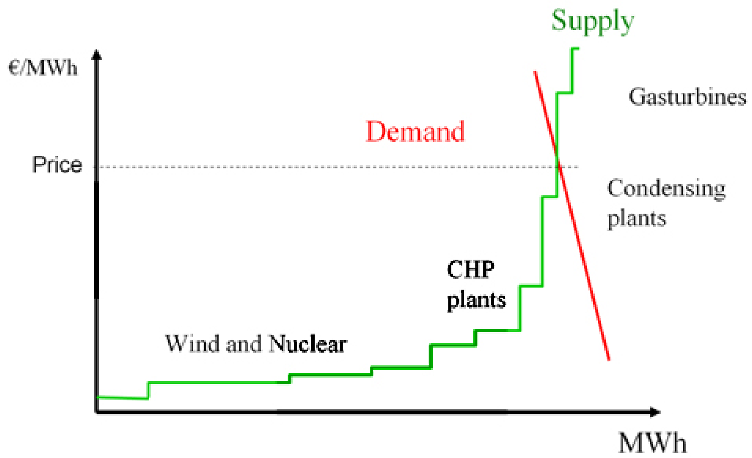


Figure 3.1: Supply and Demand of NordPool's power market. Source: Risø DTU.

Coming back to the project, it is only going to focus in the day-ahead electricity market, this is, electricity spot prices. The prices are set up according to the energy supply available in each region. Thus, the energy source directly affects the spot price of electricity, since its the cost of production that producers take into account to set the price for the bid, where renewable energy sources and nuclear energy are cheaper to produce and hence result in lower spot prices.

Regarding Monta, the company now operates primarily in Norway, Sweden and Denmark, though it also has operations in Germany, Finland, the UK, Ireland and Austria. Of these countries, this project is going to focus in the Scandinavian region: Denmark, Norway and Sweden. These three countries have similar grid structures, though they are not unique and have sub-grids even within their borders, as can be seen in figure 3.2

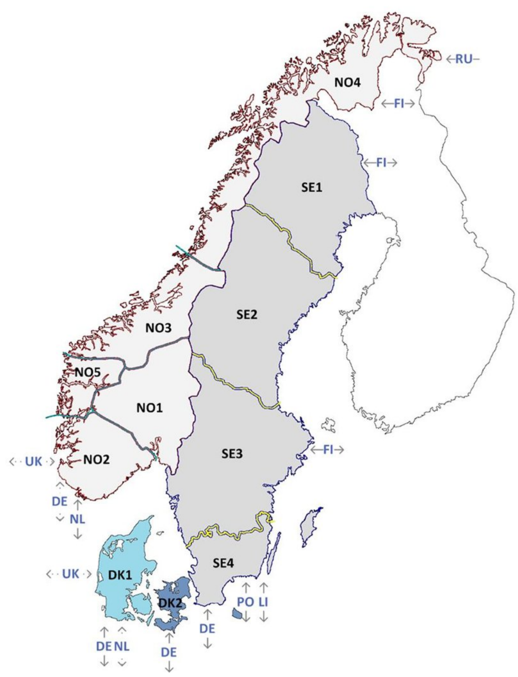


Figure 3.2: Scandinavian NordPool price areas.

The regions are defined by NordPool and the countries' electricity grid to handle large and long-term congestions. In particular, DK1 and DK2 for Denmark West and East respectively; SE1, SE2, SE3, SE4 from North to South of Sweden, where SE1 and SE2 define the price together; NO1, NO2, NO3, NO4, NO5, where NO1 and NO2 on the one hand, as well as NO3 and NO4 on the other have historically defined the spot price in conjunction.

$$Regions \equiv \{DK1, DK2, NO1, NO3, NO5, SE1, SE3, SE4\} \tag{3.1}$$

Therefore, the project is going to focus on obtaining a model that can predict the electricity price for the next 96 hours - 4 days ahead - in the regions DK1, DK2, NO1, NO3, NO5, SE1, SE3, SE4. The data for these regions is present in data tables within the ftp server provided. The data tables date back all the way to 1990s, though in line with the project's scope and for the sake of simplicity, the focus is on the last 5 and a half years, from 2017 onwards. The data tables contain the data points for the 24 hours of a day, for all days of a calendar year and some days that overlap in the first or last week of the year. The data tables are of the format seen below:

Hour of day	1	2	3	...	22	23	24
01/01/2021	24,95	24,35	23,98		26,98	26,44	25,64
02/01/2021	25,59	24,91	24,74		26,96	26,85	25,94
03/01/2021	25,27	24,59	23,86		27,04	26,18	24,92

Table 3.1: Example of NordPool data table. Price for NO1 in 2021. Source: NordPool.

Focusing on data from the selected regions, the data needs to be converted to a series-like format to be useful for the project. The process of preparing the data is as follows: Reading all data tables (2017 to 2022), concatenating its rows one by one, adjusting the data types of the needed columns and interpolating the missing values with a linear interpolation of the closest values. The method of linear interpolation has been decided because the absolute number of missing values for each region was very low, at around four to eight missing values over all years.

The process above is done for each region of interest and the data saved for further analysis. Once processed, the data is a univariate time series, with a frequency of one hour, with values ranging from January the 2^{nd}, 2017 to March the 11^{th}, 2022. This means the time series for each region i is composed of 45456 data points $P_{i,t}$, as seen below:

$$P_i = \{P_{i,t}, \quad t = 1, ..., 45456\}, \quad i \in Regions \tag{3.2}$$

For the purpose of this study, it would have been interesting to look into exogenous variables that could explain the price of electricity. In line with what is discussed about the electricity markets, it would have been interesting to incorporate into the study the energy source at any given time, the total co2 generated per region or the exchange of electricity between regions. This was not possible due to lacking access to such data, which motivated the interest in, instead, studying the individual time series.

3.2.1 Exploratory Data Analysis

This section addresses the exploration of the study data. The purpose of this is to gain a better understanding of the time series with regards to the other regions and with regards to itself. As part of this work, the series are added together with date and time columns to explore patterns with regards to time. Interesting work in this realm are on one side plots that allow us to understand the price series behaviour, and on the other side the time series analysis, especially lag plots and other visualisations that allow us to understand the series' autocorrelation, but also statistical tests to check for stationarity, for instance. The latter work is centered around NO1 region, for reasons that will be explained along the exploration of the data.

As an exploration of the data, it is interesting to look at the original series and how they differ from the other regions. This plot gives an indication of how good could a model generalize between regions, which can be seen in figure 3.3:

Figure 3.3: Price evolution in all regions.

It can be seen that northern regions have been much more stable over the recent price hikes, whereas southernmost regions, DK1, DK2, SE4 and NO1, display much higher volatility and seem to be heteroscedastic. This means the series do not have constant variance with time, but have varying variance, which seems to be growing with time. This will be addressed later on, though it can be an issue for classical time series models since they assume time series to be homoscedastic.

To follow along the line of discussion of the variance of the prices, it is interesting to look at the distribution of the price by region in figure 3.4. When the distributions are plotted symmetrically next to each other, it can be seen that, in fact, the northern regions NO3 and SE1 have very similar price distributions. Interestingly enough, it can be seen that all regions have had negative spot prices, with DK1 and DK2 having considerable outliers in the negative range. These negative prices are indeed possible, due to the supply-demand bidding mechanism of the day-ahead market. If too much is produced, particularly from hard to store energy sources, the negative price encourages the use of electricity, reducing the load. In the context of EVs, this periods are very appealing since it will be a passive income from charging your car.

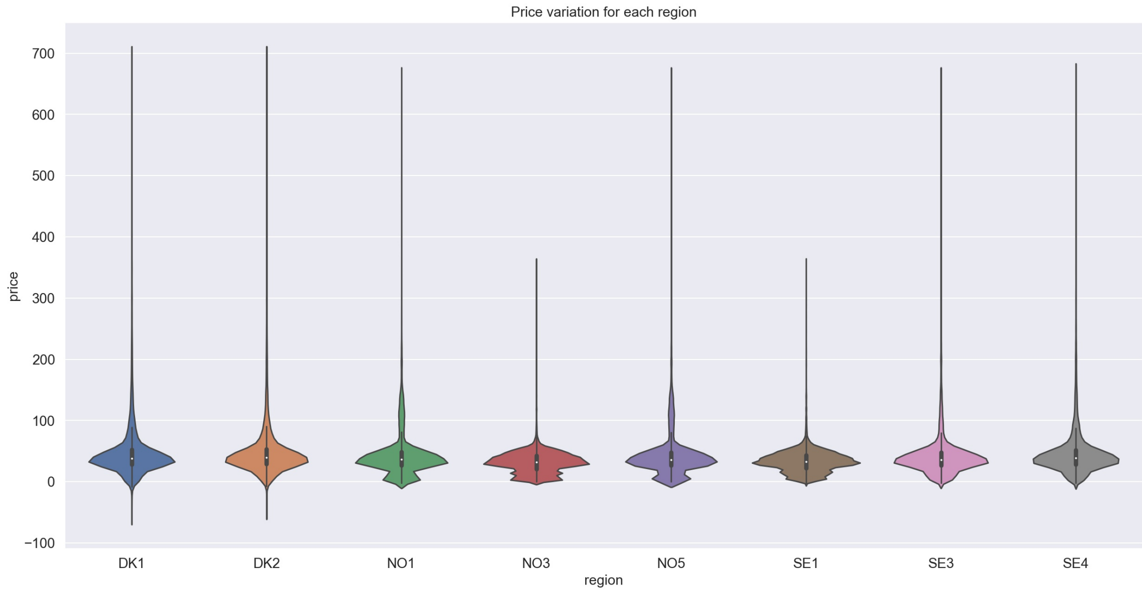


Figure 3.4: Price distribution of all regions.

The above distributions give an overview of how the price varies in each region, but seeing how the price varies in absolute terms over time is not as insightful as it may seem. In the context of smart charging, it is the blocks of time during each day that are most interesting, specifically working hours and night time. This is because that's when there is more room of time without the car being used and hence when the extra information derived from having the price four days ahead is of more value. Figure 3.5 represents the price range for each time of the day by region. This plot gives a better understanding of the price evolution along the hours of the day and hence whether smart charging will be beneficial for Monta's users. With this in mind, looking at the below figure it is clear that there are regularly two price hikes, around 10:00 am and around 21:00pm. This observation confirms the hypothesis of the working hours and night time, thus proving the benefit for Monta's users of connecting the car to a smart charging point for the whole night and charging some time later than 21:00 pm.

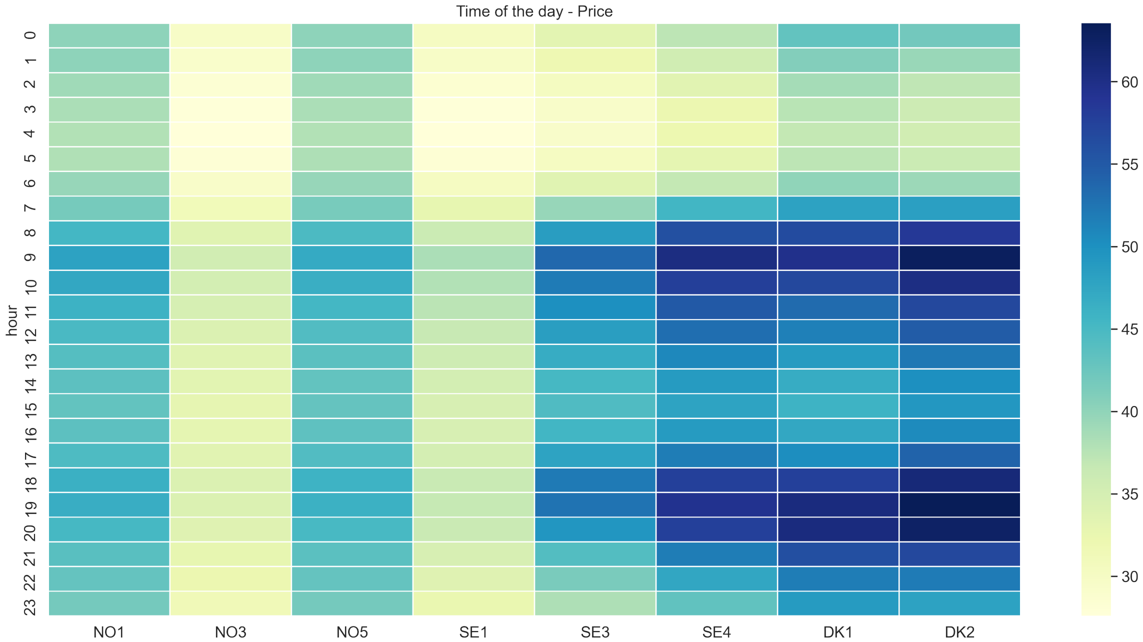


Figure 3.5: Regional spot price by time of the day.

In the following, the exploration of the data and the time series analysis is conducted for the NO1 region. The purpose of this is to focus on a region and prepare models accordingly, rather than having models for each region and deciding which one to keep using forward. The NO1 region is the selected one since it is not the most volatile region nor the most stable, thus allowing for better generalization of the models to the rest of the regions. More specifically, it is not the most heteroscedastic, which will be detrimental when modelling since it will make the models be too overfitted to a region and not characterize the other series properly. To start with, it is a good practice to simply look at the time series and its weekly and monthly moving averages, to depict the trends and movements of the series. Figure 3.6 shows the evolution of the price, where it can be seen there are several sudden price peaks which can be considered outliers.

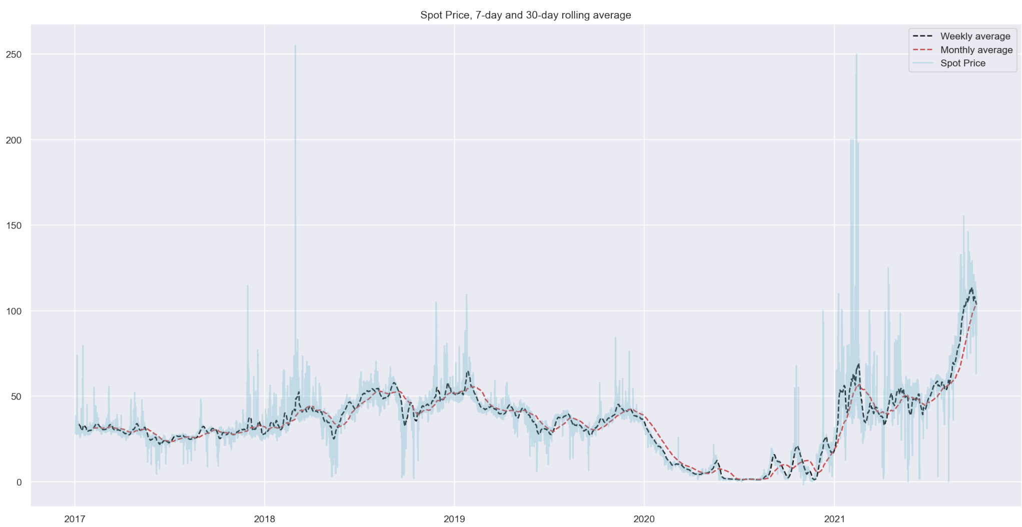


Figure 3.6: Detailed price evolution of NO1.

From the above chart it can be seen that price ranges differently every year, with 2020 having lower minimum and maximum price than 2021, for instance. Looking further into that, figure 3.7 shows the maximum, minimum, average and sum of prices for each month and year. This figure brings light to how the price evolves during the year. Though not fully showing seasonalities, after looking at the chart it is clear that the model needs to be selected based on different periods of the years available, otherwise it will be very prone to overfitting due to ressembling too much a specific period of the year.

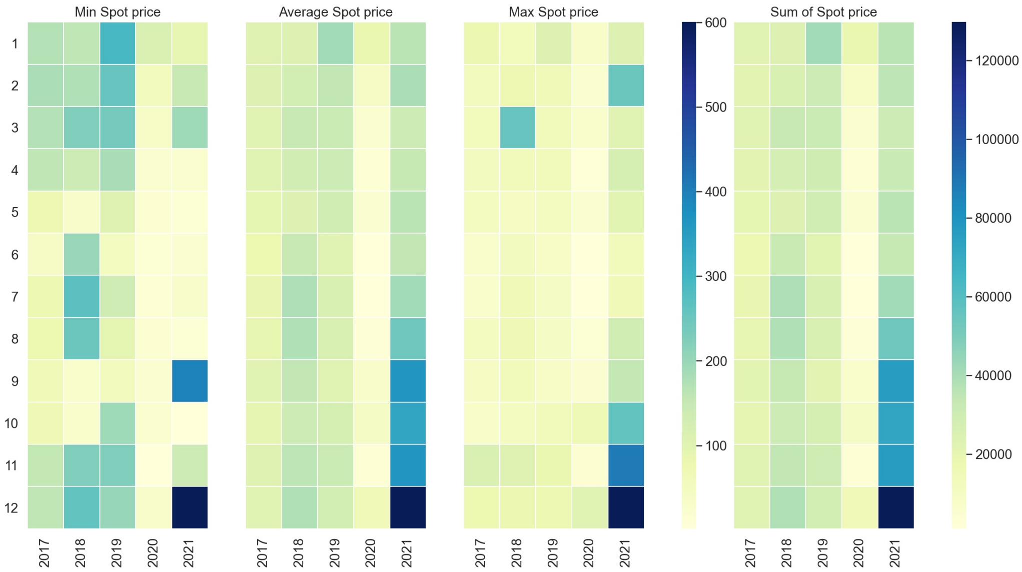


Figure 3.7: Price occurrence by year and month.

