

EVALUATING THE SUITABILITY OF GAUSSIAN PROCESS REGRESSION AND XGBOOST ON ELECTRICITY PRICE FORECASTING

OWEN LIU

Master's thesis
2021:E57



LUND UNIVERSITY

Faculty of Science
Centre for Mathematical Sciences
Mathematical Statistics

Abstract

Electricity finds itself different from other fresh-ware commodities, it cannot easily be stored. This characteristic trait of electricity results in traditional pricing methods not working for electricity pricing. Thus different pricing schemes are needed, such as Price Forward Curves (PFC) or pricing against a price level. The price forward curves are constructed through a mix of historical market data and model predictions, and the price levels are computed by dividing the price of each hour by the average monthly price to get a ratio, so called Hour-to-month ratio (H2M). This ratio can then be used instead of prices to create predictions. Furthermore, the German electricity sector is changing, with a rapid growth of renewable energy production a better understanding on how future electricity prices and how to model the future price curves is needed.

In this thesis, I will first study how different energy production types work as explanatory variables through linear regression on differentiated data. That knowledge will then be taken and put it to use in Gaussian Process Regression but with H2M ratios instead of prices. Then some exploration on how to include dummy variables in Gaussian Process Regression is done, with the use of different model families to easily compare the result within each model group. Lastly a short evaluation on whether the XGBoost software is a good fit for the problem is done. This study will be done in the German power market and uses data from smard.de, which can be found in chapter 3. It shows that renewables are a good predictor and later on the discussions about the different model structures will be found.

Keywords: Power Markets, Seasonality, Electricity Spot Price, Gaussian Process Regression, XGBoost

Acknowledgements

This thesis was done in cooperation with EnBW, Karlsruhe, Germany. I would like to extend my deepest gratitude towards my supervisor at EnBW: Rikard Green, as well as the rest of the team: Heider Pascal, Etesami Seyyed and Ungerer Sorana. I would like to thank you for giving me a chance to do this thesis as well as the massive learning opportunity I got from this time.

I would also like to thank my university supervisor Prof. Erik Lindström for his mentoring for my thesis and for all knowledge sharing during this time.

Last but not least, I would like to thank all my friends and family who supported me through my studies.

Lund, 2021

Owen Liu

Contents

1	Introduction	4
2	Theory	7
2.1	Machine Learning	7
2.1.1	Supervised Learning	8
2.2	Statistical Theory	10
2.2.1	Bayesian Statistics	10
2.2.2	Linear Regression	11
2.2.3	Stochastic Process	11
2.2.4	Gaussian Process	12
2.3	Gaussian Process Regression	13
2.3.1	The Regression	13
2.3.2	Covariance Kernels	15
2.3.3	Hyperparameters in Gaussian Procecss Regression	18
2.4	Binary Trees	20
2.4.1	XGBoost	21

3	Data	26
4	Empirical Study	34
4.1	Linear Regression Approach	34
4.2	Naive Gaussian Process Regression Approach	41
4.3	Gaussian Process Regression Using Dummy Variables	44
4.3.1	GPR Model Group 1	45
4.3.2	GPR Model Group 2	47
4.3.3	GPR Model Group 3	49
4.3.4	GPR Model Group 4	50
4.4	XGBoost	51
4.5	Sensitivity Testing of the Models	53
5	Discussion	56
6	Conclusion	60
	Bibliography	61
A	Supplementary plots	63
B	Detailed Results	71

Chapter 1

Introduction

Background

Electricity is a very much integral commodity in our daily lives, almost everything we use daily requires it. However, it is very different from most other commodities. While being a fresh-ware, it is still way different from all other fresh-wares as it cannot be stored at all, with the limited exception of storing water at a higher elevation to use at a later time. This leads to certain complications that needs to be fulfilled, where the produced electricity needs to match the demand at all times, as the electricity needs to be consumed at the time of production or it may cause damage to the grid. Couple this with the inelasticity of demand and production, sensitivity to weather and different intensities of business and it leads to a commodity market unlike most others, with complex seasonal patterns, potential huge volatility in prices and price spikes, both in the positive and negative. With the level of volatility present in the market, it comes as no surprise that the actors wish to hedge themselves against future uncertainties in the electricity market.

In the electricity market, the parties can trade either in the day-ahead market on an hour-by-hour basis by placing bids to either sell or buy quantities of electricity at certain hours or the futures market, where long term contracts are signed where a trade further into the future is made. However, these contracts are often not liquid enough to use to gauge and predict the future electricity prices as even the most liquid futures tend to be in front-months or front-years, e.g. the nearest full future month or year, therefore different methods to predict the future prices are necessary, leading to the

implementation of an over-the-counter (OTC) market for non-standard contracts. In the OTC market, Price Forward Curves (PFC) enters the picture. A PFC in the power market is constructed using a mix between market and model, with the goal of capturing the different seasonal and in-season patterns. A forward price curve is the price curve of a commodity, describing the price as seen today for a delivery in the future. [KYOS 2020]

Furthermore, the power market in Germany is seeing a major change. With a massive growth in the renewable energy production sector, from a 63.4 TWh production from renewables in 2005 to 105.2 TWh production in 2010, 188.8 TWh production in 2015 and a predicted continuous growth. The share of which was produced from renewables was around 30% in 2016, and is predicted to reach 40-45% in 2025, thus raising a need to understand how renewables affects the price curve.

This thesis will focus on using models to get an understanding on whether renewables are a good explanatory variable for the power prices. Then it will explore some different model families and use them for predicting future electricity price levels.

Research Questions

The purpose of this thesis is to first create an understanding on how renewables and conventional power production affects the power prices. Following this, the goal is to evaluate whether the statistical models Gaussian Process Regression and XGBoost are good options for predicting electricity prices in Germany. Along with this, a personal goal to learn and understand the aforementioned toolboxes is set.

The objectives for this thesis can be formulated as follows: Are statistical model primarily using renewable sources such as solar power and wind power a viable approach to predict future electricity price levels?

Are the chosen model families, e.g. Gaussian Process Regression and XGBoost suitable model structures for this problem?

Thesis Outline

This thesis is structured as follows. Chapter 2 will briefly introduce how Machine Learning works, as all following theory builds upon it. Then it will mainly describe Gaussian Process Regression and a brief introduction on XGBoost, as Gaussian Process Regression will be this thesis' main focus and XGBoost its secondary focus. Chapter 3 discusses the electricity market along with the data available along with the pre-processing of said data. Chapter 4 describes the models used and the results from said models. Chapter 5 contains the discussion about all the models, the approach to the problem and thoughts about what could have been done to improve the models. Finally the conclusions are presented in chapter 7.

Related Research

In R. Greens paper from 2014, he constructs a PFC with hydropower production dependence. Using a feed-forward artificial neural network trained on the historical hourly spot prices from Nord Pool and weekly measurements he aims to capture the intra-daily and intra-weekly patterns in the price. The seasonal patterns are then estimated using historical futures prices from Nasdaq OMX. Furthermore, three different scenarios were established: normal, wet and dry seasons, and in his paper he researches how these different scenarios change the spread between peak and off-peak prices.

In some other analysis by Beolet, de Jong and Enev, 2014 ([KYOS 2014]) and de Jong, Van Dijken and Enev, 2013 ([KYOS 2013]) they studied how renewables can be used to predict the intra-daily seasonal patterns in power prices. In their analysis, they observed how an increase in solar or wind power production in Germany resulted in a fall in spot prices, where both production types affect the prices at a comparable level. By incorporating this information into the construction of PFCs, they observed an on average better fit, and could observe an increase in prices in early evening and decrease in prices during the middle of the day on average, which is to be expected with the major influx of solar power production in Germany.

Chapter 2

Theory

In this chapter, a brief overview of theory, definitions and notations will be shown to the reader, to familiarize the reader with the writing style in this thesis. It will contain a brief note on Machine Learning as a concept as all following theory build upon it, even if not explicitly mentioned, as well as the theory behind the major model structures used.

2.1 Machine Learning

Machine Learning, a subset of artificial intelligence, are the computer algorithms that improve through experience without human interference. The field can be divided into three broad categories: Supervised Learning, Unsupervised Learning and Reinforcement Learning. Unsupervised learning involves tasking the algorithm to find structure in the input without any prior labels and reinforcement learning involves tasking an agent to perform actions in a dynamic environment and receives feedback on its choices.

This thesis will use algorithms that belong to Supervised Learning, where the computer will receive pairs of inputs and desired outputs and will be tasked to learn a general rule that maps the pairs. A more mathematically rigorous definition, taken from [W1] will be shown below.

2.1.1 Supervised Learning

Given a training set of form $\{(x_1, y_1), \dots, (x_N, y_N)\}$, where x_i is the feature vector of the i -th pair of the training set and y_i is the label or class, a learning algorithm seeks a function $g : \mathbf{X} \rightarrow \mathbf{Y}$, where \mathbf{X} is the input space and \mathbf{Y} is the output space. The function g is a function from the function space \mathbf{G} , usually called the hypothesis space. Sometimes g can be represented using a scoring function $f : \mathbf{X} \times \mathbf{Y} \rightarrow \mathbb{R}^+$ and let g be defined by returning the y value that gives the highest score: $g(x) = \arg \max_y f(x, y)$. The space of scoring functions will be denoted \mathbf{F} .

While \mathbf{G} and \mathbf{F} can be any arbitrary space of functions, one example choice would be in a probabilistic setting, where one can set g as the conditional probability $g(x) = P(x|y)$ and/or f as their joint probability $f(x, y) = P(x, y)$. As for choosing these functions f and g , there are two basic approaches: either Empirical Risk Minimization or Structural Risk Minimization. The first seems to find the function that fits the training data the best while the second includes a penalty function to control the bias/variance tradeoff. Define the Loss function as $L : Y \times Y \rightarrow \mathbb{R}^+$, where the loss for predicting value \hat{y}_i is $L(y_i, \hat{y}_i)$, where \hat{y}_i is generally obtained through some function g . Then regardless of method chosen, under the assumption that all pairs in the training set are independent and identically distributed, the aim is to minimize the expected loss, or the risk, of g , *i.e.* $R(g)$. This risk can be estimated from the training data as

$$R_{emp}(g) = \frac{1}{N} \sum_i L(y_i, g(x_i)). \quad (2.1)$$

One commonly used loss function is the weighted quadratic loss, where the loss would be defined as

$$L(y_i, \hat{y}_i) = C(y_i - \hat{y}_i)^2 \quad (2.2)$$

for some scaling constant C . However, as one can note, the term $(y_i - \hat{y}_i)^2$ is unbounded if $E(y^2) = \infty$, and thus one has to be careful with approaches involving unbounded functions. A statistician might instead the negative log likelihood instead, where the likelihood of observing x given the parameters θ is defined as

$$\mathcal{L}(\theta|x) = f_\theta(x). \quad (2.3)$$

[W1]

Empirical Risk Minimization

In Empirical Risk Minimization one seeks to let the supervised learning algorithm find the function g that minimizes $R(g)$ by some optimization algorithm. If one let g be a conditional probability distribution $P(y|x)$ and the loss function be the negative log likelihood $L(y, \hat{y}) = -\log P(y|x)$ then the empirical risk minimization is equivalent to the maximum likelihood estimation. However, if the training set is too small or if the set \mathbf{G} is too large the empirical risk minimization may lead to high variance and poor generalization, as the learning algorithm may memorize the training examples instead of general rule. This phenomena is called overfitting. However, as mentioned above, the MLE in theory holds by

$$\arg \max_{\theta} \log p_{\theta}(x) = \arg \max_{\theta} \frac{1}{N} \sum_{i=1}^N \log p_{\theta}(x_i) \approx E_{\theta}[\log P(X)] \quad (2.4)$$

and one can look at the Kullback-Leibler Divergence to study how this would converge. Let q_{θ} denote the fitted density, and p_{θ_0} denote the true density, then

$$\int (\log q_{\theta}(x) - \log p_{\theta_0}(x)) p_{\theta_0}(x) dx = \int \log \frac{q_{\theta}(x)}{p_{\theta_0}(x)} p_{\theta_0}(x) dx \geq \log \int q_{\theta}(x) dx = 0 \quad (2.5)$$

where the inequality holds due to Jensens inequality and equality only holds if $q_{\theta} = p_{\theta_0}$ almost everywhere. Thus one can show that this approach holds in theory, but might lead to errors in practice [W1, W2].

Structural Risk Minimization

The more common approach to avoid overfitting and to create models that generalize better is to add a penalizing term to the function we minimize, such as finding a g that minimizes

$$J(g) = R_{\text{emp}}(g) + \lambda C(g) \quad (2.6)$$

where λ is a scaling factor on the bias-variance trade off and C is a penalty function on g . This corresponds to a Bayesian model with R_{emp} being the likelihood and $\lambda C(g)$ the log prior.

One very common way to quantify the goodness of fit for a model is Akaike Information Criterion, or AIC for short. It uses a mix of the loss function as well as penalizing the increased complexity of the model to evaluate which model to choose from a range of different ones.

$$\text{AIC} = 2k - 2\ln(\hat{L}), \quad (2.7)$$

where k denotes the number of estimated parameters and $\hat{L} = L(\hat{\theta})$ is the estimated maximized likelihood value for the model. From this function, one can see that AIC rewards goodness of fit, but penalizes increasing model complexity to make sure unnecessary parameters are removed from the model, thus creating a balance between overfitting and underfitting. There are a range of similar methods, for example Bayesian information criterion (BIC) that follows a similar structure, but penalizes increasing complexity of the model differently [W3]. In practice, these scorings are only used when the models are nested, thus sharing structures, and then one would pick the model with the lowest AIC score.

2.2 Statistical Theory

2.2.1 Bayesian Statistics

For continuous univariate or multivariate variables a and b with joint probability $p(a, b)$, the marginal probability is given by

$$p(a) = \int_{-\infty}^{\infty} p(a, b) db \quad (2.8)$$

and the conditional probability function is given as

$$p(a|b) = \frac{p(a, b)}{p(b)}. \quad (2.9)$$

Using the previous equation twice, one gets Bayes' rule:

$$p(a|b) = \frac{p(b|a)p(a)}{p(b)}. \quad (2.10)$$

If a and b are jointly Gaussian, i.e.

$$\begin{bmatrix} a \\ b \end{bmatrix} = \mathcal{N} \left(\begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \begin{bmatrix} A & C \\ C^T & B \end{bmatrix} \right) \quad (2.11)$$

they have marginal distributions

$$a \sim \mathcal{N}(\mu_a, A) \quad (2.12)$$

$$b \sim \mathcal{N}(\mu_b, B). \quad (2.13)$$

Following this the respective conditional distributions can be obtained as follows

$$a|b \sim \mathcal{N}(\mu_a + AB^{-1}(b - \mu_b), A - CB^{-1}C^T) \quad (2.14)$$

$$b|a \sim \mathcal{N}(\mu_b + BA^{-1}(a - \mu_a), B - CA^{-1}C^T) \quad (2.15)$$