Time Series Analysis

The purpose of this section is to understand the behaviour of the series to better configure the models. Notably, the aim is to make the series stationary and understand which lags are the most autocorrelated with the current time step, which is relevant to define the sequence length for the predictive models.

The analysis is carried out visually, through the Autocorrelation and Partial Autocorrelation plots, as well as numerically through the Augmented Dickey-Fuller test (ADF test onwards) and the Kwiatkowski-Phillips-Schmidt-Shin test (KPSS test onwards).

The original series seems heteroscedastic by how the variance of 3.6 seem to have an increasing trend. A good start is to check the lags of the series to see if they have a strong correlation:

Figure 3.8: NO1 Lag plot.

The figure above shows the first to eigth lags of the series compared to the next (latest) point. Interestingly, the correlation between them is clear, since they are concentrated across the 45º line region of the chart. However, there are other factors that drive the correlation between lags which are unknown, which is clear when looking at the small concentration of points around the $[200, 50]$ for the *t+1 vs t-8* subplot. This can be an indication that the autocorrelation is not going to be directly visible with a single differentiation.

Diving into the stationarity of the series, a standard procedure defined in [4] is to first look at the Autocorrelation, later at the Partial Autocorrelation and the confirm the hypothesis with the results from the statistical tests. Following this procedure means checking for stationarity at different levels of differencing of the series, starting from the original series, following with regular differencing and seasonal differencing onwards. The original series autocorrelation plot shows the series is not stationary originally, as can be seen in figure 3.9:

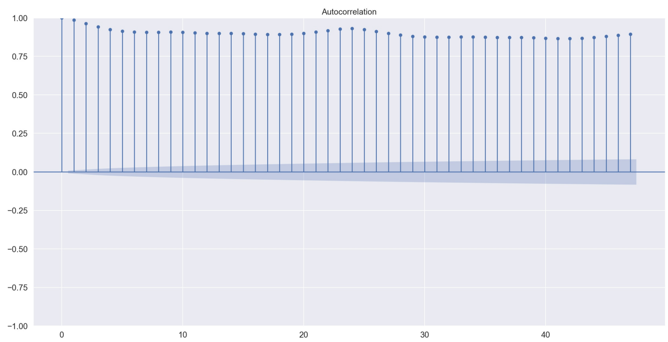
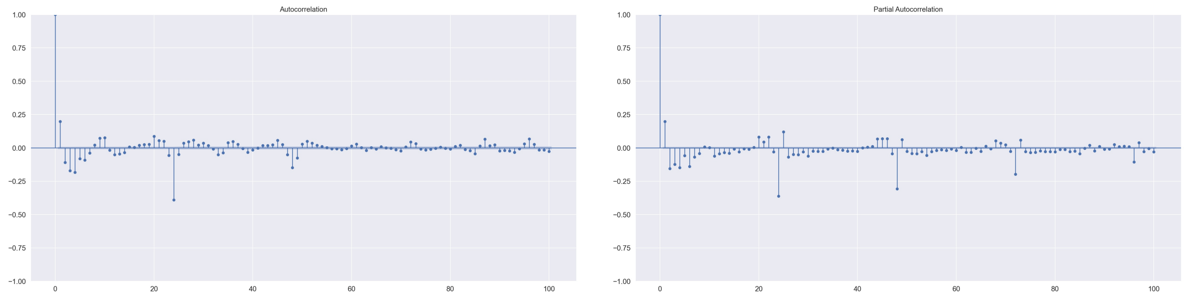


Figure 3.9: Autocorrelation of the original series.

The figure clearly shows the series is not stationary, as none of the lags are below the confidence interval, meaning all of the lags are significant. The purpose is to reach a stationary series, so in the context of autocorrelation, ideally only the first lag would be statistically significant, above the confidence interval represented by the blue-coloured area. Following the same chain of thought, the autocorrelation plot for the differenced series is checked and the series still is not stationary.

Finally, the series is both seasonally and regularly differenced to check the autocorrelation. This means the lags are calculated as the difference of the series with the values from the previous seasonal period and the difference of these with the prior values as in equation 1. When doing this, the series seems to be stationarity and it can be seen that the most relevant lags are X, Y and Z. The autocorrelation plot sets no red alarms, so the partial autocorrelation is also checked and it is clear, according to the visual inspection, that the series is stationary, as can be seen below:



(a) Autocorrelation of the differenced series, NO1 price.

(b) Partial autocorrelation of the seasonally differenced series, NO1 price.

Figure 3.10: Visual inspection of the autocorrelation of the NO1 series.

The above charts represent stationary behaviour of the series, though not the canonical representation of a stationary series, white noise-like. To ensure the differencing of the series was the correct one, several more orders were tried out, though the results were clearly worse, either because the lags became more significant or the decay in the weights significance is so high that there is growing negative autocorrelation. This behaviour is represented by a sinusoidal-like chart, where the autocorrelation of the lags does not converge but oscillates between the positive and negative autocorrelation.

Finally, the series is tested with the ADF and the KPSS tests. The results of these tests clearly shows too that the seasonally differenced series is stationary, which means the series is ready for forecasting with classical time series models. These tests respectively measure the p-value of the series being stationary and non-stationary, which in both cases show the series is indeed stationary:

P - value	1%	5%	10%
ADF Test	-3.431	-2.862	-2.567
KPSS Test	0.739	0.574	0.347

Table 3.2: Stationarity tests and their statistical significance based on the p-value.

3.3 Experimental setup

This section covers the experiments done throughout the study, how are they organized and why. The following part of the project ensures the work is reproducible and it groups all the best practices used along the data science pipeline of the study.

The project aims to create a framework for iteratively increasing the modelling complexity with the end result of obtaining the best prediction performance. To do so, starting from simple historical data aggregations, the complexity of the models is increased until reaching what is the best depiction of the true behaviour of the time series. In other words, starting from historical averages, then classical time series models, then several LSTM architectures and finally deep learning architectures with probabilistic layers. At each step the aim is to obtain the best prediction possible and only increase the modelling complexity when achieving better predictions with the more complex models. This process of defining the best configuration of the models is all going to be conducted for the NO1 region. As stated in the Exploratory Data Analysis, the region sits in between the rest in terms of variance and difficulty of modelling. After this, the models are tested in an out-of-sample period to evaluate how do they compare with each other. With this, it is possible to properly compare the real performance across regions.

This framework inherits from common rationale, but also from Software development practices of creating and delivering continuously better results or products. Taking it one step further, and related to the author's background in Industrial Engineering, this frameworks inherits from Lean and Kaizen, the japanese-born mentality that small, continuous, incremental steps are key to make the big exponential leaps that technological developments are borne from. With this in mind, and related to the temporal characteristics of the problem at hand, it is key that the models to be developed generalize correctly and are not characterisations of specific times of the year. Inspiration comes from Bengio et al. (2002) [34], where they define a framework to evaluate models based on a selected metric, especially for time series forecasting. This framework defines the need to effectively compare models based on a unique metric, which for the case of study will be Root Mean Squared Error.

$$RMSE = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2}, \quad \hat{y}_i = predicted, \quad y_i = observed \tag{3.3}$$

They also refer to sequential validation, defined to be "based on the analysis of the sequence of losses obtained by sliding a learning algorithm over the time sequence", where "the most important result of the sequential validation is the average loss". With such purposes, the data for October 2021 onward is left for out-of-sample analysis. The remaining data points, January 2017 to September 2021, are divided into 20 blocks of time in which the model is going to be trained, validated and tested out. The purpose of doing a temporal cross validation is to ensure that the model generalizes correctly, though it is important to bear in mind that the neural networks will, at least in theory, benefit from being fed considerable amounts of data rather than a small data set. Hence, the compromise is achieved at having the 20 periods, where 74 to 82 days are used for training, eight days for validation and four days for testing purposes. A simplification of the structure can be seen in figure 3.11:

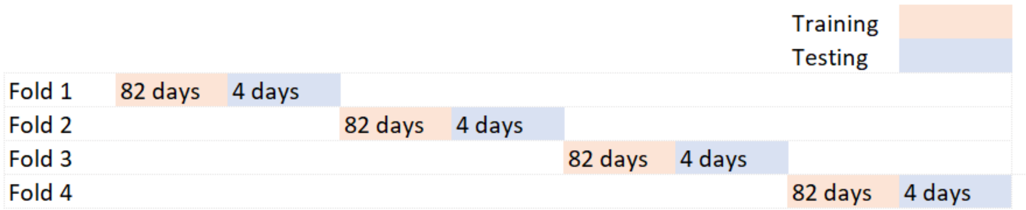


Figure 3.11: Sequential validation windows for classical models.

As can be seen in the figure above, there is no overlap between the different periods. There should not be since that would incur in information about the data being leaked to the test, which will not properly show how the models are performing [4]. On top of that, it is important that the time is not shuffled for the cross validation periods, since this will break the temporal relationships present in the data (think of Autocorrelation) and the models will not properly characterize the time series [8].

The framework for the neural networks is slightly different, however. The same principles remain since they are related to the core objective of the project, but there are minor differences to better suit the deep learning perspective. Particularly, the model now works as a rolling window, with which the model now has a sequential validation inside the previously defined one. This rolling window is composed of 168 data points (hours), one week, to predict the following 96 hours, four days. These rolling window slides through the data of each period, for which 10% of the previously defined training data is now validation data, and the rest remains as training data. The points reserved for testing are the same at each period to preserve the comparability of the results between models. The figure below represents the newly described framework:

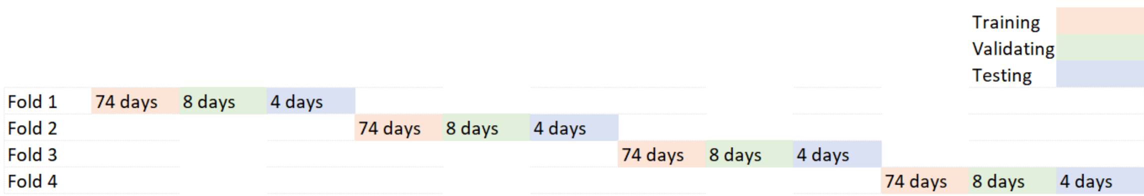


Figure 3.12: Sequential validation windows for deep learning models.

Summing up the structure, the different model architectures are tried out across the 20 periods and their errors noted down, which are then averaged and is this metric that is used to compare the models. In particular, the metric used is Root Mean Squared Error, and it is selected due to the importance it gives to the most significant errors [35]. This is of interest as the time series at hand has several outliers, as discussed in 3.2, so using the RMSE raises the bar of considering a model better than a previous simpler one. More complex models, especially LSTM architectures, have the ability to characterise the outliers, which makes the RMSE worse for the scenarios in which the period has outliers. In the business context of the problem, it is also meaningful to have a more robust model, as obtained when using RMSE over MAE, because it ensures the outliers are correctly predicted and hence Monta's user will be able to benefit from surprisingly low prices or not suffer price peaks. Apart from the RMSE, Mean Absolute Percentage error (MAPE onwards) is also used as a secondary indication of the validity of the obtained accuracy. In evaluating predictions, only RMSE is taken into account, though MAPE is calculated to give the business a more tractable metric of how the model predicts the electricity prices.

$$MAPE = \frac{1}{N}\Sigma_{i=1}^{N}\frac{|\hat{y}_i - y_i|}{|y_i|} \cdot 100, \quad \hat{y}_i = predicted, \quad y_i = observed \tag{3.4}$$

The designed framework also takes care of scaling the data. Specifically, the data is scaled by a MinMax normalization. That is, subtracting the data by the minimum value and dividing by the difference between the maximum and minimum values. The reason to use this feature scaling technique is that it preserves the original distribution of the data. Since the distribution the data follows is unknown, this technique does not introduce assumptions nor bias into the model.

3.3.1 Process

The framework defines a process to search for the best predictive models. This procedure sets the rules upon which to decide if a model is better or worse than the rest.

The temporal evaluation framework takes into account every model configuration considered. The models are evaluated following the steps followed in the in figure 3.13. In short, if the model fits the defined framework, it is trained according to the training split. If the training contains a validation split, the inference of the model is evaluated looking at the loss function, and after evaluating the training it can be trained again if needed. Following the training, the model predicts the defined window of time, which is then evaluated based on RMSE and MAPE. Because of the greater susceptibility of MAPE to overforecasts than to underforecasts, the primary metric is RMSE, and MAPE is used as a gudidance and a business-like metric to understand the predictions. The figure depicts boxes with vertical sided lines, which represent steps where no further interaction is needed. These are also the steps where the calculations, be it training or forecasting are done for every period of the historical data.

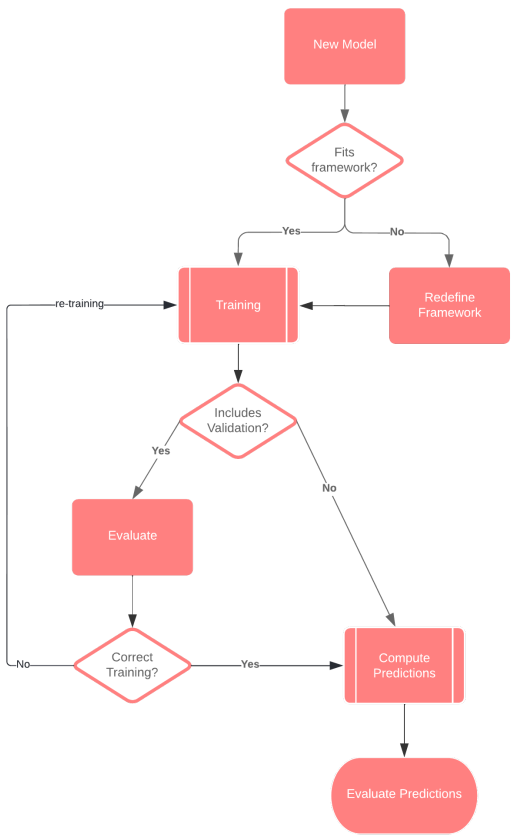


Figure 3.13: Process to compare predictive models.

3.3.2 Biases and Dilemmas

There are several considerations that need to be addressed as they make the models biased and could lead to ethical dilemmas.

The first thing to point out is Normalization. Though it is common in the industry to simply normalize the available data, this makes information to be leaked from the train to the test and validation sets. It is not paramount to address this, however, because as mentioned it is a common ”bad” practice, as well as because the final predictions are to be made out-of-sample, thus the final evaluation of results will be correct regardless of how the model used has been defined as the best.

Regarding the time splits, it could also be interesting to make a bigger number of temporal periods, breaking perhaps the data into 40 periods instead of 20. This will better ensure that data is not overfitted to specific periods of time.

Regarding the modelling techniques, most have been selected because they are very present in the previous literature and are of extended use. However, it is known that the time series are heteroscedastic, while most of the models used assume variance to be static across the series. Regardless of the assumptions of the intermediate models, the probabilistic deep learning model can account for both the uncertainty innate to neural networks (epistemic) as well as to the uncertainty of the data that conforms the time series (aleatoric). This last model is the ”best” depiction of the time series and, as such, one could argue it doesn´t bias the results due to its architecture.

3.4 Prediction algorithm

The project iteratively searches for the best prediction algorithm based on the historical data of the NO1 region, to then evaluate the models again on all regions over a future period. This section addresses why the algorithms are selected, their structure and benefits or shortcomings.

3.4.1 Historical Averages

Firstly, before jumping into complex modelling simple historical averages are calculated. This is the most straightforward way of evaluating the trade-off between complexity of the models and results. If a model, which fairly easily will be more complex than a simple historical average, is not forecasting more accurately, then its a worst model. Following this reasoning, three different historical averages are calculated:

- *Historical average 1:* All historical data points are grouped by hour, day and month of the year. Plainly, the value predicted for the 1^{st} of April at 22:00 will be the average of the price at that hour in the month of April over the previous years.
- *Historical average 2:* All historical data points are grouped by hour and day. This is, the value predicted for the 1^{st} of April at 22:00 will be the average of the price at that hour on the 1^{st} of every month over the previous years.
- *Historical average 3:* All historical data points are grouped by hour. the value predicted for the 1^{st} of April at 22:00 will be the average of the price at that hour over every day, for every month over the previous years.

3.4.2 Classical Time Series models

The next level of complexity are models of the family of classical time series models. Particularly, the models used are Seasonal AutoRegressive Integrated Moving Averages (SARIMA onwards). It is a statistical analysis model that uses the moving average and the seasonal autocorrelation with previous values of the time series to predict future points of the series. The two components, AR and MA, are commonly chosen according to the PACF and the ACF accordingly (see figure 3.10), since each of the functions studies the corresponding components of a series. With regards to the question of why to use SARIMA, it is extensively used and proven to output good results. The model is widely used in the finance and supply chain sector [36], which makes the whole research more transparent to the common reader. The model has the general parameters of ARIMA models, as well as the parameters due to seasonality. The details are in Appendix B, though it is not part of the scope of the project to dive into the mechanisms of these models. In short, there are 3 configurations tried out. Given the analysis carried out in subsubsection 3.2.1, the model is stationary when differenced one order seasonally and one order of regular differencing. Therefore, 3 combinations are defined looking at the stationary series and hypothesizing which will perform best. The 3 combinations tried out are:

- *SARIMA 1:* SARIMA $(0, 1, 1)(0, 1, 0)_{24}$ - Model that differences the series as defined and takes one lag for the Moving Average.
- *SARIMA 2:* SARIMA $(1, 1, 0)(0, 1, 0)_{24}$ - Model that differences the series as defined and takes one lag for the Autoregression.
- *SARIMA 3:* SARIMA $(0, 1, 1)(1, 1, 0)_{24}$ - Model that differences the series as defined and takes one lag for the Moving Average and one for the seasonal autoregression.