2.2.2 Linear Regression

Given a dataset $\{y_i\}$ with observations and a dataset $x_i = \{x_{i,1}, \dots, x_{i,p}\}, i = 1, \dots, n$ consisting of n features along with the assumption that the relationship between X-vector and y is linear, we can construct a linear regressor of the form

$$y_i = \beta_0 + \beta_1 X_{i,1} + \dots + \beta_p X_{i,p} + \varepsilon_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n, \quad (2.16)$$

where $X^T \boldsymbol{\beta} = g(x)$ from Section 2.1.1. More commonly, these equations are written in matrix notation, given as follows

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad (2.17)$$

where

$$\begin{aligned} \mathbf{y} &= \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \\ \mathbf{X} &= \begin{pmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_n^\top \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}, \\ \boldsymbol{\beta} &= \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}. \end{aligned} \quad (2.18)$$

2.2.3 Stochastic Process

A stochastic process $X = \{X(t); t \in T\}$ is a collection of random variables indexed by some index set T which are all defined a common probability

space.

Furthermore, some well known properties for a stochastic process X will be defined. The mean function is given as

$$m_X(t) = \mathbb{E}[X(t)].$$

From this mean function $m_X(t)$, the variance function $\text{Var}_X(t) = \mathbb{E}[X^2(t) - m_X(t)^2]$, the autocorrelation function $r_X(t, s) = \mathbb{E}[X(t)X(s)]$ and the autocovariance function

$$k_X(t, s) = \mathbb{E}[(X(t) - m_X(t))(X(s) - m_X(s))] = r_X(t, s) - m_X(t)m_X(s)$$

can be defined.

This latter defined function, the autocovariance function, is often referred to as the *covariance function* or *covariance kernel* when working in a machine learning setting. It is symmetric for multivariate stochastic processes, $k_X(t, s) = k_X(s, t)$ and additionally it is positive semi-definite and thus any matrix K , a kernel matrix, with entries $K_{ij} = k_X(t_i, t_j)$ is a positive semidefinite matrix.

2.2.4 Gaussian Process

A continuous time stochastic process $\{X_i; i \in I\}$ is a Gaussian process, or sometimes \mathcal{GP} in shorthand notation, if and only if for every finite set of indices $i = 1, \dots, p$ in I

$$X_{1, \dots, p} = (X_1, \dots, X_p) \tag{2.19}$$

is a multivariate Gaussian random variable.

The same Gaussian property can also be formulated using the characteristic function: $\{X_i; i \in I\}$ is a Gaussian process if and only if for every finite set of indices $i = 1, \dots, n$ in I , there are real valued $\sigma_{l,i}, \mu_l$ with $\sigma_{i,i} > 0$ such that

$$\mathbb{E} \left[\exp \left(j \sum_{i=1}^k s_i X_i \right) \right] = \exp \left(-\frac{1}{2} \sum_{l,i} \sigma_{l,i} s_i s_l + j \sum_l \mu_l s_l \right) \tag{2.20}$$

holds for all $s_1, \dots, s_k \in \mathbb{R}$ and j denotes the imaginary unit such that $j^2 = -1$, $\sigma_{l,i} = \text{cor}(X_l, X_i)$ and $\sigma_{l,l} = \sqrt{\text{var}(X_l)}$.

As a Gaussian Process is uniquely defined by its mean and autocovariance function, it can also be written as

$$X(t) \sim \mathcal{GP}(m_X(t), k_X(t, s)) \quad (2.21)$$

2.3 Gaussian Process Regression

Gaussian processes are one tool used in machine learning, or more specifically, supervised learning, to capture a non-linear relationship between an input set of dimension D and an output set of either labels or targets, by estimating an unknown function $y = f(x)$. The objective is to learn enough about the relationship between the input and output sets to create a suitable prediction of outputs for a new input set. The matrix with input data will be denoted X , with each row denoted $x_i, i = 1, \dots, n$ and similarly vector of outputs will be denoted y with each individual output denoted as $y_i, i = 1, \dots, n$. The training data as a whole will then be denoted as $\mathcal{D} = (X, y) = \{(x_i, y_i); i = 1, \dots, n\}$. And furthermore, new data will be denoted as $\hat{x}_i, \hat{X}, \hat{y}, \hat{y}_i$ and $\hat{\mathcal{D}}$. The goal is then to estimate $\hat{y}|\hat{X}, X, y$, which is estimating how new data (\hat{y} and \hat{X}) behaves based on our current knowledge of their relation (y and X).

2.3.1 The Regression

As stated above, the goal is to model the relation between the inputs and outputs as a Gaussian process $F = f(x); x \in \mathcal{X}$ with a mean function $m(x)$ and covariance function $k(x, x')$

$$f(x) \sim GP(m(x), k(x, x')). \quad (2.22)$$

The known outputs y and unknown outputs \hat{y} will relate to the corresponding inputs X and \hat{X} by a multivariate Gaussian distribution

$$\begin{bmatrix} y \\ \hat{y} \end{bmatrix} \Big|_{X, \hat{X}} \sim \mathcal{N} \left(\begin{bmatrix} m(X) \\ m(\hat{X}) \end{bmatrix}, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, \hat{X}) \\ K(\hat{X}, X) & K(\hat{X}, \hat{X}) \end{bmatrix}, \right) \quad (2.23)$$

where $m(X)$ has elements $[m(x_1) \dots m(x_n)]^T$ and K_{ij} has the elements $k(x_i, x_j)$ and same rules apply for the expressions containing \hat{X} . As stated earlier in section 2.1.2, this K matrix is positive semi-definite and thus forms a valid covariance matrix. To allow for noisy outputs of form $y = f(x) + \varepsilon_n$, a noise term $\sigma_n^2 I$ is added to the first block matrix, with $\sigma_n = 0$ if there is no noise. As only the training outputs are affected by noise, the remaining three blocks have no noise term.

One important distinction to note here is that modelling $f(x)$ as a Gaussian process does *not* assume that $f(x)$ has the shape of a normal distribution. What it does is that we assume that for each given x , the value $f(x)$ is drawn from a normal distribution with mean $m(x)$ and variance and covariances given by the covariance function k . In the same vein, every finite dimensional vector $[y, \hat{y}]^T | X, \hat{X}$ corresponds to a vector drawn from a multivariate Gaussian distribution. Based on the kernel structure, one will get different structures on the output. The assumptions on the output the kernel makes will be discussed in greater detail in a later chapter.

The regression itself will be done by Bayesian inference. Conditioning on y using results from 2.2.1 gives

$$\hat{y} | \hat{X}, X, y \sim \mathcal{N}(\mu_{\hat{y}}, C_{\hat{y}}) \quad (2.24)$$

where

$$\mu_{\hat{y}} = m(\hat{X}) + K(X, \hat{X}) [K(X, X) + \sigma_n^2 I]^{-1} [y - m(X)] \quad (2.25)$$

$$C_{\hat{y}} = K(\hat{X}, \hat{X}) - K(\hat{X}, X) [K(X, X) + \sigma_n^2 I]^{-1} K(X, \hat{X}). \quad (2.26)$$

Using the predicted mean values and covariance matrices for the conditional probability $\hat{y} | \hat{X}, X, y$ one can then obtain predictions for the values of \hat{y} . Define a loss function $L(\hat{y}_{\text{true}}, \hat{y}_{\text{pred}})$, the optimal prediction would be the one that minimizes the expected loss. If the predicted distribution is Gaussian and the loss function is symmetric, which means that overestimating and underestimating are penalized equivalently, then the best prediction is the predicted mean, $\mu_{\hat{y}}$ (or median to be more precise). In this report, the loss functions will be symmetrical and the predictions Gaussian and thus the predicted mean will be used as the prediction.

In a Bayesian sense, the mean function $m(x)$ and covariance function $k(x, x')$ correspond to the prior, $p(a)$ in Bayes' rule, and Equation (2.22) is at times referred to as the \mathcal{GP} prior. The training outputs y then correspond to

the likelihood, and together in equations (2.24), (2.25) and (2.26) corresponds to the posterior. If prior knowledge about the mean is known, it can be incorporated into $m(x)$, else if no prior knowledge exists the prior mean is usually set to zero. Note that setting $m(x) = 0$ does not restrict the posterior mean nor the prediction in any way, as it becomes

$$\mu_{\hat{y}} = K(\hat{X}, X) [K(X, X) + \sigma_n^2 I]^{-1} y \quad (2.27)$$

As one can see from Equation (2.27), the best prediction of $\mu_{\hat{y}}$ for \hat{y} are linear combinations of the training outputs y , thus the Gaussian Process regressor is a linear smoother. One can then view the \mathcal{GP} regressor as an estimator for the underlying function as a sum of radial basis functions, each with a basis function centered at a training input. Furthermore, it is possible to use a non-deterministic prior mean function, see section 2.7 in [GPML].

2.3.2 Covariance Kernels

While the mean function $m(x)$ can be set to zero without restricting the \mathcal{GP} posterior in any sense, the covariance function, the *kernel*, is the key in Gaussian process regression. This kernel contains the specifications of the structure of the Gaussian process. An assumption that two inputs x and x' being close, i.e. Euclidian distance, to each other in the input space means that the two corresponding outputs are close corresponds to a smoothness assumption made on the GP. Similarly, if a prior knowledge about $f(x)$ being periodic is known, this can be incorporated into the covariance kernel [GPML]. As known from previous sections, a valid covariance kernel must be symmetric and positive semi-definite, and thus those requirements are posed on any suggested kernel $k(x, x')$.

The Squared Exponential and Matérn Kernels

A common isotropic covariance kernel is the squared exponential (SE) kernel:

$$k_{\text{SE}}(r) = \exp\left(-\frac{r^2}{2l^2}\right) = \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right). \quad (2.28)$$

The parameter l is the *characteristic length scale* and describes the rate that the covariance between two points is expected to decrease with respect to distance. Points separated by a distance $r < l$ are assumed to be highly correlated. the SE kernel is also called the *RBF kernel* or *Gaussian kernel*. From Equation (2.28) one can clearly see that the SE kernel is infinitely differentiable, and thus the corresponding assumption is made that the Gaussian process has a mean square derivative of all orders, i.e. $\partial^k f(x)/\partial x_i^k$ exists for all i and k , which is a smoothness assumption that can often be too strong for practical models.

A more practical isotropic covariance function is the Matérn class covariance kernel:

$$k_{\text{Matérn}}(r) = \frac{2^{1-\nu}}{\Gamma\nu} \left(\frac{\sqrt{2\nu}r}{l} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu}r}{l} \right) \quad (2.29)$$

where K_ν is a modified Bessel function of the second kind. The Matérn kernel is similar to the SE kernel but has an additional parameter ν , which represents the smoothness of this kernel. The \mathcal{GP} is mean square differentiable up to ν times, i.e. the \mathcal{GP} is mean square differentiable m times if and only if $\nu > m$. If one let $\nu \rightarrow \infty$ the SE kernel is obtained. The most commonly used values for ν are half-integers, as then one receive the nice property that the kernel becomes a product of an exponential and a polynomial. For most application, $\nu = 3/2$ and $\nu = 5/2$ are used, as $\nu = 1/2$ is generally considered too rough and $\nu \geq 7/2$ is too similar to the SE kernel. Thus the two commonly used Matérn kernels are:

$$k_{\text{Matérn}\frac{3}{2}}(r) = \left(1 + \frac{\sqrt{3}r}{l} \right) \exp \left(-\frac{\sqrt{3}r}{l} \right) \quad (2.30)$$

$$k_{\text{Matérn}\frac{5}{2}}(r) = \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2} \right) \exp \left(-\frac{\sqrt{5}r}{l} \right). \quad (2.31)$$

While all covariance kernels presented in this subsection are monotonically decreasing functions of $r = \|x - x'\|$, this is not a necessity for a function to be a valid covariance kernel. They can be based on dot products and polynomials as well, which are often used in support vector machine classification tasks [GPML].

Periodic Kernels

When the data is known to have structures or phenomena that repeats in a predictable pattern, a kernel that captures an assumed periodic behaviour can be used. The periodic kernel derived by David Mackay is given as

$$k_{\text{Per}}(r) = \exp\left(-\frac{2 \sin^2(\pi r/p)}{l^2}\right). \quad (2.32)$$

Similarly to the SE kernel and Matérn kernel, the l parameter describes the rate that the covariance decreases between two points based with respect to their distance from each other. The p parameter in turn describes the length of each period, or the distance until the function repeats itself.

Combining Kernels

While each kernel has their own strengths and weaknesses, and assumptions they make and behaviours they capture, sometimes it is not enough on its own. Thus combining kernels and creating new kernels is necessary. The sum, product and convolutions of all valid covariance kernels will form a new covariance kernel [GPML, KRNL]. In particular, kernels are often multiplied by a *variance* parameter σ_0^2 , that sets the global variance of x . This is valid since $k(x, x) = \sigma_0^2$ is a valid covariance function. An example is given as follows:

$$k_{\text{SE}, \sigma_0^2}(r) = \sigma_0^2 \exp\left(-\frac{r^2}{2l^2}\right) = \sigma_0^2 \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right). \quad (2.33)$$

Another frequent case where a combined kernel is used is when modelling a process with noise. Under the assumption that the noise ε is additive and i.i.d Gaussian with zero mean and variance σ_n^2 , then

$$y = f(x) + \varepsilon \quad (2.34)$$

and can be modelled as

$$\varepsilon \sim \mathcal{GP}(0, \sigma_n^2 \delta(x, x')), \quad (2.35)$$

where δ denotes Dirac's delta function. Thus all variances are σ_n^2 and all covariances are zero. The resulting covariance function $k_{\text{noise}}(x, x') = \sigma_n^2 \delta(x, x')$

is symmetric and positive semi-definite and thus a valid covariance function. As a result of this the process with noise can be modelled using the sum of two covariance kernels

$$k(x, x') = k_{\text{process}}(x, x') + k_{\text{noise}}(x, x'). \quad (2.36)$$

The corresponding covariance matrix for all points X is then

$$K(X, X') = K_{\text{process}}(X, X') + \sigma_n^2 I, \quad (2.37)$$

which returns us to the term used in Equation (2.23). The noise does not have to be equal for every x_i either, different assumptions can be made on the noise by changing the structure on $\sigma_n^2 I$. One example would be that an assumption that if x and x' are close together in input space, then εx and $\varepsilon x'$ are potentially lightly correlated. Then one could add a second Matérn kernel with a small lengthscale l_n and global variance σ_n^2 . Similarly, specifying prior knowledge into the covariance kernel can be done by using a Matérn kernel for long term trends, a periodic kernel for season variations etc.