3.4.3 Deep Learning

Advancing towards more complex models there is the Deep Learning approach. These family of models are built from structures similar to biological neural networks [20], from which they inherit the name. Neural networks allow for much greater flexibility of design, for instance allowing to apply regularization to the input, to the weights or to the output, or changing the loss function, to name a few. Moreover, and this is explained further in the next few lines, these are non-linear models, which better reflect the inherent behaviour of the electricity prices, the series at hand. Regarding the structure, these neural networks are in short groups (networks) of cells in which information is processed. The information flows down through the network and is transformed through each layer of the network as the information is processed by a non-linear transformation. This is the activation function, which coming back to the comparison with the human brain, this can be interpreted as the result of the chemical reactions happening in our brain that make the information flow from neuron to neuron. Going back to the architecture, figure 3.14 shows a simplified architecture of a neural network, where given inputs $\underset{i}{\overset{m}{X}}$, these are multiplied by weights $\underset{k}{\overset{m}{W}}$ and added together with the Bias b_k, and passed through the activation function φ.

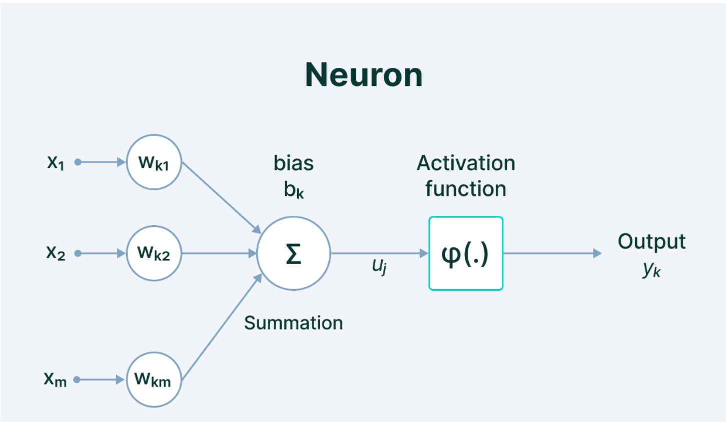


Figure 3.14: Simplified diagram of a Neural Network. Source: [37].

The neural networks similar to the one above have, however, several shortcomings. The most notable is perhaps that they have trouble in keeping a ”memory” of past observations, so in the case of sequential data they make the models too reliant in the latest observations. This problem can be addressed by the use of Recurrent neural networks, which reiterate the information over the units through time, though they also have their shortcomings, especially that the information can be largely lost due to the gradient exploding or vanishing [11]. This issue is taken care of in this project by using LSTMs, a kind of recurrent neural network which has the capability of ”remembering” both long and short-term observations thanks to the use of ”gates” that make it remember or forget both the internal state and the activation [6]. This new architecture is explained in detail in Appendix C. The description given is for a single LSTM unit, though in practice several units compose an LSTM layer, which magnifies the effect sought after with the use of the LSTM units. An illustration of an LSTM layer can be seen in Appendix C. Because of the combination of gates and states, an LSTM layer is able to keep information over time, both recent and long in the past, but this is just an addition to the key component of neural networks: how the optimize to achieve good results.

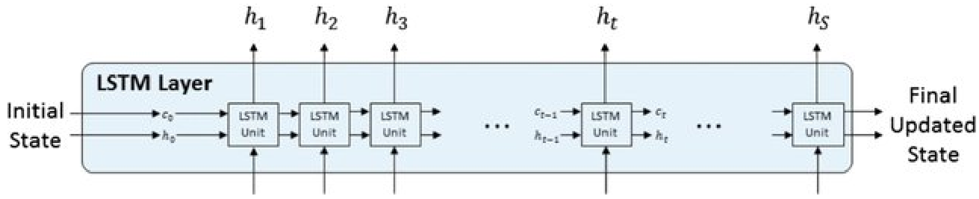


Figure 3.15: Example of an LSTM layer. Source: [38].

But how do they optimize to achieve good results, i.e: How do they learn? The networks learn by a two-sided process: The inputs are processed by the first layer, and the weights are set as the outputs of each layer are passed forward until it reaches the final output layer. The last layer’s output is evaluated towards a loss function through Stochastic Gradient Descent and the weights of each layer are updated backwards depending on the gradient of the weights with regards to the error of the output [39]. The process by which weights are set downwards is called forward propagation, whereas the update of the weights with regards to the loss function is called Back-propagation, and is what makes neural networks so good. Getting more into detail, the mechanism used to determine how to update the weights backwards is Stochastic Gradient Descent. This is an algorithm which seeks to minimize an objective function $J(\theta)$ by iteratively updating each parameter θ by a small amount based on the negative gradient of a given data set and according to a learning rate previously defined. The algorithm calculates the gradient for one training example at each iteration, and the gradient determines how to update the weight parameters at every iteration (epoch). The formula to determine the updated weights is:

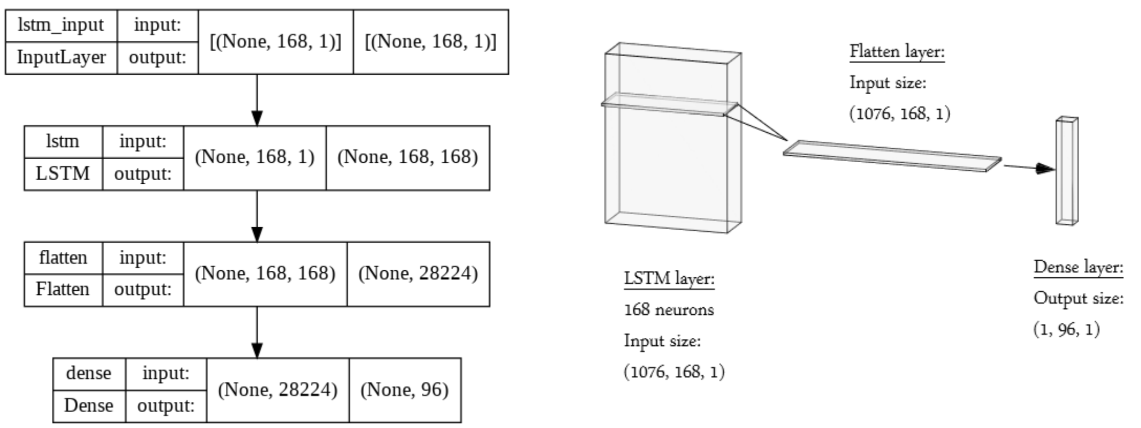
$$\theta_{i+1} = \theta_i - \alpha \times \nabla_\theta J\Big(\theta; x^j; y^i\Big) \tag{3.5}$$

Coming back to the project itself, now that the functioning of neural networks and the LSTM units is clear, there has been three configurations tried out, all of which follow the framework of the 20 periods and for all of them the sequence there is a rolling window of input size 168 steps (one week) and output size of 96 time steps (four days). Furthermore, all of the models use Adam [40] as an optimizer to perform the Gradient Descent, which is an optimization algorithm to reach the absolute minima of the error, this is, minimize the error. This optimizer builds on top of the original gradient descent by adding momentum and requires less memory. The loss function represents the objective of the minimization problem. In the case of predicting the electricity price, the aim is to minimize the error of the forecasts with regards to the actual prices. The loss function broadly used across the project is the mean absolute error, or L1 Loss. This is due to the fact that it is less susceptible to outliers - sudden price hikes, very common in the electricity markets. However, in the beginning mean squared error is used as the loss function since it generally lowers the errors of the predictions and is widely used both for time series and regression [35].

The models tried out are as follows:

- *AR LSTM:* The first configuration corresponds to an autorregressive LSTM. This network takes the consecutive rolling windows and processes one sequence at a time, similar to an autorregressive model but with a neural network structure. The loss function for this model is the mean squared error.
- *Improved LSTM:* The second configuration breaks with the first architecture, though is not autorregressive anymore in the sense that processes all sequences which are then ”flattened”, processed together into the final non-linear transformation and outputted. Furthermore, this layer takes the mean absolute error as the loss function.
- *Improved LSTM with Input regularization:* The third configuration builds upon the *Improved LSTM*, since the previous model already showed considerable performance improvements. This model stands out for having both L1 and L2 input regularization, of order 0.001. The L1 regularization term allows the model to not be heavily affected by sudden price movements [41] by turning off the weights of those price hikes, whereas the L2 regularization restricts the model to over fit to specific values and be more precise in general terms.

The last configuration can be seen in the figure below, both on a programming perspective and a conceptual perspective.



(a) Tensorflow diagram of the Improved LSTM. (b) Conceptual diagram of the Improved LSTM.

Figure 3.16: LSTM model architecture.

3.4.4 Probabilistic Deep Learning

On to the final level of models, the probabilistic LSTMs. These models, which are neural networks with the ability to infer distributions from the data points to enable sampling from the predicted distributions, are very interesting because they add even more modelling flexibility to the deep learning models. Where we have been previously assuming that the uncertainty of our data was constant, this is, the variance of the historical data points was constant over time, the reality, as shown in the Exploratory Data Analysis 3.2, is that the time series are heteroscedastic and the variance of the electricity price has grown over time. In light of the last two years of macroeconomic instability, the energy production has been disrupted hence disrupting the electricity consumption and production, hence disrupting the price. This is key because the probabilistic LSTM models can account for the uncertainty intrinsic to the data Aleatoric Uncertainty, hence making the models more realistic although more difficult to fine-tune. Furthermore, when correctly fine tuned, these models should more appropriately generalise across regions. This is because, though from different energy sources, it is the same factors that affect the electricity prices in all regions.

Regarding the functioning of Probabilistic models, the models used in the project learn the parameters of a Gaussian through Tensorflow [22], they account for Aleatoric uncertainty, outputting the parameters of a Gaussian distribution, the mean and the variance of the predictions. With these parameters it is now necessary to sample the distributions and average those samples to get the predictions. It's interesting to note that calculating different percentiles at each time step gives a good understanding of how good the model is and how certain it is about the predictions. These models are optimized by the use of a new loss function, the negative log likelihood. This change of focus occurs because when working with probabilistic models, we want to maximize the likelihood that the inferred distribution is actually the real one, i.e: that the Independent and Identically Distributed (IID) [42] condition holds. By applying the natural logarithm and making it negative, we ensure the values are smoother, we make the computations significantly faster and we convert it back to a minimization problem, which is the standard approach in Machine Learning [43]. In the end, we are computing the maximum likelihood estimation by sampling from the distribution and then getting the negative log likelihood, which then defines how the model's weights are updated as in regular deep learning models with gradient descent. The negative log likelihood can be written down as:

$$L\Big(\theta\Big) = -\frac{N}{2}\Big(log(2\pi) + log(\sigma^2)\Big) - \frac{1}{2\sigma^2}\sum_i (x_i - \mu)^2 \tag{3.6}$$

Where the gaussian distribution, namely each output of the last layer, can be parametrised as:

$$\hat{\mu} = \frac{\sum_{i=1}^{N} x_i}{N}, \quad \text{sample mean} \tag{3.7}$$

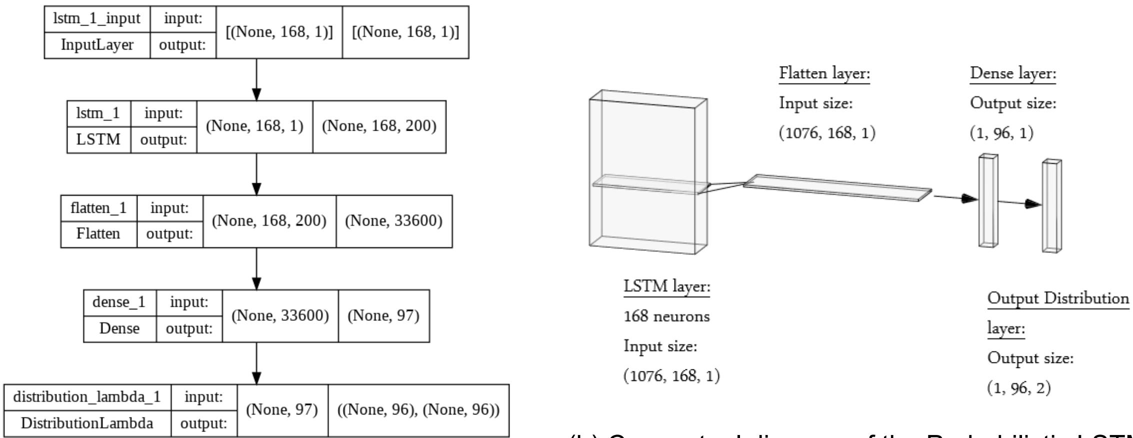
$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{N} (x_i - \hat{\mu})^2}{N}, \quad \text{sample variance} \tag{3.8}$$

$$\theta = \{\mu, \sigma\} \tag{3.9}$$

Regarding model architectures, there has also been three combinations tried out, all of which have been composed with a wrapper for the gaussian distribution as the final output layer, called Distribution Lambda layer. This layer needs to sample the priors and posteriors, where the initialization inserts uncertainty to the model, the same as in regular neural networks. The priors and posteriors of the distribution are defined to follow an Independent distribution composed from a batch of Normal distributions whose means and variance are defined by the bias size and smoothed so there is no drastic drifting throughout the learning process, i.e: the equivalent of the learning rate here is defined as a combination of stable mean and moderate variance so there are no big jumps across the latent space. Regarding the Variance of of the output distribution as well as the ones for the priors and weights, a softmax function is used to ensure the values are in the positive range.

The three combinations tried out are:

- *Probabilistic LSTM 1:* The first probabilistic model includes the above described Distribution Lambda layer and the same architecture as *Improved LSTM with Input regularization*.
- *Probabilistic LSTM 2:* The second probabilistic model builds upon *Probabilistic LSTM 1*, with the differences that it has a bigger number of LSTM units, 250, and there is more L2 regularization, 0.0015.The reason to increase the number of LSTM units is that the model is now able to incorporate uncertainty thanks to the output distribution layer, hence if the model does learn appropriately it can be the case that the increase in hidden units does not result in overfitting, as it can be the case when increasing the hidden layer's size. The regularization is to further restrict the overfitting.
- *Probabilistic LSTM 3:* The third probabilistic model builds upon *Probabilistic LSTM 1* too, this time the differences are with regards to the distribution which the model infers. Specifically, here the scale of the posterior (think learning rate of the sampling) is further smoothed from $1e^{-5}$ to $1e^{-4}$. Moreover, and similar to the increase of regularization in *Probabilistic LSTM 2*, the scale of the normal distribution in output layer is increased, to further smooth finding the variance of the distribution for each time step.



(a) Tensorflow diagram of the Probabilistic LSTM.

(b) Conceptual diagram of the Probabilistic LSTM.

Figure 3.17: Probabilistic LSTM model architecture.

3.5 Conclusion

The chapter has covered the components of the project, from the data set used and why forecasting the price is intrinsically complicated, to the framework defined to not introduce biases but also end up with the best forecasting models, to the algorithms finally used, why are they designed as they are and especially why the use of the negative loglikelihood is a change of paradigm when optimizing the deep learning model.

4 Results

4.1 Introduction

The following chapter addresses the results obtained throughout the experiments of the project. Not to confuse with Chapter 5, here the results of the experiments are shown explaining which part of the project they pertain to. The section has three parts:

- Model Performance for NO1 region: Determine which configurations are the best from each model family;
- Performance for all regions: How do the best models perform across regions for a specific test data set.
- Impact for Monta's users: How much money (if any) do they save with the use of the prediction algorithm.

As a side note, the experiments involve several regions and models, which not all are included in this chapter. To see other results, please see Appendix D.

4.2 Model Performance for NO1 region

This section addresses which are the best configurations out of the different types of models tried out for NO1 region. As a reminder, NO1's historical data is used to determine which configurations predict best, in a framework that compares the results across 20 periods of time. For NO1 region, the performance of the different models is summarized in figure 4.4. After comparing the error of the forecasts for each model, the ones that stand out are *Historical Average 2*, *SARIMA 3*, *Improved LSTM with Input regularization* and *Probabilistic LSTM 1*. These are the models that forecast better across the 20 periods for each type of model, and are the ones highlighted in the figure below:

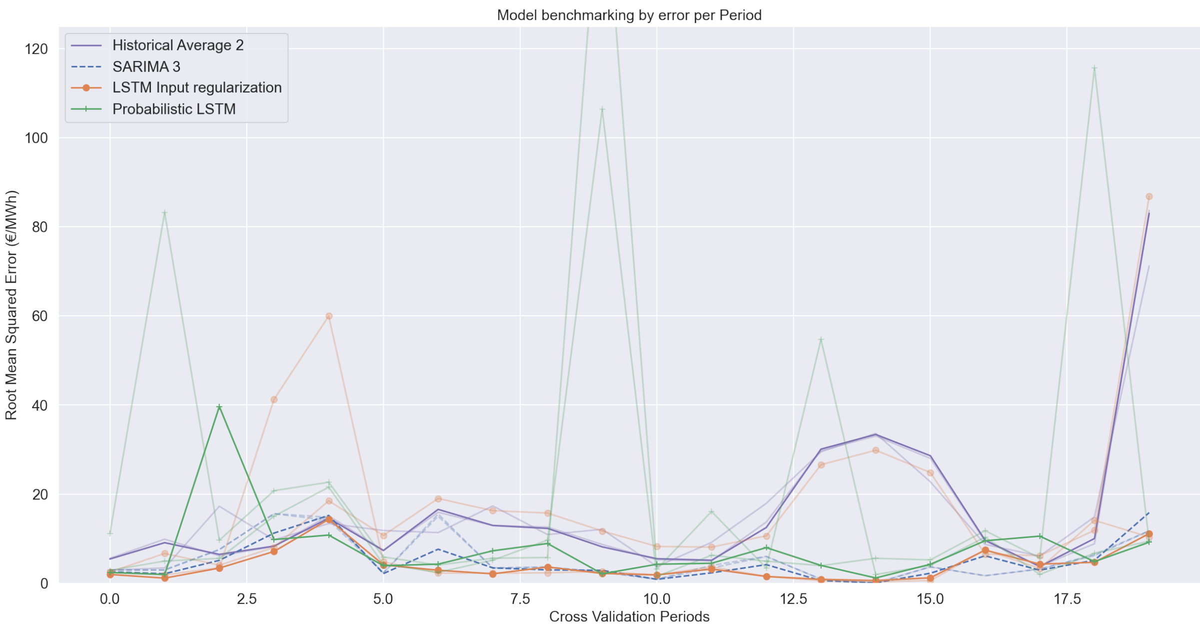


Figure 4.1: Error of each model at each period for NO1 region.

The forecasts of the price have not always been accurate so as to consider a model apt, as is the case for the second configuration of LSTM. Particularly, the model was very accurate in forecasting almost all periods, but there is one period with a sudden price hike that sets off the weights of the neural network and makes the gradients explode, resulting in bad predictions. An example of such bad predictions can be seen below:

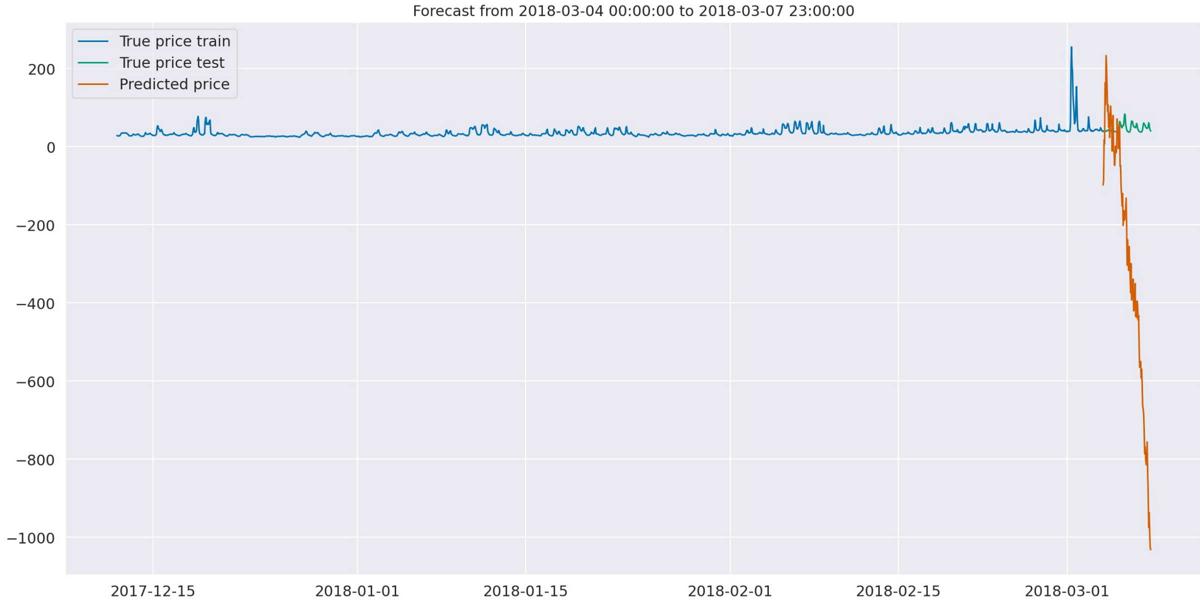


Figure 4.2: Forecasts with Improved LSTM.

The results shown above contrast drastically with the results with the same model architecture with regularization, which helps the weights of the model not to be carried away by the big price hike.

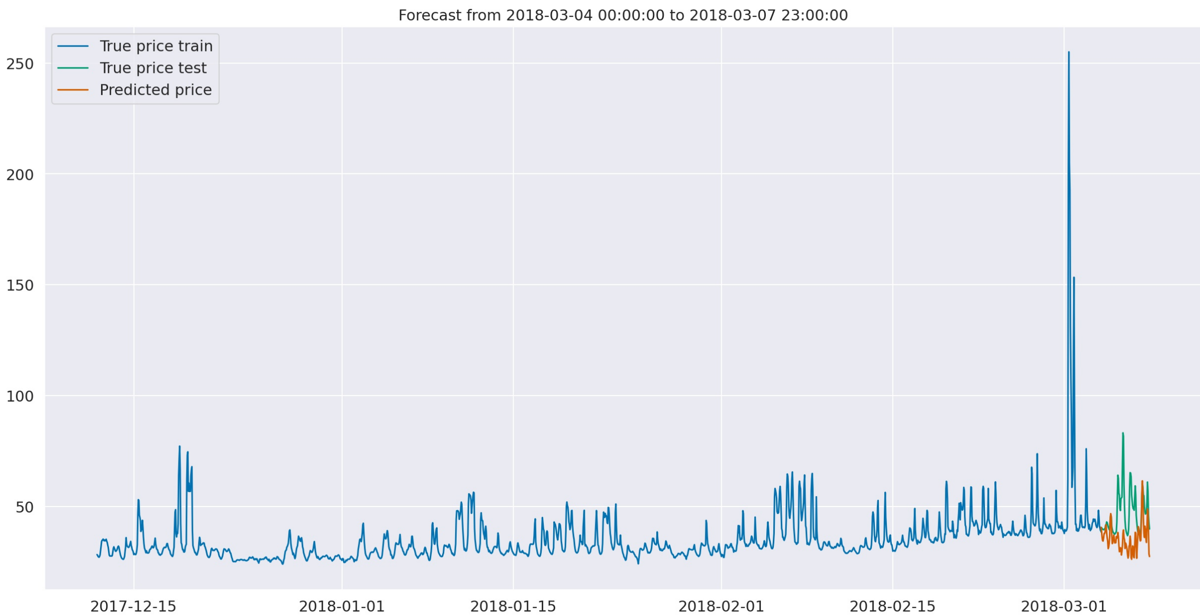


Figure 4.3: Forecasts with Improved LSTM with Input regularization.

The forecasting errors of the different models can be seen together in 4.4, where the best configurations of each type of model are highlighted. The error seen in the figure is the RMSE, which has been the main metric for evaluation. Nevertheless, the MAPE has also been calculated not to be biased in the evaluation of the forecasts. Precisely, the forecasting errors can be seen in detail in the table below:

Model	RMSE	MAPE
SARIMA(0,1,1 - 0,1,0,24)	5.439	15.768
SARIMA(1,1,0 - 0,1,0,24)	5.458	15.843
SARIMA(0,1,1 - 1,1,0,24)	**4.793**	**13.214**
Historical Average 1	16.406	182.029
Historical Average 2	**16.154**	**183.352**
Historical Average 3	16.168	181.768
AR LSTM	16.656	166.609
Improved LSTM	8.366	21.391
LSTM with Input regularization	**3.992**	**12.173**
Probabilistic LSTM	**7.579**	**29.028**
Probabilistic LSTM 2	17.393	63.769
Probabilistic LSTM 3	22.337	117.451

Table 4.1: Average error across periods for each model configuration.

From the above table it is clear the best forecasting model is *LSTM with Input regularization*. The RMSE of the forecasts can be seen over time in figure 4.4, where the best models are highlighted.

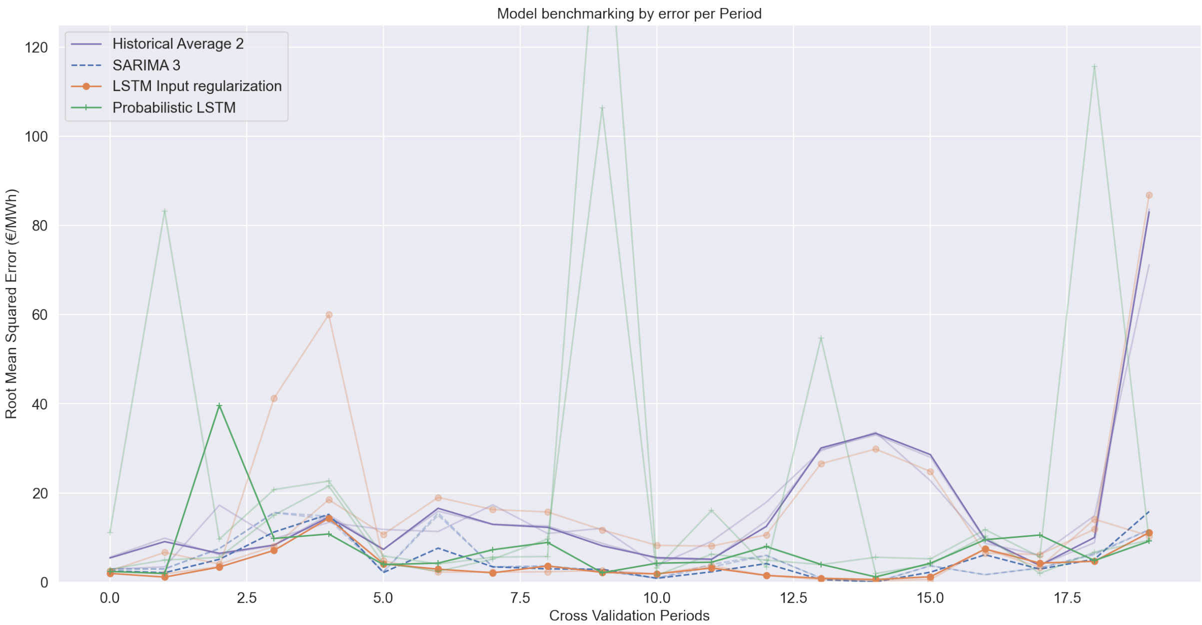


Figure 4.4: Error of each model at each period for NO1 region.

Moreover, in the context of Smart Charging it is interesting to see if the models perform best at a certain time of the day or specific days of the week. Therefore, the RMSE has also been calculated for the hour of the day, the day of the week and a combination of both. Firstly, figure

4.5 illustrates the forecasting error by hour of the day regardless the day of the week, on average for the 20 periods.

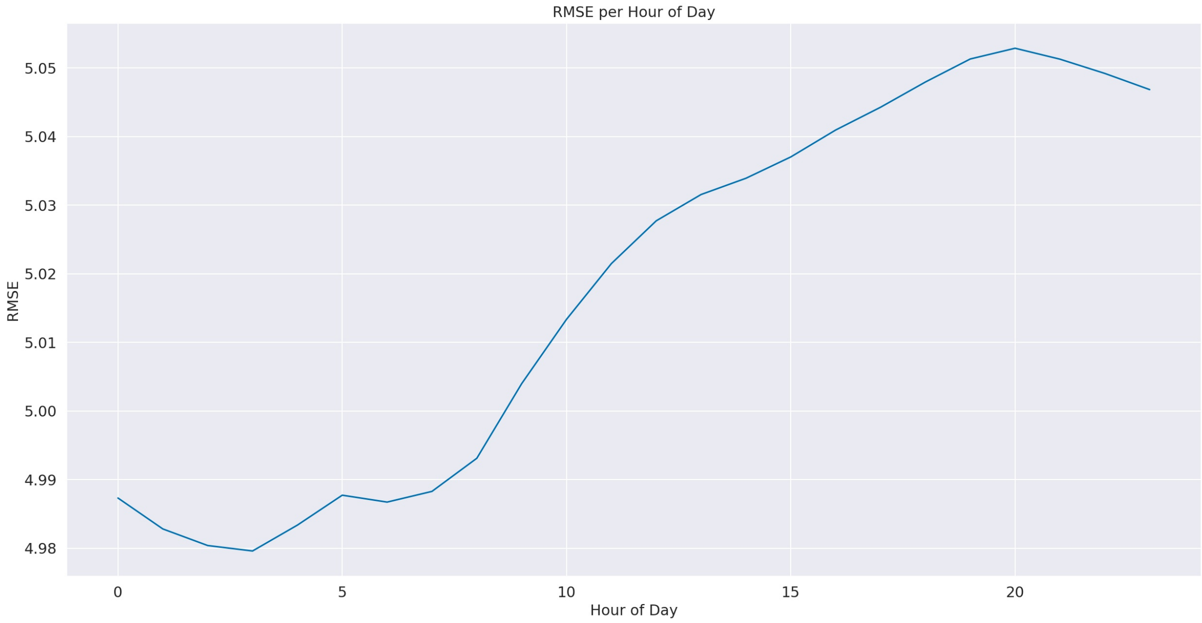


Figure 4.5: Error of best model by hour of the day.

Secondly, figure 4.6 illustrates the forecasting error for each day of the week regardless of the hour across the 20 periods:

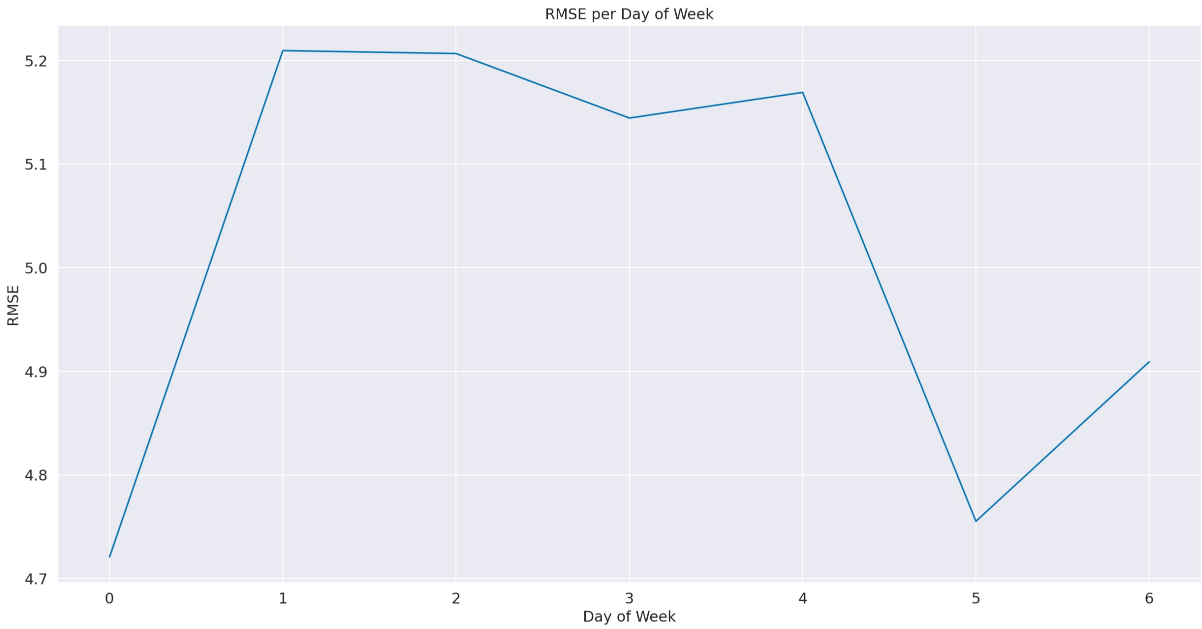


Figure 4.6: Error of best model by day of the week.

Thirdly, figure 4.7 illustrates the average forecasting error for each hour of the day for each day of the week across the 20 periods:

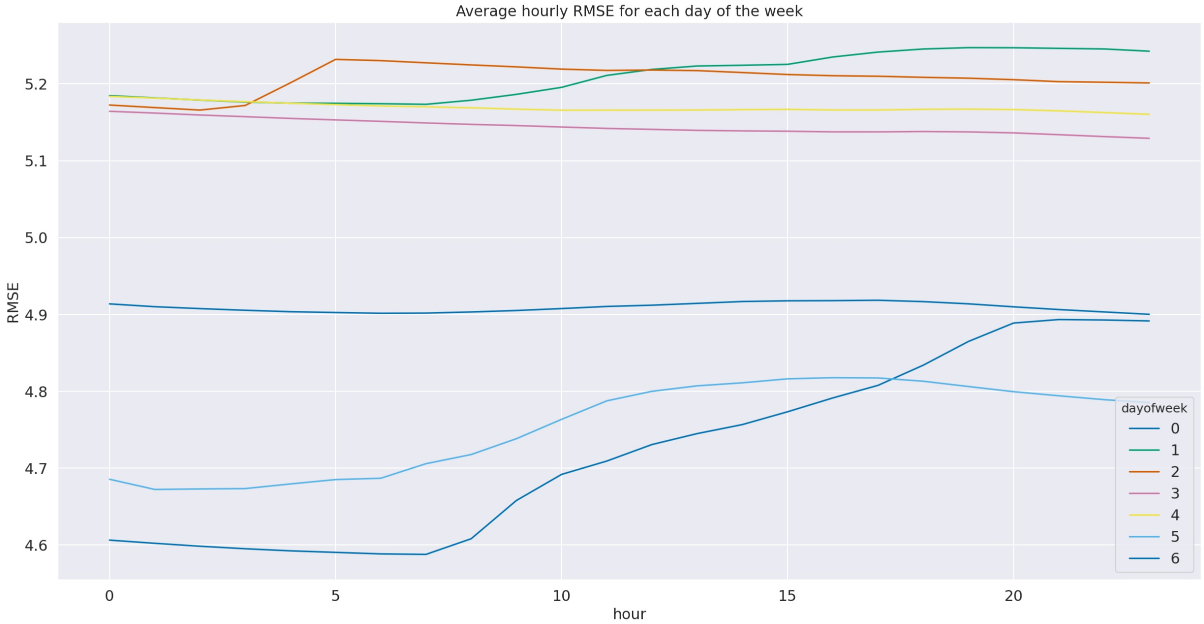


Figure 4.7: Error of best model by hour for each day.