Another case that will be used frequently throughout the thesis is when combining two kernels by multiplication, which in loose terms would form an AND gate in computer terms, a new kernel that would return high values if and only if both the component kernels returns high values. An example combination would be

$$k_{\text{Prod}}(r) = k_{\text{SE}}(r)k_{\text{Per}}(r), \quad (2.38)$$

which combines properties of a distance measuring kernel and periodic repeats. Another typically used product kernel is the combination of two linear kernels

$$k_{\text{Prod}}(r) = k_{\text{Lin } 1}(r)k_{\text{Lin } 2}(r), \quad (2.39)$$

which yields a function that is quadratic. Thus this result can be used to do Bayesian polynomial regression of any degree [KRNL].

2.3.3 Hyperparameters in Gaussian Process Regression

Gaussian process regression requires a lot of model selection decisions, ranging from prior mean and covariance functions to setting the hyperpa-

rameters for the kernels. The hyperparameters in question depends on the choice of kernels, for example for the Matérn kernel the hyperparameters $\theta = \{l, \sigma_0, \sigma_n\}$, the length scale l , the global standard deviation σ_0 and the noise standard deviation σ_n . One way for estimating these hyperparameters is Bayesian model selection by using the maximum likelihood (ML) estimate, in other words, maximizing the probability of the obtained outputs given the inputs and parameters for the model. This is often done in stages, where each stage - model parameters, hyperparameters and structure, are often estimated separately. These stages can also be performed manually, for example choosing the \mathcal{GP} prior and/or which kernels to use.

Bayesian optimization with a \mathcal{GP} model is a non-parametric method, and thus the model parameters are not well-defined and can have multiple interpretations. One of these interpretations is to let the function values be $\mathbf{f} = f(x)$, and by combining equations (2.8) and (2.9) the marginal likelihood $p(y|X)$ becomes

$$p(y|X) = \int_{-\infty}^{\infty} p(y|\mathbf{f}, X)p(\mathbf{f}|X)d\mathbf{f}. \quad (2.40)$$

Using the prior from section 2.3.1 and setting the mean $m(x)$ to zero as it does not restrict the predictions, the log marginal likelihood becomes

$$\log p(y|X, \theta) = -\frac{1}{2}y^T [K_X + \sigma_n^2 I]^{-1} y - \frac{1}{2} \log |K_X + \sigma_n^2| - \frac{n}{2} \log(2\pi), \quad (2.41)$$

where $|\star|$ denotes the matrix determinant and $K_X = K(X, X)$ due to space restraint. Further details can be found in sections 2.2 and 5.4 in [GPML]. The first term in Equation (2.41) can be seen as the general modelling of the data, how well the model fits the data, the second term as a penalty term for increasing complexity of the model as discussed in section 2.1.1 and the last term as a normalization constant. Thus this maximum likelihood estimate of the parameters and hyperparameters attempts to balance the model fit against the complexity, also known as the bias-variance trade-off in machine learning and thus reducing overfitting [GPML].

The log marginal likelihood can then be maximized using standard methods, one example would be taking the partial derivatives with respect

to each hyperparameter

$$\frac{\partial}{\partial \theta_j} \log p(y|X, \theta) = \frac{1}{2} y^T K_X^{-1} \frac{\partial K_X}{\partial \theta_k} K_X^{-1} y - \frac{1}{2} \text{tr} \left(K_X^{-1} \frac{\partial K_X}{\partial \theta_j} \right), \quad (2.42)$$

where $K_X = K(X, X)$ once again and setting this expression to zero. See [GPML] for further details. However a few problems arise here, finding the maximum likelihood is often not a simple task and furthermore, the (log) marginal likelihood often has multiple local minima, which each correspond to different interpretations of the data. If the outputs vary very rapidly they can be explained by an underlying function with a very small lengthscale, or by some long-term trend with noise. The more hyperparameters incorporated into the model the more complex it becomes, which allows for an increased number of ways to interpret the results, and thus more data is needed to accurately estimate the hyperparameter values.

2.4 Binary Trees

As Binary Trees, or rather the algorithm XGBoost is used as a comparative method to the Gaussian process regression, an introduction to the theory will be given here. However, as this is not the main focus of the thesis but merely used as a comparison point, this section will be kept very brief.

In computer science, a binary tree is a data structure based on nodes, where each node at most has two children, usually referred to as the left and the right child. This data structure can be used in two different ways:

First, as a means of a searching and sorting algorithm, where each node contains some value or label. If labelled this way, one can implement binary search trees or binary heaps. In these trees, the designation of nodes as the left and right node matters in some of the applications, even if there is only one child node. However, for example in a regular binary search tree, the placement of nodes depends almost entirely on the order of addition, and thus can be re-arranged to balance the tree without changing the meaning.

Second, the implementation used here, is to use the binary tree as a representation of data with a relevant bifurcating structure. In this case, the ordering of the nodes matters majorly, the arrangements of nodes to the left, right over or under is part of the information describing the data.

2.4.1 XGBoost

XGBoost, or Extreme Gradient Boosting, is a statistical algorithm that makes use of an ensemble of binary trees to predict an outcome. The term Gradient Boosting comes from the paper *Greedy Function Approximation: A Gradient Boosting Machine*, by Friedman, which will not be discussed further here [GBM].

The tree ensemble consists of Classification And Regression Trees, CART for short. An example tree could look like this (fig. 2.1):

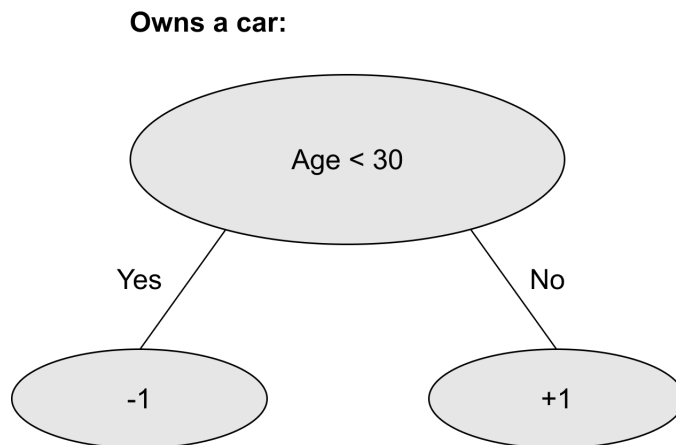


Figure 2.1: Tree model taking age as input.

The tree takes age as an input to evaluate whether someone owns a car by separating the input into two categories and giving a prediction score based on group - a label. For continuous inputs one could use strict inequalities instead of the regular inequality shown in the figure. However, a single is usually not enough to be used in practice, thus leading to an ensemble of trees being used. Each using different inputs and label scores and yielding a prediction together. An example can be found below in figure 2.2

Thus a prediction for someone who is a male and under 30 would yield $f = 1.3 - 1 = 0.3$, the sum of the score from each individual tree. These two trees would complement each other and thus give better prediction than one tree alone. A tree does not necessarily be limited to only having 1 child either, and each branch can have their own children, which creates depth and improves the explanatory ability of a tree. An example of a deeper tree can be seen in Figure 2.3.