To wrap up the initial forecasting of NO1 prices, it is important to address the learning of the models. The deep learning models iteratively learn by updating the weights at every epoch, which enables allows for evaluating the learning process of the algorithm. Given the variety of deep leaning configurations tried out, the focus is on the best model, the *Improved LSTM with Input regularization*. Along the design phase, several number of iterations are tried out to identify the optimal ”learning length” across the 20 periods. The learning process for the best model in the period with the sudden price hike can be seen in figure 4.8. The figure shows the MAE and MSE for every iteration of the model.

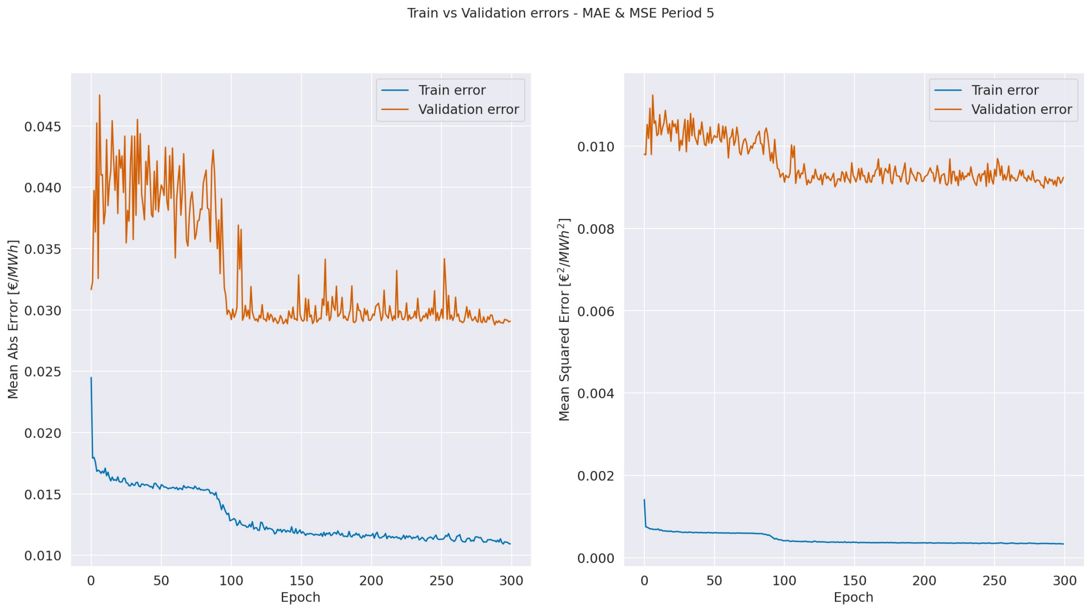


Figure 4.8: Training and Validation loss of LSTM model for the price hike period.

4.3 Performance for all regions

This section addresses how well the defined as best configurations predict across regions. The aim is to understand how extrapolable are the assumptions and conclusions drawn out from the study of the NO1 region to all other Scandinavian regions.

In this sense, the best models of each type are used to forecast the electricity prices for a defined test data set, the period from 09/03/2022 00:00:00 to 12/03/2022 23:00:00 across al regions. The models forecast the price of these four days, and the predictions are evaluated and analysed.

Regarding the results, the predictions of the different models can be seen in figure 4.9, which shows the predictions of the probabilistic LSTM for NO5:

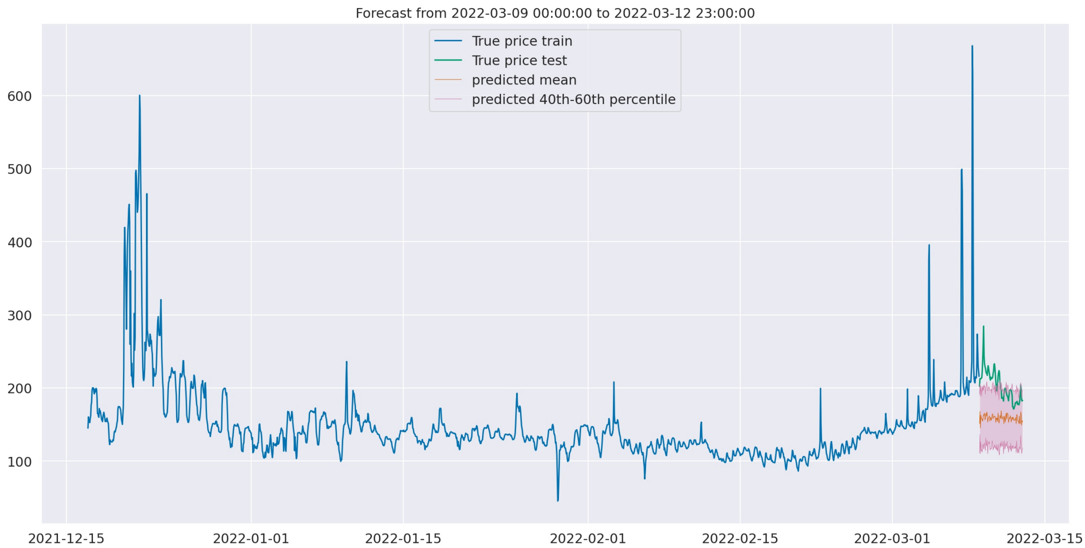


Figure 4.9: Forecast of electricity price using probabilistic LSTM for NO5.

It is important to notice that the evaluation of the models is conducted over a recent period in time to make it more realistic, though it happens to be very volatile times due to greater macroeconomic circumstances, hence the predictions have very high variance. The forecast errors per region can be seen in the table below:

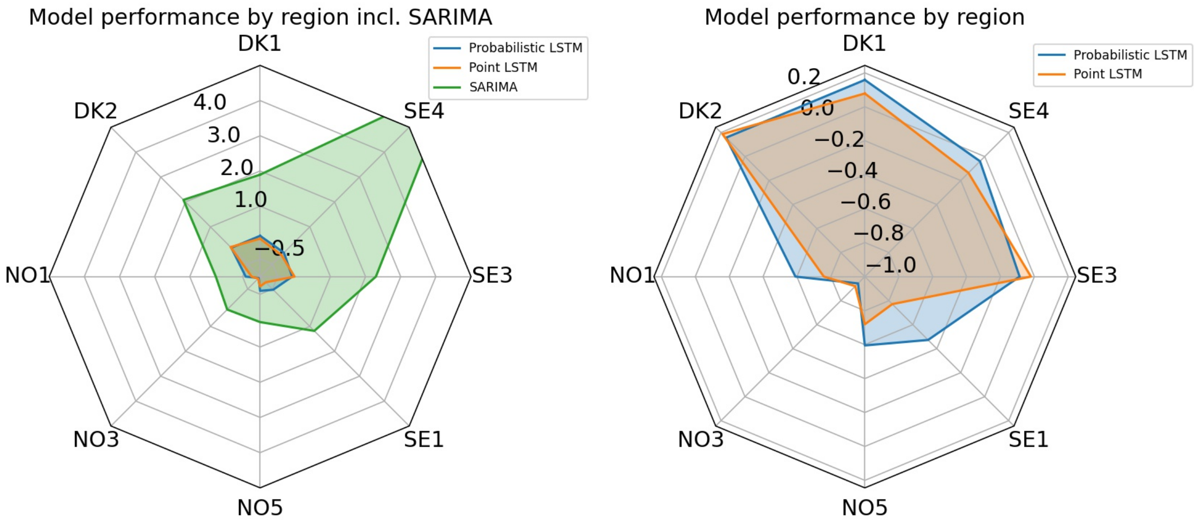
Region	Probabilistic LSTM	Historical average	LSTM	SARIMA
DK1	176.052	151.972	163.903	440.077
DK2	174.636	150.882	179.181	465.171
NO1	50.038	122.435	29.523	154.182
NO3	1.097	19.714	1.554	26.071
NO5	48.914	120.900	33.865	156.133
SE1	13.878	26.335	6.011	57.410
SE3	102.346	112.125	109.845	369.371
SE4	113.619	118.277	102.211	911.976
Average	85.073	102.830	78.262	322.549

Table 4.2: RMSE (€/KWh) by model across regions.

To further compare predictions, the average error of the predictions of each model is used to define a ratio of how much better a model is with regard to a historical average. The aim of this is to understand if it is worth to dedicate resources to forecasting and if so how relevant it is to do it with a certain type of methodology - time series models, LSTM, probabilistic LSTM. About the ratios, each data point is calculated as the incremental difference of the RMSE between each model and the historical averages. The ratio is defined as follows:

$$ratio_{model} = \frac{RMSE_{model} - RMSE_{H.A}}{RMSE_{H.A}} \cdot 100\% \tag{4.1}$$

This interesting point of view can be seen in figure 4.10. Here, the zero line represents the RMSE of the predictions using the historical average, and a value of $ratio$ = -1 means that the model reduces the error 100%. The results of the ratios for the different regions and models can be seen below:



(a) Model performance by region including SARIMA.

(b) Model performance by region.

Figure 4.10: Model performance across regions.

4.4 Impact on Monta's users

This section explains the impact the forecasting algorithms could have on Monta's users. Specifically, the aim is to determine how much would a user save if charging when the predictions indicate to do so as compared to immediately when plugging in the car.

It is necessary to remind that Monta's position is that of a facilitator. The company is not an operator nor an electricity producer and therefore has no interest in making the users pay more for electricity. The benefit Monta can have in Smart Charging is that users like the feature, are motivated by it and increase their use or maybe even push other people to start using Monta. With this in mind, when a person decides to charge their car and if they do not need it immediately, they can opt to use the Smart Charging feature, which enables them to select a time horizon in which the car needs to be charged. Within this time window, the car can be charged at any given time based on the electricity prices, as well as the source of electricity and the CO2 the production of the electricity generated. As a reminder, only the electricity price is within the scope of this project.

Regarding the forecasting models, it is interesting to validate them with regards to the users of Monta because is a straightforward way for the company to measure the ROI of the project.

Therefore, the predictions are evaluated based on how much money a user would save by charging when the forecasts say the price is lowest as compared to the savings from charging immediately when the car is plugged in. To measure this, the cost of charging is defined as the electricity price multiplied by the amount of battery needed times the speed at which it is provided by the CP. This cost is calculated for the moment the car is connected to the CP as well as for the moment which the forecasts say the price is lowest, so the user pays less to charge their car. The difference in both amounts is the money saved.

The calculations in particular assume a common car for all regions due to its popularity within Electric Carsharing companies, the Renault Zoe. The car has a battery of 41 KWh [44], which is assumed to be at a battery level of 15% when plugged in. It is assumed the charger is an average one, with a power supply of 50 KW. Moreover, the savings are calculated as the difference between cost of charging in the very first moment minus charging when the forecasted price is lowest, i.e: positive cost of charging means savings and negative means extra expenses for using the forecasts. The table below shows the results for the end user:

Region	DK1	DK2	NO1	NO3	NO5	SE1	SE3	SE4
LSTM	106.53€	9.21€	8.08€	1.04€	8.62€	2.28€	72.65€	78.30€
probabilistic LSTM	161.33€	182.16€	-22.67€	1.04€	99.59€	2.19€	78.58€	48.07€

Table 4.3: Money saved in smart charging a Renault Zoe based on the price forecasts.

4.5 Conclusion

The chapter collected the results obtained throughout the experiment for the reader to evaluate itself the performance. Particularly, the section covers an analysis centered on NO1 based on historical data from 2017 to 2021 included, an analysis solely on March 2022 across the different Scandinavian grid regions and finally an analysis of the savings a user would benefit from using the four-day-ahead forecasts of the electricity price .

5 Discussion

5.1 Introduction

This Chapter addresses the analysis of the results of the project. The aim is to make sense of the results and make conclusions that are insightful for the business.

Following the structure defined in Chapter 4, this section is focused first on evaluating models based on historical data for NO1. It follows by analyzing results of the best models across regions. Lastly, it studies how the forecasts would affect Monta's users.

5.2 Model Performance for NO1 region

Starting with the results of the historical data of NO1, these clearly show that the best model configuration is the LSTM with input regularization. It can be seen from table 4.1 that it has the lowest average RMSE and MAPE, which clearly indicates it outperforms the rest of configurations. Interestingly enough, SARIMA 3 comes very close in terms of the value of the errors. This shows the value of the time series analysis to make the models appropriate for the time series models, especially in terms of reaching stationarity. In terms of the error per period, and as discussed when talking about regularization, it is seen that this technique plays a big role in making the neural network predict aptly. The sudden price hike seen in figure 4.2 makes the gradients explode and predict poorly. On the other hand, the regularization term of the model, especially L1 regularization, enables the model to skip those exploding gradients by turning the weights to zero, hence not being overinfluenced by the price hike. L2 regularization ensures the model is not overfitted and allows for the model to be trained longer (more iterations) [39]. The results clearly show the AR-LSTM approach is not as effective when forecasting these long sequences (168 hours), and as per recommendation of the co-supervisor Frederik Boe Hüttel, the regular LSTM approached turned out to forecast better. The regularization contributed to training for longer iterations without overfitting. A good prove of the effects of regularization is figure 4.8. This figure shows the training and validation loss for the period that has the price hike generating the exploding gradients. It can be seen that, when the model is trained for 300 epochs, it still improves the performance though it is stabilizing. Interestingly, the learning improves significantly around the 100th epoch. These long trainings resulted in overfitting without the use of regularization.

Keeping the focus in table 4.1, its clear that the initial configuration of the probabilistic LSTM is the best performing one. The changes made to the configurations 2 and 3 are related to the learning length and regularization and the scale of the variance of priors and posteriors respectively. The intention behind the changes was to enable longer training without overfitting, which didn't result in better results; and to reduce the variance of the sampling of the parameters of the Normal distribution, which further worsened the forecasts. It can be concluded that smaller variance of the sampling posterior restrains the model from achieving the correct weights and optimizing for better results.

Regarding the errors per period of the different models, figure 4.4 shows there are varying ranges of error depending on the type of model for each period. For instance, in period 13 the best historical average model has considerable lower performance than the rest. It is also the case for period 5 where there is an LSTM model that has a bigger value than any other model, it is the AR-LSTM. On a different note, the probabilistic models can have sudden worse performance for a specific period without it meaning the model is considerably worse for that period. These models output the mean and variance of the predictions, which is sued to sample from a distribution with such mean and variance, thus the values sampled are then averaged and that are the forecasts of price. Because of this aleatoric component introduced with the sampling, it can be the case that it there is not a direct relationship between a bad performance at a period meaning the mdoel is not suitable for such period. To take care of this, following the Law of Large Numbers, the price is sampled 1000 times and averaged so that the price reflects significantly the real price. Taking this into account, at period 9 the probabilistic LSTMs 2 3 have big errors and it can be concluded that the models are not forecasting appropriately those periods of time thanks to having sampled a high number of times.

Regarding the error by hour, looking at figure 4.5 brings light that the dawn of the day is the best moment to charge. The predictions are significantly better from 00:00 am to 08:00 am, which means there is potential for effective Smart Charging over the night. On the contrary, the evening is the worst time to smart charge based on the price forecasts. This is in line with the increase in the use of electricity in the afternoon and evening, since it is necessary for illumination and people not at work make use of it for household duties. Regarding the error by day of the week, it can be seen that Mondays, Saturdays and Sundays are significantly better predicted than the other weekdays. In line with the assumptions related to the use of electricity, the use generally reduces during the weekends since illumination of workplaces is not needed. These results and hypothesis can be validated with the errors by hour of the day and day of the week, figure 4.7. Interestingly, the hypothesis that forecasting the price for the dawn of the day holds across all days of the week except Sundays, which remains pretty stable across the whole day.

5.3 Performance for all regions

The results of the historical data for NO1 show that the best configurations of each type of model are *Historical Average 2*, *SARIMA 3*, *Improved LSTM with Input regularization* and *Probabilistic LSTM 1*. These models are used to forecast the price of the test window, ranging from 09/03/2022 to 12/03/2022. The forecasts are for all regions, and the errors are used to determine which models generalize the best. The errors also help bring light to the functioning of the nordic electricity markets for future research.

Regarding the errors across regions, it is inportant to bear in mind that the results that appear in 4.2 are in €$/KWh$. This is, the SARIMA model, which was the second best model for the time series prediction when the model was design specifically for the historical data, fails to predict the price of the out of sample data. Moreover, the forecasts are, on average, wrong by 322€$/KWh$, which means the model is nowhere near being apt for forecasting the price.

On the other hand, the LSTM and the probabilistic LSTM, respectively, deviate on average 78 €$/KWh$ and 85 €$/KWh$ across regions. These, far from being desirable, are much more acceptable values taking into account the heteroscedasticity of the different series, i.e: how the price of each region varies differently over time, especially for Q1 2022 with the Russia-Ukraine conflict, Supply Chain shortages and other macroeconomic factors which, though shouldn't, do interfere with the price of electricity. An interesting insight that can be drawn out of the results from table 4.2 is that forecasting the price is a significantly easier problem in the northern Scandinavian regions, such as NO3 and SE1. This is due to the fact that the price in this regions is very dependant on cheap and renewable energy sources which make the prices to be more stable all year round. This opens up the following question, which is out of the scope of the project: Are those prices innately stable, or is making them stable disrupting the market for other regions?

Coming back to the results of the experiments, perhaps the most illustrative results for comparing model performance are displayed in figure 4.10. These figures show how good the models are in terms of RMSE as compared to regular historical averages, particularly *Historical Average 2*. It shows that both the LSTM and the probabilistic LSTM outperform the historical averages, with more or less accuracy across regions. Notably, the regions that have more stable prices are the ones where these models outperform the most the simpler models. The SARIMA, however, not only it does not perform better than the historical averages but it does so having errors to the factor of 6x.

All in all, this experiment clearly shows forecasting electricity prices is a complex problem, though one that using the correct approaches - LSTM, probabilistic LSTM - can result in good forecasts to incorporate into a business.

5.4 Impact on Monta's users

End users are the reason why Monta is interested in the project in the first place. Therefore, it comes as no surprise that the forecasts need to be evaluated then with regards to Monta's users. To do this, assuming a specific model and charger, it is possible to calculate the savings of charging profiting from the price forecasts as oppposed to charging immediately.

This analysis brings light to the benefits of Smart Charging: regardless of the region, a user would save money when charging based on the forecasts of the LSTM, and almost in every region when charging based on the forecasts of the probabilistic LSTM. In fact, the probabilistic LSTM has proven to be better at telling if the time is going to be lower, this is, whether to wait for the charge or not. Generally, the probabilistic LSTM's forecasts result in greater savings than the LSTM's, with the great example of DK2, where charging when the probabilistic LSTM forecasts lower prices results in savings of 182.16€, as compared to 9.21€ with the LSTM forecasts. The increased cost of 22.67€ for NO1 region means that the forecast was inaccurate when predicting lower prices. This is an issue to look after since it would directly affect the users and it has come up in a region where the price is not the most varying nor heteroscedastic.

5.5 Conclusion

The chapter has detailed the insights obtained from analysing the chapter 4. Specifically, it has been detailed that SARIMA models reach apt levels of performance when correctly modelled and customized, comparable to the level of moderately tuned LSTM and probabilistic LSTM models. However, it has been shown that the latter models generalize much better and are able to forecast appropriately in other regions, so to say, for other time series than the ones the model had been designed for. Moreover, it has been explained that deep learning models significantly improve predictions as compared to simple historical averages, setting the path as to which kind of models are really interesting to use in the case of further integrating forecasting into Monta's operations. Furthermore, it has been discussed that both LSTM and probabilistic LSTMs are good models for Smart Charging in terms of money saved by the customer 4.3.

6 Conclusions

Along the project a framework to determine what is the best model for forecasting electricity prices has been developed. The methods allow to forecast electricity prices 96 hours ahead - 4 days - which allow Monta to offer their users Smart Charging based on longer horizons than what's currently possible.

The project starts with a detailed study of the series, the electricity price. Given the multiple series for each region, the project focuses in a time series in particular, NO1, to conduct further time series analysis and adequately design the time series models. Thereof, along the project a framework to end up with the best forecasting models is designed without introducing biases into the predictions. These framework builds on the study of the data to forecast over 20 periods and evaluate the forecasts in different periods. The framework tests out different configurations of 4 types of models and helps determine which are the most suited for electricity price forecasting. Along the way, it has been explained why the best models are considered so and what was taken into consideration when designing them. Particularly, it has been explained why the decision to use deep learning models and why the MAE as a loss function, and even more why to use probabilistic deep learning and how the use of the negative loglikelihood is a change of paradigm when optimizing the deep learning model. Moreover, the project continued analyzing how the models designed for NO1 behave when forecasting other regions' price. It has been shown that the deep learning models generalize much better and are able to forecast appropriately in other regions. Furthermore, it has been explained that deep learning models significantly improve predictions as compared to simpler forecasting models, removing any doubts of which kind of models to be used when further integrating forecasting into Monta's operations. Finally, the project also covers how the forecasting of price could affect Monta's users in terms of money saved per charge. It has been shown, though in a test-scale, that these forecasts (in)accuracy when predicting the prices is not an issue as long as the price predicted is indeed lower than the initial plug-in price, hence making it interesting to wait for charging.

The results clearly show enlarging the Smart Charging window will clearly benefit Monta's users individually, but what would happen when Monta has a big user base, big enough to perhaps overload the grid? When Monta's expansion becomes a reality and its users profit from the Smart charging, it is necessary to make further analysis of how would that affect the balancing of the grid. Would it be ethical to do smart charging then? It could be fairly easy to limit the impact, however it needs to be taken into consideration. What is more, what would be the consequences (extra C02, missed use, overoccupation of the CPs) of such overloading?

The project has covered the desired theoretical concepts, such as time series analysis, the use of deep learning for time series, as well as the use of probabilistic deep learning. It is still far, however, from accurate predictions that could be incorporated into robust and scalable smart charging. In this sense, it would be interesting to further improve the models, especially the probabilistic LSTM, which has not been fully fine-tuned. Moreover, it could be interesting to develop transformer-based solutions, perhaps such as in [45]. In line with the interest in probabilistic deep learning, the latest models developed in the project approximate the series to Normal distributions, which comes from the fact that it is simple and one of the most common distributions, so it's a good starting point. However, the data is not being fully characterized by the distribution, as can be determined by the fact that the forecasts are not depicting the real behaviour of the data. It would be interesting to also account for epistemic uncertainty, inherent to the model. This would be done learning a distribution over the weights using variational inference, which can be done similar to how the probabilistic LSTM accounts for aleatoric uncertainty. Furthermore, and perhaps the mos interesting future point of work, would be to make the model predict simultaneously all the series. The model could learn the joint distribution of the predictions and hence the price of one series would be influencing the price of other, as happens in reality because of the imports and exports of power between regions. Stepping away from changes of the models, an interesting aspect would have been to have available data of exogenous variables. In the ideal case, having forecasts of say, wind power production, can be used to modify the deep learning architectures so that these exogenous series help the model learn backwards whereas the historical values of the price make the model learn forward, which would be a modification of a bidirectional LSTM [46]. This is however far from achievable with the current information, first and foremost due to the lack of historical data of the exogenous variables, let alone the historical forecasts of those. It would be an interesting topic that will surely be explored in the future.

Bibliography

[1] *EV Smart Charging Explained*. https://monta.com/uk/blog/ev-smart-charging-explained/. Accessed: 2022-06-08.

[2] Stefan Cronholm, Sarah Guss, and Vince Bruno. "Learning Observation–Introducing the Role of a Meta-Observer". In: (2006).

[3] Elaheh Fata, Igor Kadota, and Ian Schneider. "Comparison of Classical and Nonlinear Models for Short-Term Electricity Price Prediction". In: *arXiv preprint arXiv:1805.05431* (2018).

[4] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[5] Jesus Lago et al. "Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark". In: *Applied Energy* 293 (2021), p. 116983.

[6] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[7] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.

[8] *Cross-Validation in Time Series*. https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4. Accessed: 2022-02-14.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[10] Sebastian Raschka, Yuxi Liu, and Vahid Mirjalili. *Machine Learning with PyTorch and Scikit-Learn*. 2022.

[11] *Deep Learning course from New York University*. https://atcold.github.io/pytorch-Deep-Learning/en/week06/06-3/. Accessed: 2022-03-31.

[12] *Keras Layer Documentation*. https://keras.io/api/layers/. Accessed: 2022-04-14.

[13] *How to Develop LSTM Models for Time Series Forecasting*. https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/. Accessed: 2022-04-15.

[14] Sourav Malakar et al. "Designing a long short-term network for short-term forecasting of global horizontal irradiance". In: *SN Applied Sciences* 3.4 (2021), pp. 1–15.

[15] Grzegorz Marcjasz, Jesus Lago, and Rafał Weron. "Neural networks in day-ahead electricity price forecasting: Single vs. multiple outputs". In: *arXiv preprint arXiv:2008.08006* (2020).

[16] Owen Q Wu, Şafak Yücel, and Yangfang Zhou. "Smart charging of electric vehicles: An innovative business model for utility firms". In: *Manufacturing & Service Operations Management* (2021).

[17] Morsy Nour et al. "Smart charging of electric vehicles according to electricity price". In: *2019 international conference on innovative trends in computer engineering (ITCE)*. IEEE. 2019, pp. 432–437.

[18] Mathijs De Weerdt, Michael Albert, and Vincent Conitzer. "Complexity of scheduling charging in the smart grid". In: *arXiv preprint arXiv:1709.07480* (2017).

[19] Yarin Gal. "Uncertainty in Deep Learning". In: 2016.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[21] Christopher M Bishop. “Model-based machine learning”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1984 (2013), p. 20120222.

[22] *Distribution Lambda Layer Documentation*. https://www.tensorflow.org/probability/api_docs/python/tfp/layers/DistributionLambda. Accessed: 2022-05-14.

[23] Eyke Hüllermeier and Willem Waegeman. “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods”. In: *Machine Learning* 110.3 (2021), pp. 457–506.

[24] *Dense Variational Layer Documentation*. https://www.tensorflow.org/probability/api_docs/python/tfp/layers/DenseVariational. Accessed: 2022-05-14.

[25] Ankush Ganguly and Samuel WF Earp. “An Introduction to Variational Inference”. In: *arXiv preprint arXiv:2108.13083* (2021).

[26] Francisco Pereira Filipe Rodrigues. “Lecture 10: MCMC methods”. In: *Model-based Machine Learning—DTU Course No. 42186*. MLSM DTU. Kongens Lyngby Hovedstaden, Denmark, 2021. URL: https://docs.google.com/document/d/1faI7hPqf6sYb3GkIaqI_W0Zj-YCf21m8TL0hMPRfrJk/edit#.

[27] Francisco Pereira Filipe Rodrigues. “Lecture 7: Temporal Models”. In: *Model-based Machine Learning—DTU Course No. 42186*. MLSM DTU. Kongens Lyngby Hovedstaden, Denmark, 2021. URL: https://docs.google.com/document/d/1faI7hPqf6sYb3GkIaqI_W0Zj-YCf21m8TL0hMPRfrJk/edit#.

[28] Ole Winther. “Lecture 7: Un- and semi-supervised learning”. In: *Deep Learning—DTU Course No. 02456*. DTU Compute. Kongens Lyngby Hovedstaden, Denmark, 2021. URL: https://docs.google.com/document/d/1GsP-G1NkfUEPV7J3c-JevdAorPedI1KgIN69dffC2Rk/edit.

[29] Nam Nguyen and Brian Quanz. “Temporal latent auto-encoder: A method for probabilistic multivariate time series forecasting”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 10. 2021, pp. 9117–9125.

[30] Marcos Ivorra Peleguer. *Forecasting of Electricity Prices using Probabilistic Deep Learning*. 2022. URL: https://github.com/Marcoscos/Monta_project.

[31] John Von Neumann and Oskar Morgenstern. “Theory of games and economic behavior, 2nd rev”. In: (1947).

[32] F.S Wen and A.K David. “Strategic bidding for electricity supply in a day-ahead energy market”. In: *Electric Power Systems Research* 59.3 (2001), pp. 197–206. ISSN: 0378-7796. DOI: https://doi.org/10.1016/S0378-7796(01)00154-7. URL: https://www.sciencedirect.com/science/article/pii/S0378779601001547.

[33] John Bower. “Modelling and Managing Competitive Electricity Markets”. In: *European Electricity Markets Structure and Trading*. OIES. Oxford United Kingdom, 2003. URL: https://a9w7k6q9.stackpathcdn.com/wpcms/wp-content/uploads/2011/02/Presentation5-EuropeanElectricityMarketsStructureandTradingI-JBower-2003.pdf.

[34] Yoshua Bengio and Nicolas Chapados. “Metric-based model selection for time-series forecasting”. In: *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*. IEEE. 2002, pp. 13–22.

[35] Tianfeng Chai and Roland R. Draxler. “Root mean square error (RMSE) or mean absolute error (MAE)”. In: *Geoscientific Model Development Discussions* 7 (2014), pp. 1525–1534.

[36] Adam Hayes. *Definition and origin of ARIMA*. 2021. URL: https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp.

[37] Pragati Baheti. *Guide of Neural networks architecture*. 2022. URL: https://www.v7labs.com/blog/neural-network-architectures-guide.

[38] Anuradhi Malshika Welhenge and Attaphongse Taparugssanagorn. “Human activity classification using long short-term memory network”. In: *Signal, Image and Video Processing* 13.4 (2019), pp. 651–656.

[39] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

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

[41] Michael Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[42] JR Blum et al. “Central limit theorems for interchangeable processes”. In: *Canadian Journal of Mathematics* 10 (1958), pp. 222–229.

[43] Francisco Pereira Filipe Rodrigues. “Lecture 1: Probability and Satatistic Review”. In: *Model-based Machine Learning—DTU Course No. 42186*. MLSM DTU. Kongens Lyngby Hovedstaden, Denmark, 2021. URL: https://docs.google.com/document/d/1faI7hPqf6sYb3GklaqI_W0Zj-YCf21m8TL0hMPRfrJk/edit#.

[44] *Eelectric Vehicle Database*. https://ev-database.org/cheatsheet/useable-battery-capacity-electric-car. Accessed: 2022-06-14.

[45] Bryan Lim et al. *Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting*. 2019. DOI: 10.48550/ARXIV.1912.09363. URL: https://arxiv.org/abs/1912.09363.

[46] Yiyuan Chen et al. “BRIM: An accurate electricity spot price prediction scheme-based bidirectional recurrent neural network and integrated market”. In: *Energies* 12.12 (2019), p. 2241.

[47] W.W. Daniel. *Applied Nonparametric Statistics*. Duxbury advanced series in statistics and decision sciences. PWS-KENT Pub., 1990. ISBN: 9780534919764. URL: https://books.google.dk/books?id=0hPvAAAAMAAJ.

[48] J. Leon. *LaTEX diagram of LSTM*. 2018. URL: https://tex.stackexchange.com/questions/432312/how-do-i-draw-an-lstm-cell-in-tikz.

A Multivariate Data Analysis

Price of electricity is not only driven by factors related to the energy sources from which it is generated, but also by the interest of a small group of producers and the agenda of governments. This translates into a time series that is highly multivariate, with a high uncertainty component that cannot be modelled due to its human factor origin.

Nevertheless, it is interesting to look at the series as a multivariate time series and understand which variables move the spot price in one direction or the other. For the selected initial region, data was collected from ENTSO-e through ElectricityMap's API, having data related to the price at one region, the energy sources, the CO2 generated, the imports and exports, as well as the forecasted consumption. As in all modelling problems, the aim was to better predict - in this case the electricity price for a region - with the simplest model that correctly captures the series dynamics. Therefore, the aim is to select the columns that most represent the variance in the spot price series. Out of the many modelling techniques, and because this analysis is not to be included to the main work of this study, the relevant features are identified through Spearman rank correlation [47]. In this case, the reason to select Spearman correlation over Pearson correlation, for instance, is that not all relationships across the variables are linear. An example is how energy imports from NO1 are in reality not linearly correlated to electricity prices in DK2, since this exports can come from long-term agreements or from energy needs of a specific day. With this, the Spearman correlation is calculated for every feature and the spot price and the most significant ones are kept, which can be seen in table A.1.

Feature	Spearman Correlation
total_production_avg	0.253202
total_import_avg	0.233728
power_production_hydro_avg	0.238619
power_consumption_gas_avg	0.246555
latest_forecasted_production_avg	0.240974

Table A.1: Spearman correlation of exogenous features with price, NO1 region.

Moreover, it is interesting to look at how the different features are distributed together, their density estimation. Looking at figure B.1 it can be seen that there are features highly correlated between them, such as the total_production_avg and the latest_forecasted_production_avg, or the former with power_production_hydro_avg. Thinking about it these are reasonable since Norway is heavily reliant on hydro power, therefore being very correlated with the total power production, which is of course correlated to the latest forecast, which is only one hour in advance of the real time. Furthermore, it is intriguing to see that the distribution followed by Total_import_avg is positively skewed. Total_import_avg represents the total energy imported for that hour from the neighbouring regions, so the fact that it is positively skewed means the region regularly imports big amounts of energy and more often than not ends up importing more than 5000MWh from neighbouring regions.