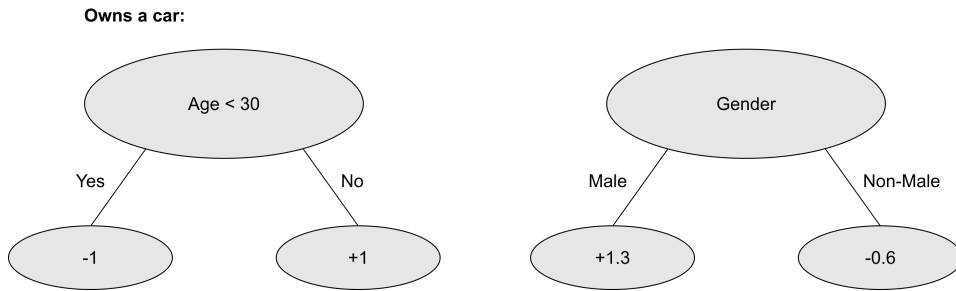


Figure 2.2: An ensemble of trees using age and gender as input.

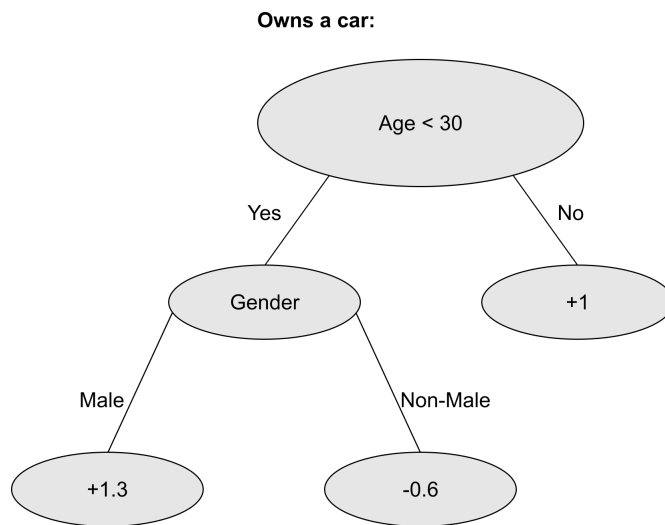


Figure 2.3: A tree where the left child has its own childs.

Mathematically, K number of trees in a model can be written as:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F} \quad (2.43)$$

where f is a tree function in the functional space \mathcal{F} , the set of all possible CARTs. The objective function is then optimized by

$$\text{obj}(\theta) = \sum_i^n l(y_i, \hat{y}_i) + \lambda \sum_{k=1}^K \Omega(f_k) \quad (2.44)$$

where λ is a scaling term punishing increasing complexity and $\sum_{k=1}^K \Omega(f_k)$ describes the complexity of the model [XGBModel].

Training the boosted trees

With the model structure constructed, the next question is natural: how to train the model? It turns out to be a harder problem than traditional optimization problems where one can simply take the gradient, since each function f_i is determined by both the structure of the tree and the leaf scores. *It is simply not feasible to learn all the trees at once, thus instead one use an additive strategy: fix what is learned and add a new tree on top of it.* Let $\hat{y}_i^{(t)}$ be the prediction value at step t , then the prediction value is given as follows:

$$\hat{y}_i^{(0)} = 0, \quad (2.45)$$

$$\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^0 + f_1(x_i), \quad (2.46)$$

$$\hat{y}_i^{(2)} = \sum_{k=1}^2 f_k(x_i) = \hat{y}_i^{(1)} + f_2(x_i), \quad (2.47)$$

$$\dots \quad (2.48)$$

$$\hat{y}_i^{(t)} = \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i). \quad (2.49)$$

With these predictions in hand, next step is to optimize the objective.

$$\text{obj}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^t \Omega(f_i) \quad (2.50)$$

$$= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + \text{constant}. \quad (2.51)$$

Using mean squared error (MSE) as loss function, the objective becomes

$$\text{obj}^{(t)} = \sum_{i=1}^n \left(y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)) \right)^2 + \sum_{i=1}^t \Omega(f_i) \quad (2.52)$$

$$= \sum_{i=1}^n \left[2(\hat{y}_i^{(t-1)} - y_i) f_t(x_i) + f_t(x_i)^2 \right] + \Omega(f_t) + \text{constant}, \quad (2.53)$$

where the first order term is usually called the residual and a quadratic term. However, for different loss functions one may not get such a nice form. Thus in the general case with a custom loss function, one takes the Taylor expansion of the loss function up to the second order:

$$\text{obj}^{(t)} = \sum_{i=1}^n \left[l(y, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + \text{constant}, \quad (2.54)$$

where g_i and h_i are defined as

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}), \quad (2.55)$$

$$h_i = \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)}). \quad (2.56)$$

Removing the constants, the objective at step t simplifies to

$$\text{obj}^{(t)} = \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t). \quad (2.57)$$

Thus the optimization goal only depends on g_i and h_i and any loss function which is twice differentiable can be used [XGBModel].

Model Complexity

The next step to take care of is the regularization term, and to do that the complexity of the tree, $\Omega(f_t)$, needs to be properly defined. First some refinement of the definition of the tree $f(x)$ is needed. Let $f(x)$ be defined as

$$f_t(x) = w_{q(x)}, \quad w \in R^T, \quad q : R^d \rightarrow \{1, 2, \dots, T\}, \quad (2.58)$$

where w is the vector of scores on the leaves, q is a function assigning data-points to their corresponding leaf and T is the number of leaves. Using this notation, the complexity can then be defined as

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2. \quad (2.59)$$

While this is just one of many ways of defining the complexity of a tree, it is chosen since it works well in practice [XGBModel].

The Structure Score

By incorporating all information from Equation (2.57) and (2.59) the objective value for the t -th tree can be written as

$$\text{obj}^{(t)} \approx \sum_{i=1}^n \left[g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2 \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (2.60)$$

$$= \sum_{j=1}^T \left[G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2 \right] + \gamma T \quad (2.61)$$

where $G_j = \sum_{i \in I_j} g_i$, $H_j = \sum_{i \in I_j} h_i$ where $I_j = \{i | q(x_i) = j\}$ which is the set of indices of datapoints assigned to the j -th leaf and all w_j are independent with respect to each other. In Equation (2.61) one can note that the form $[\star]$ is quadratic and thus the best w_j for a given structure $q(x)$ and best objective reduction one can get is

$$w_j^* = -\frac{G_j}{H_j + \lambda}, \quad (2.62)$$

$$\text{obj}^* = -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \gamma T, \quad (2.63)$$

where the last equation measures *how good* a tree structure $q(x)$ is [XGBModel].

Choosing a Tree Structure

By having a measure of how good a tree is, ideally one would wish to enumerate over all possible trees and choose the best tree. However, in practice this is impossible, and thus the optimization has to be done one level of a tree at a time. By splitting a leaf into two, the gain can be described as follows

$$\text{Gain} = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma, \quad (2.64)$$

where the terms in order from left to right means: 1) the score on the new left leaf, 2) the score on the new right leaf. 3) the score on the original leaf and 4) regularization on the additional leaf. The result of this equation is that if the gain of splitting a leaf is less than γ , the tree fares better by not adding a branch. By repeatedly applying this technique, one can find the optimal tree structure for each tree and in a whole for the entire random forest [XGBModel].