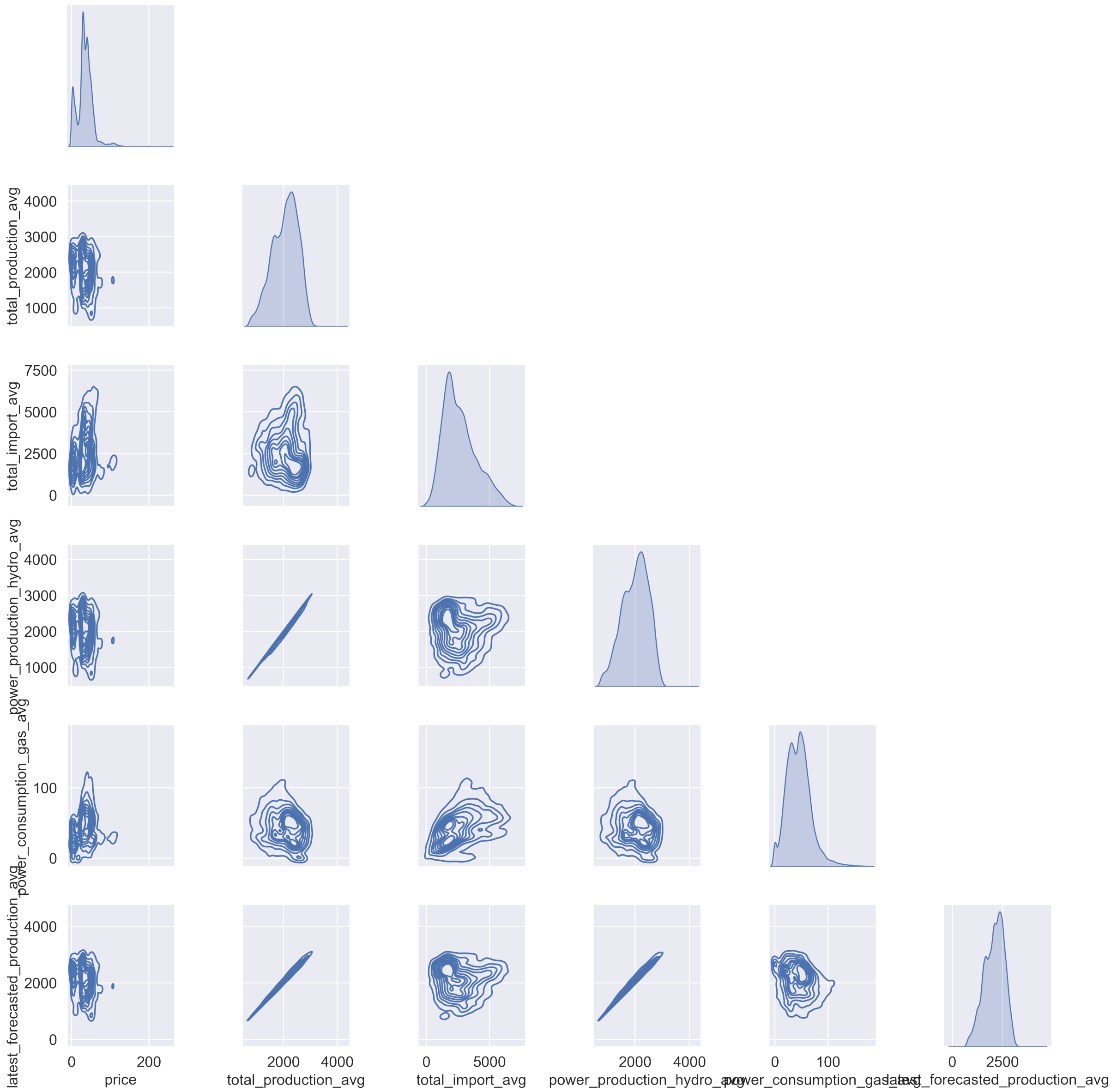


Figure A.1: Distribution and KDE of exogenous variables.

B Classical Time Series Models

SARIMA models are a variation of ARIMA models, which are as well an integrated combination of moving averages and autoregressive models. SARIMA takes into account the seasonality of the time series not only when differencing but also when composing the AR or MA terms of the model. The SARIMA general architecture will help understand how the models defined in 3.4 are composed. These models have non-seasonal and seasonal terms:

$$\text{ARIMA} \quad \underbrace{(p,d,q)}_{\uparrow \atop \text{Non-seasonal part of the model}} \quad \underbrace{(P,D,Q)_m}_{\uparrow \atop \text{Seasonal part of the model}}$$

Figure B.1: Seasonal ARIMA Model composition. Source: [4]

About the formulation of the model, the 7 parameters (p,d,q)(P,D,Q)m are defined as:

- p: the non-seasonal lag order
- d: the non-seasonal degree of differencing
- q: the non-seasonal moving average order
- P: the seasonal lag order
- D: the seasonal degree of differencing
- Q: the seasonal moving average order
- m: the number of observations per year, i.e: seasonality.

Looking further into the mathematical composition of the model, the equation for SARIMA 3 is:

$$(1-\Phi_1 B^{24})(1-B)(1-B^{24})y_t = (1+\Theta_1 B)\varepsilon_t \tag{B.1}$$

- B represent Lags, this is, y_{t-1};
- Φ_1 = seasonal autoregressive coefficient, obtained after inference;
- Θ_1 = moving average coefficient, obtained after inference;
- ε_t = y_t - y_{t-1}.

C Long Short-Term Memory

Long Short-Term Memory (LSTM) neural networks are conceived to solve an issue introduced by the Recurrent Neural Network (RNN) with long-term dependencies. LSTMs have a feedback connection that enables them to process entire sequences of data by keeping the useful information of each of the previous data processed in the sequence. Figure C.1 shows how a LSTM block is structured in order to have the property previously introduced. It is composed of 3 gates with different actions and a fourth block that behaves like a memory:

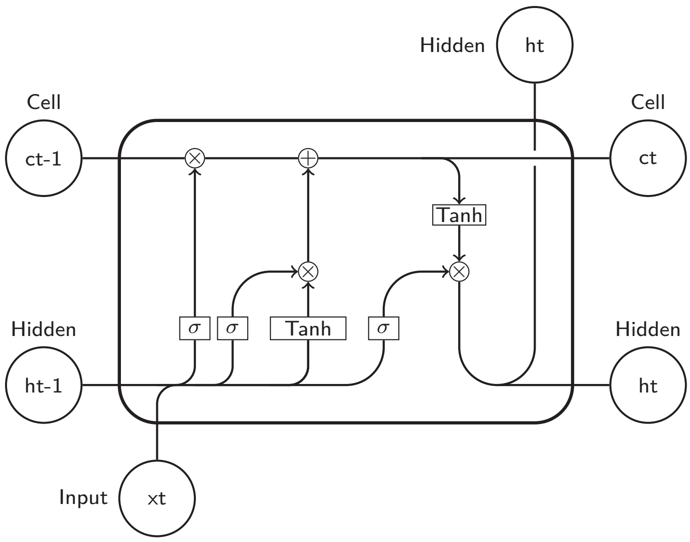


Figure C.1: LSTM unit diagram. Source: [48].

- *Forget gate:* this is the first gate of the LSTM block where it is decided which pieces of long-term memory should be forgotten at each stage. The previous hidden state and the new input data are feed into the neural network. In this gate, it will be decided which information of the *cell state* is useful given the previous hidden state and the new input data. In the diagram above it is the first vertical section starting from the left, f_t.

$$f_t = \sigma\Big(x_t U^f + h_{t-1} W^f\Big) \tag{C.1}$$

- *Input gate:* the goal of this gate is to decide which information of the new input data should be added to the long-term memory. As in the previous gate, this decision is also given the previous hidden state and the new input data. It is represented with i_t.

$$i_t = \sigma\Big(x_t U^i + h_{t-1} W^i\Big) \tag{C.2}$$

- *Output gate:* it decided the new hidden state. It is using the information from the new updated cell state, the previous hidden stated and the new input data. It is represented with o_t.

$$o_t = \sigma\Big(x_t U^o + h_{t-1} W^o\Big) \tag{C.3}$$

$$h_t = \tanh\Big(C_t\Big) * o_t \tag{C.4}$$

- *Cell state:* it carries the long-term memory of the network by encoding the aggregation of useful data from all the previous time-steps that have been processed. In the diagram below it is represented by the C_{t-1} to C_t section.

$$\hat{C}_t = \tanh\Big(x_t U^g + h_{t-1} W^g\Big) \tag{C.5}$$

$$C_t = \sigma\Big(f_t * C_{t-1} + i_t * \hat{C}_t\Big) \tag{C.6}$$

D Other results

This section covers the results of the study that are not displayed in Chapter 4. Since the project aims to do a benchmark of models for NO1 and evaluate the generalisability to other regions, the vastness of results is too broad to include in a single section and comment deeply on all of them. The results included here are grouped within the same categories as in the original Chapter, though there are no extra results with regard to the impact on the end users.

D.1 Model Performance for NO1 region

This section holds together results from the first set of experiments, that determine the best configurations forecasting in NO1.

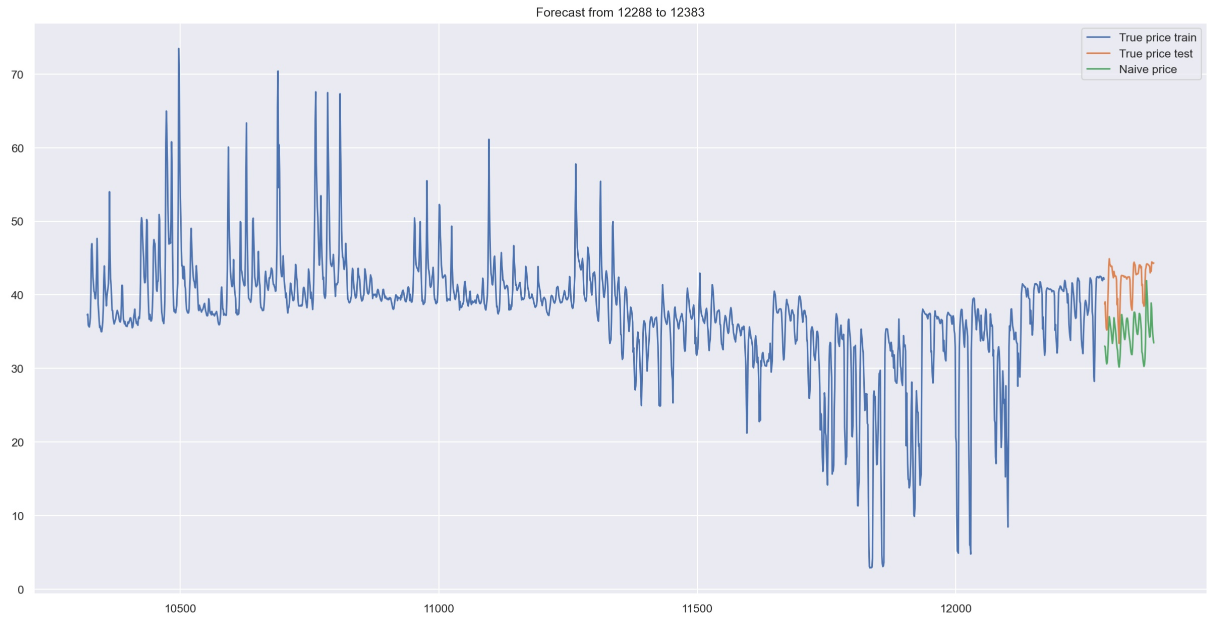


Figure D.1: Historical Average 2 forecasts for period 6.

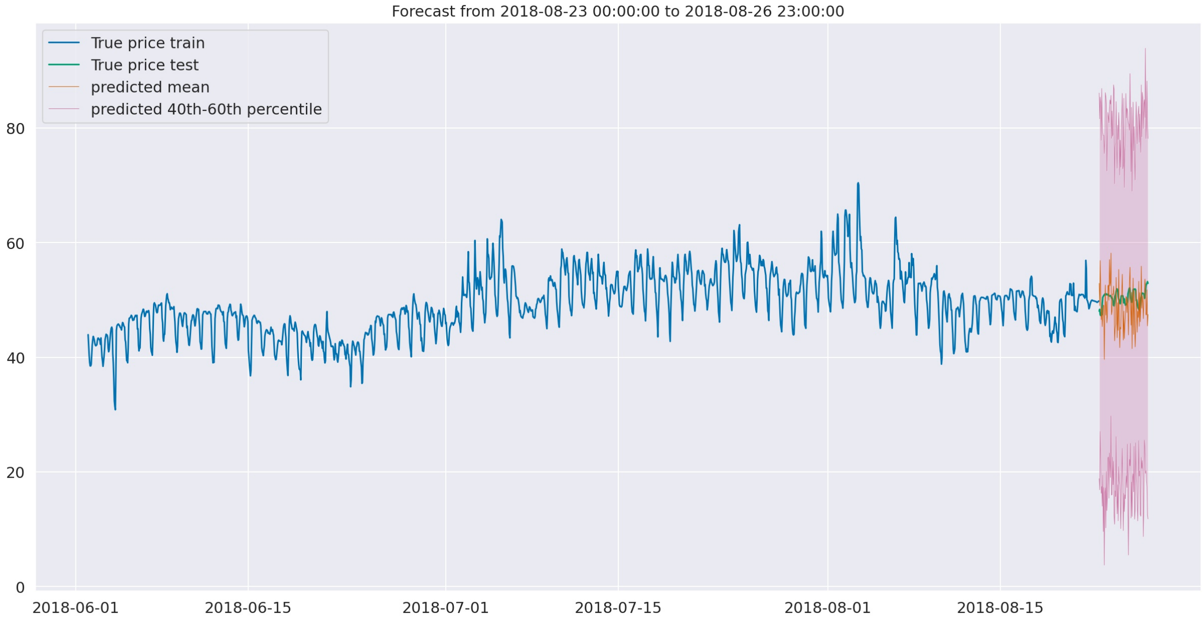


Figure D.2: Probabilistic LSTM forecasts for period 6.

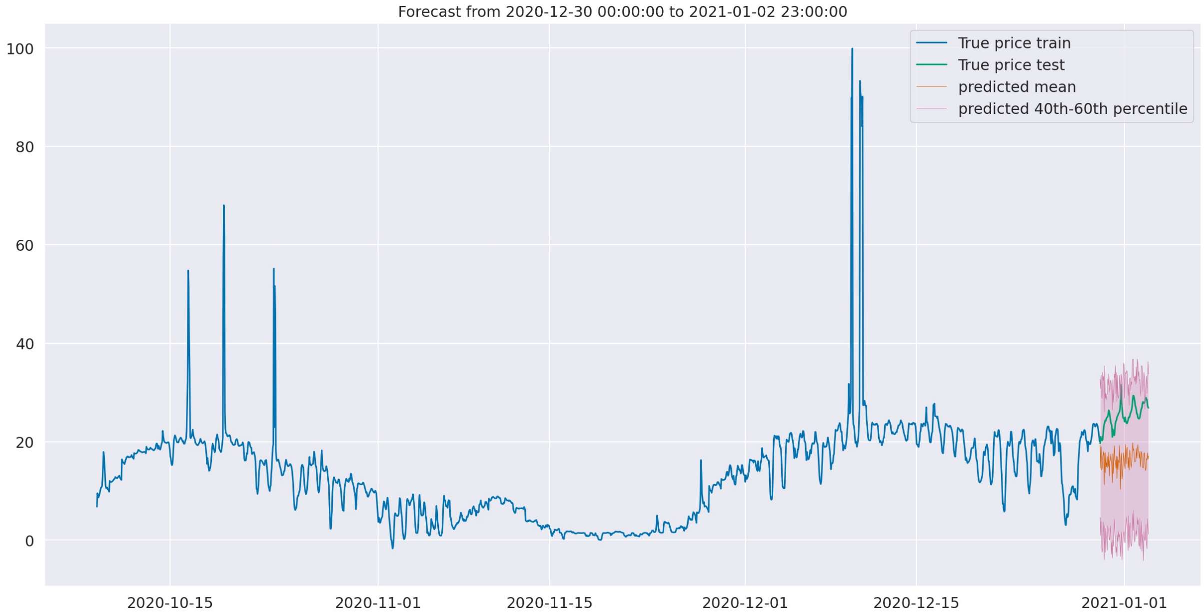


Figure D.3: Probabilistic LSTM forecasts for period 16.

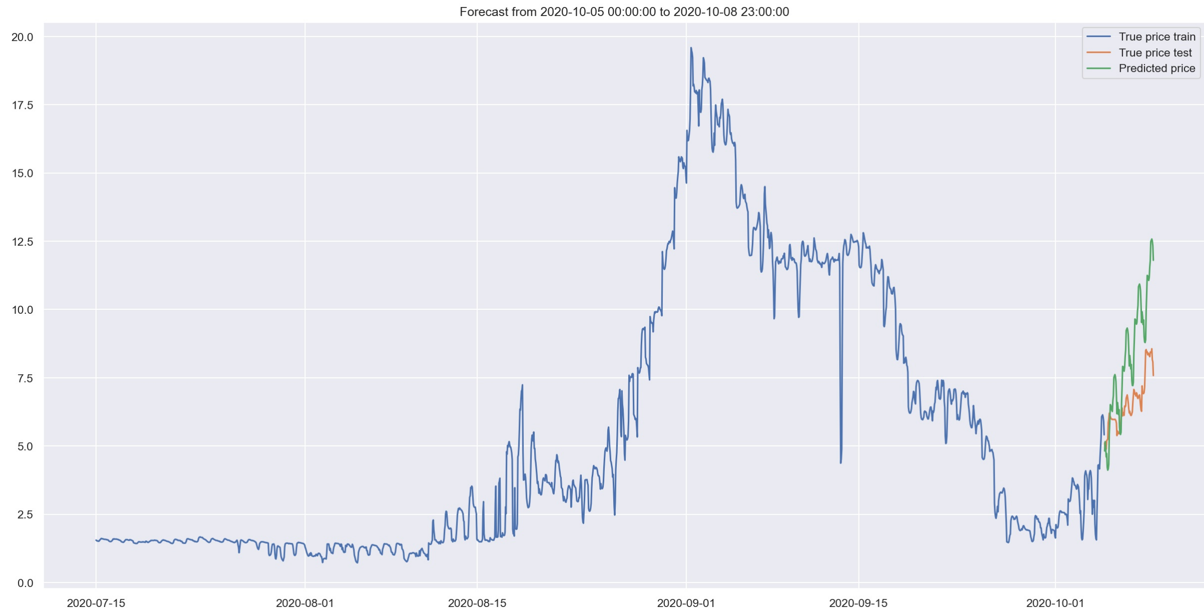


Figure D.4: SARIMA 2 forecasts for period 16.