Chapter 3

Data

The Electricity Market

Before going through the data, an introduction to the energy market is in order. The energy market is largely split up into two parts, the physical market and the financial market. But before introducing the different markets, the commodity electricity should be properly introduced. Electricity is a fresh-ware, but is unlike other fresh-ware commodities very difficult to store, and thus needs to be consumed at the same time it is produced. While batteries and other power storage solutions such as hydro-pumped power storage exists, they are neither developed enough for large scale usage, or simply not profitable enough to employ. Thus this leads to several phenomena that are rather unique to this market: first one being a constantly moving and fluctuating price over the day, as high demand but low supply leads to higher prices while the converse leads to lower prices. Secondly, the latter in its extreme can lead to negative prices, where shutting down the plants becomes more costly than paying the consumer to use the electricity. [SMARD]

Physical Market

The physical market is where the electricity is produced, distributed and consumed. The production is separated into power plants of different priority levels, where the lower per-unit production plants has priority. A typical low marginal cost is photovoltaics (solar panels), hydro power and wind turbines, which have major upfront cost in construction and then rather low operational cost. The plants mentioned here are often referred to as the renewables. A higher marginal cost power plant would be plants such as brown coal or fossil gas power plants, that have a fair operational cost due to the fuel consumption and maintenance with personnel. These plants are often called the conventionals. However, both types of plants are important for the system, as while the solar panels and wind turbines produce very cheap electricity, the level of control over their production is low and relies on weather, while the conventional fuel burning plants can run regardless of natural conditions. When deciding what powerplants that should be running, one starts with the cheapest plants, and add more and more plants with higher marginal costs until the electricity demand is met. However, some plants are except from this general rule, as some plants are combined electricity and heat production plants, and therefore they might need to run regardless of cost. Thus, other than the economical consequence of increasing the price, has no effect on quality as all produced electricity is equal regardless of origin. Furthermore, this is emphasized by the entire electricity grid in Europe being connected by cross border powerlines.

Another important part of the physical energy market is the balancing act, as stated before, the production needs to match the consumption. And while the conventional fuel burning power plants can readjust their production on demand, the change does not happen instantaneously. Thus the need of balancing services comes into picture. In Germany, every producer and consumer is included in a balancing group, and this group works as a book-keeper, keeping a record of how much influx and efflux of electricity each participant feeds or takes from the grid. Furthermore, they ensure that the difference between the influx and efflux is zero. However, regardless of planning, it may happen that the actual efflux does not match the influx, and then the operators needs to correct the supply and demand by employing certain balancing services. The first category is called Frequency Containment Reserve, which are balancing services that needs to be fully active within 30 seconds, and will last for up to 5 minutes, which is when the second category, the Automatic Frequency Restoration Reserve, kicks in. This group of ser-

VICES will run until the last group activates at 15 minutes, or sometimes even beyond that. The last balancing service, the Manual Frequency Restoration Reserve, will be fully active by 15 minutes and last until resumption of normal operation, which normally happens within an hour. The costs then associated with running these balance services are billed to the balancing group, which act like an economic penalty if a group does not stick to their schedule, raising an incentive to stick to the devised schedule. [SMARD]

Financial Market

On the financial side of the electricity market, the power is bid for and sold, which can be done in a few different ways. In the exchange, one can trade in either the day-ahead market or intraday market. In the day-ahead market, each hourly price the coming 24 hours is bid for at noon the day before. Parties need to submit their bids and offers detailing price and amount by that time. The exchange then sets the wholesale price for each hour of the next day based on the bids and accepts winning bids. Furthermore, this wholesale price works like a reference value for the electricity market, similar to a closing price for a stock market. In the intraday market, electricity can be traded until 30 minutes before delivery, with this timer becoming 5 min within each individual control area. Parties can also directly decide on price and amount between themselves, in over-the-counter trading. This includes trading in form of both long-term contracts, comparable to futures, or short-term contracts, similar to the regular exchange trading.

Another part of the financial market would be the futures market, where long term contracts are traded. Here the electricity can be traded several years in advance, and buyers use these contracts to hedge themselves against rising prices, while the seller secures funds for new infrastructure. For the buyer, this comes at an extra premium, which the seller then registers as additional revenue. [SMARD]

Price Forward Curves

A Price Forward Curve (PFC) is essentially a mix between market and model, and not necessarily evenly split between the two parts depending on available market data. This then becomes more than just the price at which

future delivery should be set at. Unlike a regular futures contract, which have a set delivery date, a price forward curve details the prices at a much higher resolution, as it sets today's price of a commodity that will be delivered at a future date. Whether the curve has been generated with more market prices or model as basis, two main structural patterns can be found: seasonal patterns, such as monthly and quarterly patterns or in-season patterns, such as daily fluctuations or hourly patterns. Both of these price pattern types are determined by the dynamics of demand and production. A country such as Norway where majority of the power production comes from hydro power, the curve will look rather flat, while Germany that relies heavily on solar and wind power, requires support from more traditional power sources such as coal to make up for the variability of production, and thus sees bumpier price curve.

The data

The data¹ used for this thesis was gathered from Smard.de, which offers all kinds of data for the electricity market in Germany and the neighbouring countries. The data gathered includes day-ahead prices and import and export data for Germany and its neighbouring countries as well as the predicted and actual consumption and production of electricity, along with the maximum production capacity, for the time period 2015-01-01 to 2020-09-23. This production and consumption data is given at quarterly resolution and all other data is given at hourly resolution, with each dataset for each power plant type being as 3 separate parts, with each spanning the years 2015-2016, 2017-2018 and 2019-2020 respectively. The only exception to this would be the maximum production capacity, which is given on a yearly resolution.

The datasets for electricity consumption is split into 2 categories, one detailing the forecasted electricity demand, and one detailing the actual demand. The forecasted demand is predicted by the Transmission System Operators based on the day-ahead trading and other empirical data and saved into the forecasted consumption dataset. The actual demand for each given hour in each day is then recorded afterwards and saved into the actual consumption dataset. The production datasets are defined differently, as they are split into various production methods, and all of those production methods

¹available at <https://drive.google.com/drive/folders/1W2T8bub34xvqXkFiqxHaquItGcBsQKwX?usp=sharing>

can be found in Table 3.1 below, where "others" means the merged unnamed minor production types.

Renewables	Conventionals
Hydropower	Fossil Brown Coal
Hydropumped Storage	Fossil Hard Coal
Photovoltaics	Fossil Gas
Wind on/off shore	Nuclear
Biomass	-
Others	Others

Table 3.1: List of datasets of productiontypes

These production types are then split into two categories for each dataset, forecasted production and actual production. The forecasted production is created in two steps, first by combining the information from the forecasted demand and using weather prognosis to predict the renewable influx from solar and wind sources one can compute the forecasted residual load - the amount of power to be generated by non-renewable sources. Then based on the amount of residual load the different marginal costs for electricity productions sets the amount produced for each type of powerplant. This forecasted production is logged into the forecasted generation dataset. The actual production for each hour is then logged into the actual generation dataset. [SMARD Cons, SMARD Prod]

In a similar vein, the cross-border flows can be forecasted, by using market data for all neighbouring pricing regions and other regions where intercountry powerlines exists, and works similarly as predicting conventional productions by using the price in the neighbouring region as marginal cost. This information is then stored in the scheduled commercial exchanges dataset. The actual flow is then later stored in the cross border physical flow dataset. The day ahead prices data is the recorded prices from the day-ahead auction detailed earlier and the installed generation capacity is the theoretical maximum generation capacity for each powerplant type, updated yearly. [SMARD CBF]

A typical graph of the production can be seen below in Figure 3.1.