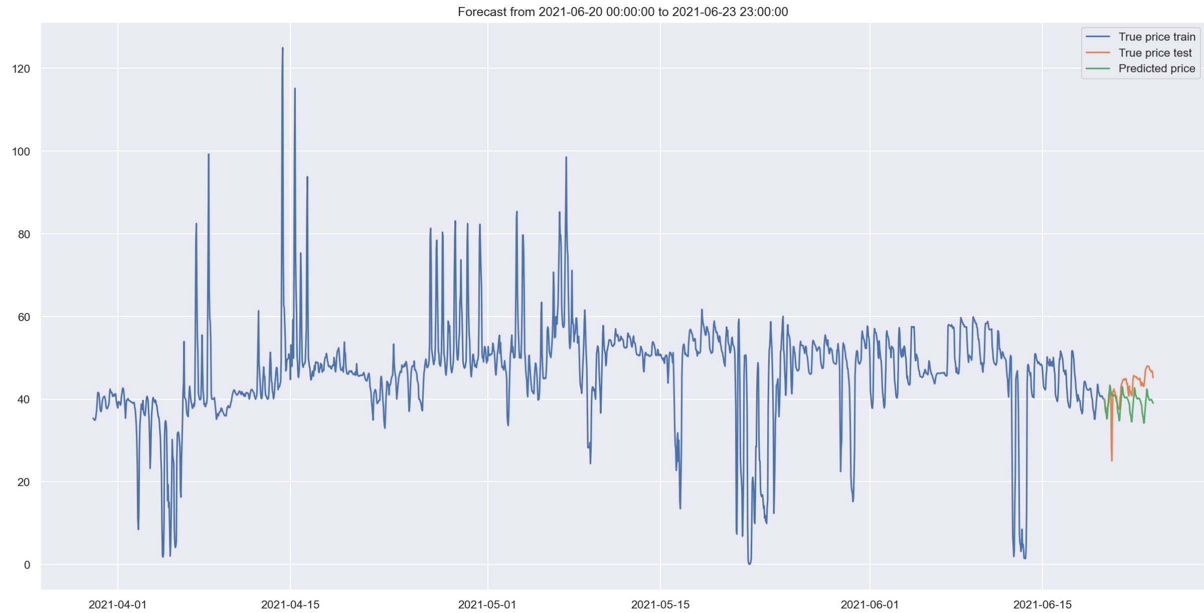


Figure D.5: SARIMA 2 forecasts for period 19.

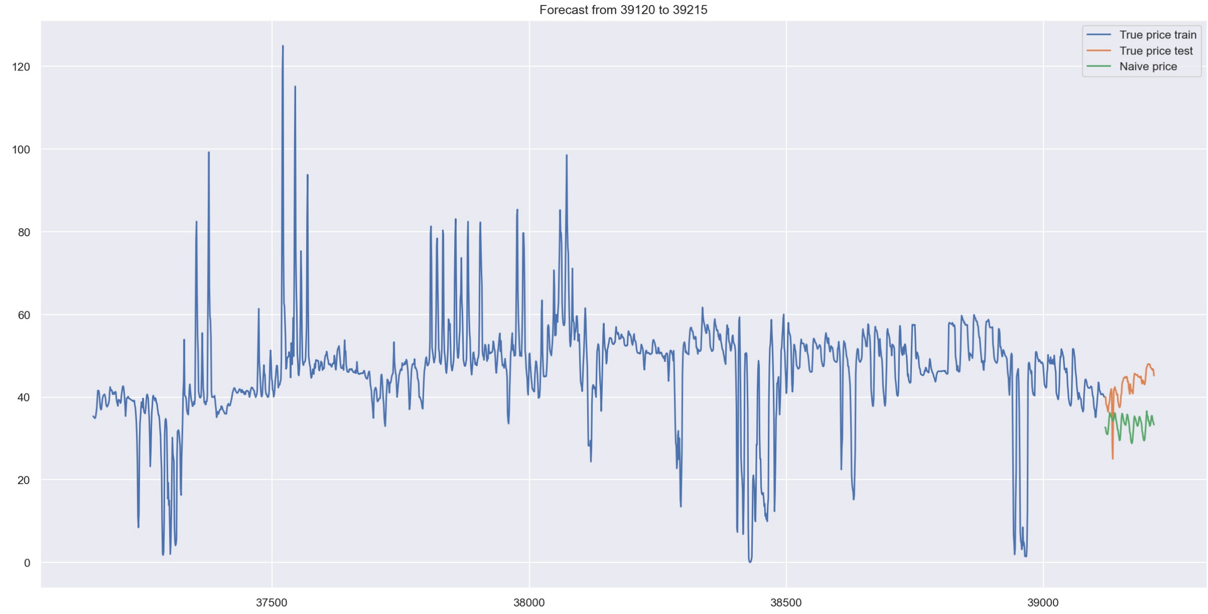


Figure D.6: Historical Average 2 forecasts for period 19.

D.2 Performance for all regions

This section holds together results from the second set of experiments, which measure whether the defined as best models can be sued across all Scandinavian regions as well as how do they perform in more volatile times.

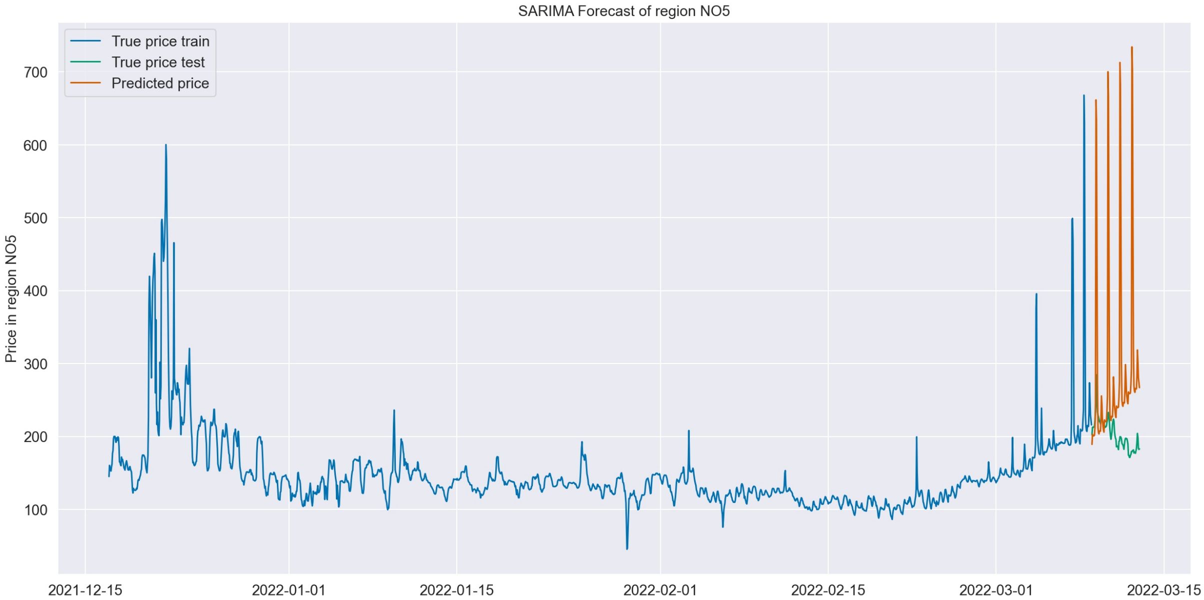


Figure D.7: SARIMA forecasts for NO5.

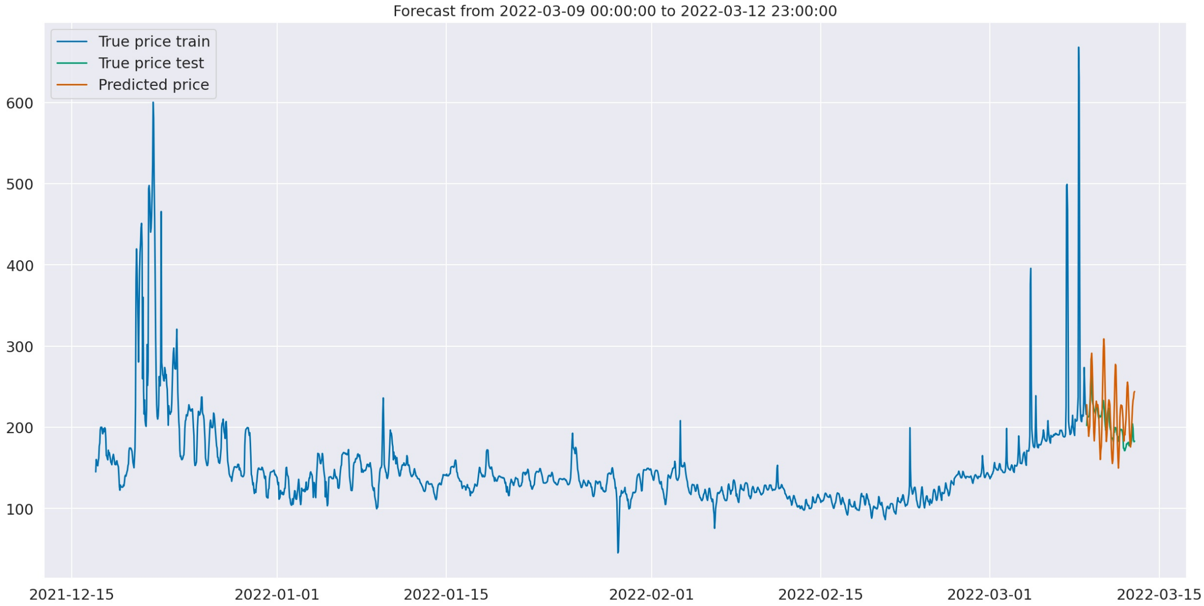


Figure D.8: Probabilistic LSTM forecasts for NO5.

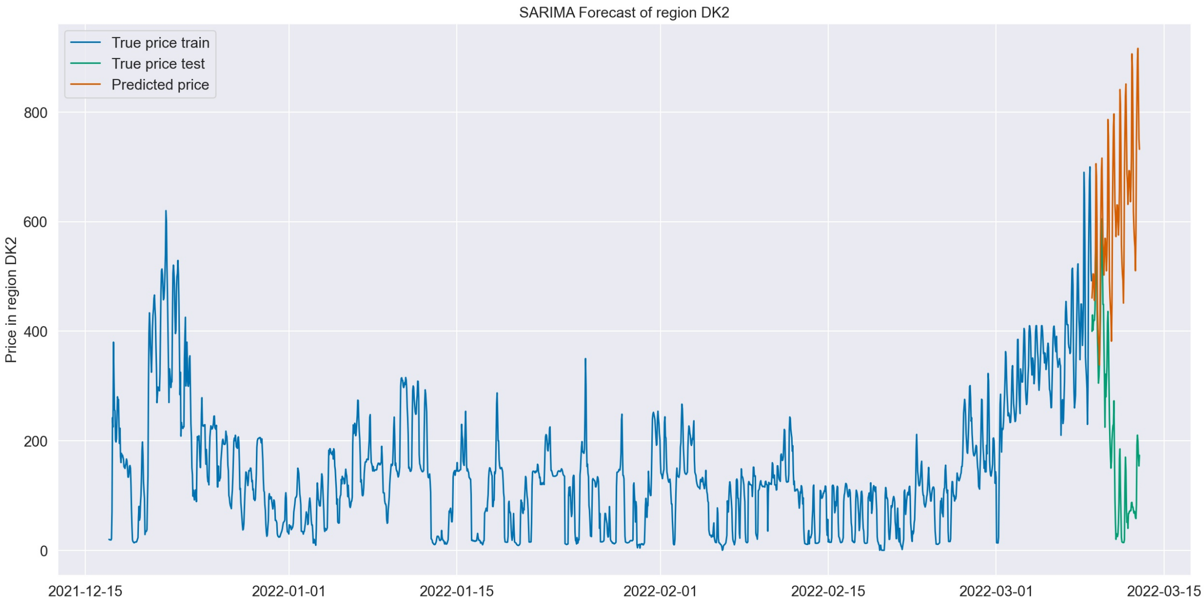


Figure D.9: SARIMA forecasts for DK2.

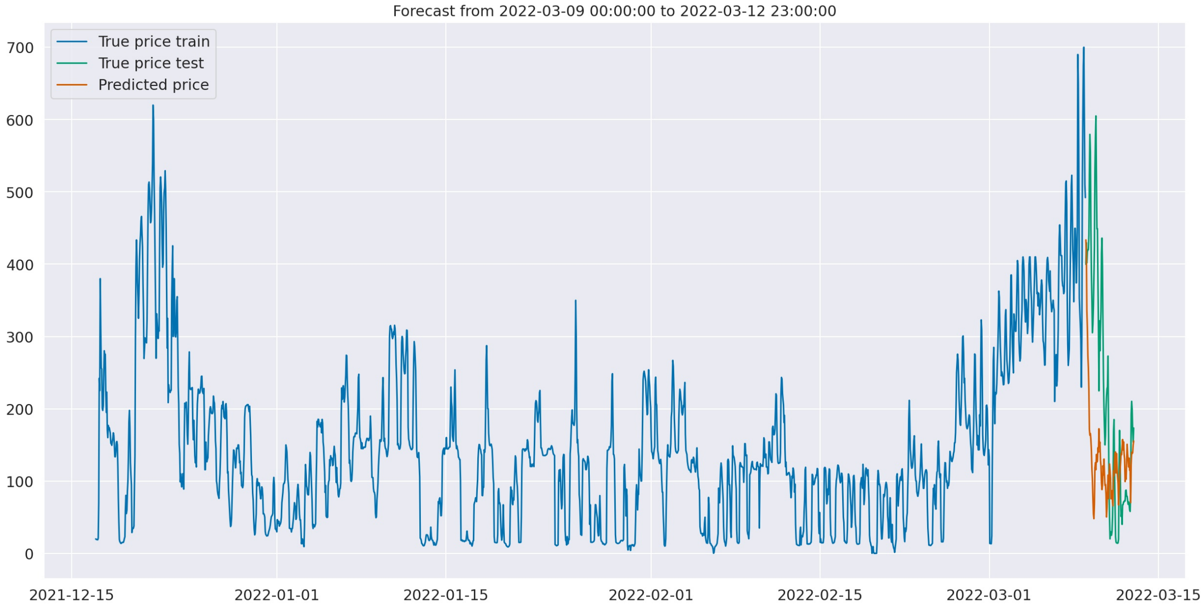


Figure D.10: SARIMA forecasts for DK2.

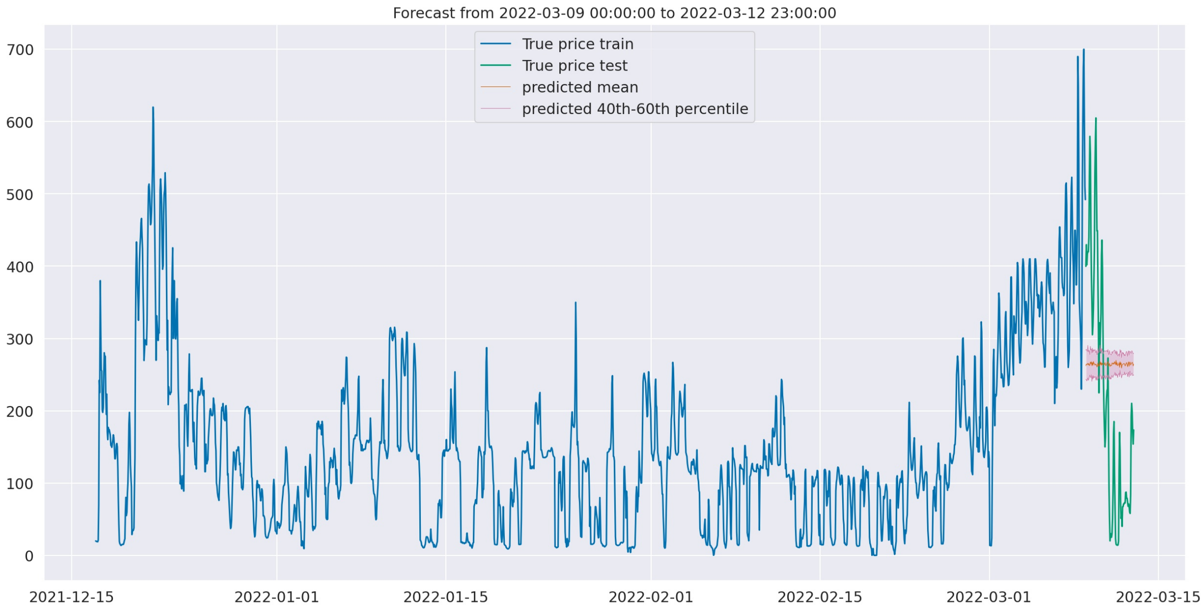


Figure D.11: Probabilistic LSTM forecasts for DK2.

E Learning Objectives

The aim of the research project is to achieve a model with the highest accuracy that forecasts electricity price in a time window of four to seven days in order to have a better planning horizon and better purchasing strategies of electricity.

Monta's interest in knowing the price lies in being able to perform smart charges with the maximum profitability. This is, being able to bill customers for charging in a low price and still make a profit.
The project's resulting model will allow Monta to adapt the price it charges their customers for the Smart Charging service [17] and increase profits.

E.1 Specific learning objectives

In regards to the specific theory and context of the thesis project, the student can:

- Describe and properly apply probabilistic deep learning models;
- Define benchmarks and evaluate the model's performance;
- Design and optimize deep learning experiments in a production environment ;
- Describe the driving forces of the european electricity markets;

E.2 General Learning objectives

In regards to the general educational motivations of the thesis project, the student can:

- Review and assess on technical scientific issues as well as the intrinsic components of the issue;
- Analyse technical problems, with the ability of decomposing into their subcomponents and apply the most fit methodologies to solve the problems;
- Collect relevant scientific knowledge and communicate in an effective and transparent way;
- Reflect on the latest scientific topics within his/her field;
- manage a technical project through all its phases, including scoping, planning, design, experimentation, solution and documentation.

Technical
University of
Denmark

Bygningstorvet 116b
2800 Kgs. Lyngby
Tlf. 4525 1700

www.transport.man.dtu.dk