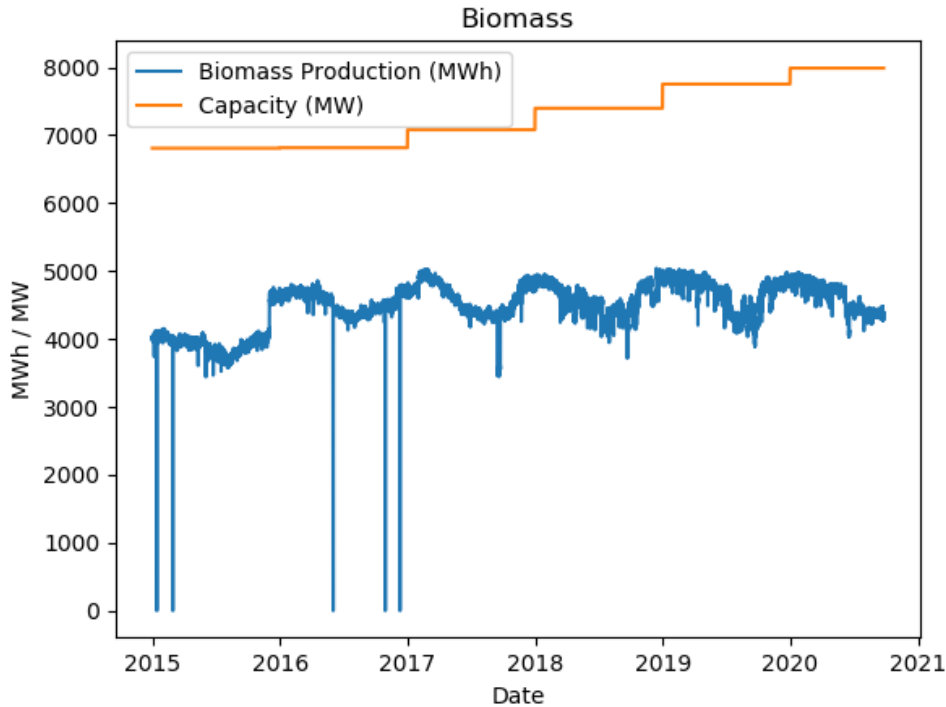


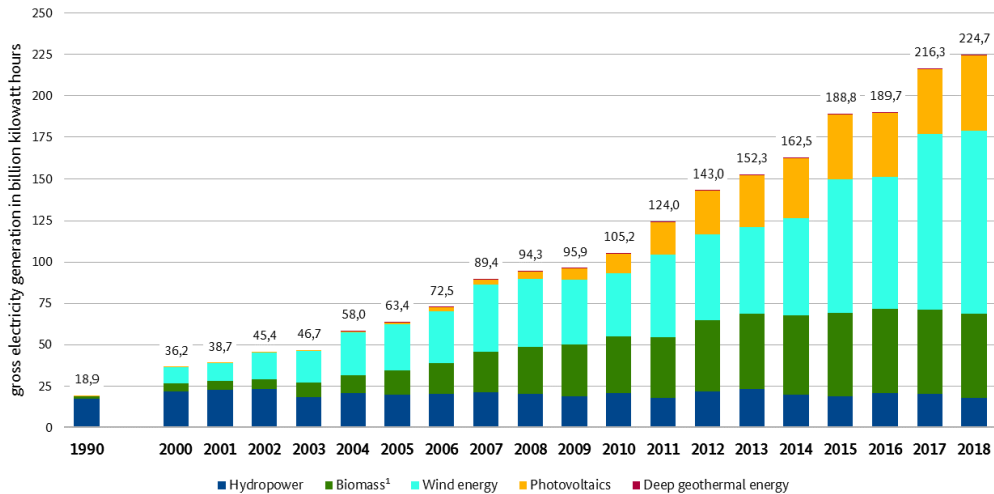
Figure 3.1: Plot of Biomass electricity generation in MWh (blue) and maximum production capacity in MW (orange).

Preprocessing of data

The first thing to note when looking at the data is that there exists a comma as thousand separator in the dataset, which in combination with Excel's parsing caused certain errors, as Excel at times interpreted the comma signs as decimal separators, which led to problems with values changing due to disappearing zeros. These were manually fixed in the dataset as they were found. Furthermore the comma was also removed to parse it properly into Python.

Once parsed into Python, the 3 dataset parts were combined into one, and all data in quarterly resolution is converted into hourly resolution. Furthermore, all the hours which were in summertime are shifted backwards one hour to be presented in winter time. Then, for each row of data, the associated hour, day, month, year, grouped daytype and datetime were added,

Development of gross electricity production from renewable energy sources in Germany



¹ incl. solid, liquid and gaseous biomass, sewage sludge and the biologic fraction of waste (in waste incineration plants estimated at 50 %, from 2008 only municipal waste)
 BMWi based on Working Group on Renewable Energy-Statistics (AGEE-Stat); as of August 2019

Figure 3.2: Energy production from renewable sources, Germany. Source: <https://www.smard.de/page/en/wiki-article/5884/6124>

where grouped daytype is the weekdays with the exception of Tuesday to Thursday sharing class, as their daily price curves tends to be similar. All of this information is then stored in Pandas Dataframe format. [Pandas]

A notable change during the time period 2015-2020 is that the electricity pricing region that Germany is part of got changed to no longer include Austria. This lead to certain production methods, most notably Hydropower and Hydro storage power no longer being a part of the Germany production region, as they were primarily located in Austria. This lead to a notable effect on electricity prices. Another notable trend is Germany’s growing share of renewable energy production, which in 2016 grew to 30% of all produced energy, and projected to reach 45% in 2025. This can be seen in Figure 3.2 below.

To get a full day-ahead price history of the region Germany is part of, both the time series with and without Austria are combined into one series to cover the entire time period. This is done since the majorly affected production area is hydropower production, which belongs to the residual load and thus should not affect the input output relationship as heavily as if the wind power production changed this drastically.

Another preprocessing that has been done was to compute the normalized production - a ratio between the produced electricity and the total production capacity. But when computing this ratio, one problem was found: sometimes the ratio would exceed 1. This can be explained by the construction of new powerplants throughout the year, while the ratio itself would only be updated at the beginning of each year. Thus to compute these ratios, some processing on the capacity data is necessary. Under the assumption that the provided yearly data are correct points, the capacity for each hour is set to be

$$\text{Capacity}_{\text{hour } i} = \max(\text{Capacity}_{\text{hour } i-1}, \max_{j \leq i} \text{Production}_{\text{hour } j})$$

through a loop over all hours i between two yearly update points, and this loop is initialized with $\text{Capacity}_{\text{hour } i}$ from the original capacity data. The capacity is simply set to either the known capacity through our dataset or the highest production level we have seen between two update points. While this extrapolation of the capacity might have its issues, which will be discussed in a later section, it should prove sufficient for needs. A realization of this process can be seen in Figure A.8 in the appendix. Following this, a normalized production, a ratio between the production and the maximum capacity can be computed.

Chapter 4

Empirical Study

To begin this thesis, a recreation of a study that KYOS Energy Consulting BV [KYOS 2013, KYOS 2014] conducted will be done, to learn about how the German electricity price market operates today. In their research they showed that the increase of renewable energy sources has negatively affected (lowered) the electricity prices, most notably during the midday period, where before one would find the peak hours while today the peak hours has moved to the mornings and evenings.

4.1 Linear Regression Approach

Using similar methods as KYOS has done in their papers, a simple understand on how the data behaves can be gained by Linear Regression. The electricity prices are clearly not a stationary time-series, as seen in figure 4.1a, and as such one cannot directly apply linear regression using it. However, the change, Thus the new time series \tilde{y} is constructed as follows

$$\tilde{y}_i = y_{i+1} - y_i, \quad i = 2, \dots, n, \quad (4.1)$$

which as seen in Figure 4.1b should be linear enough to do linear regression on, although possibly heteroscedastic.

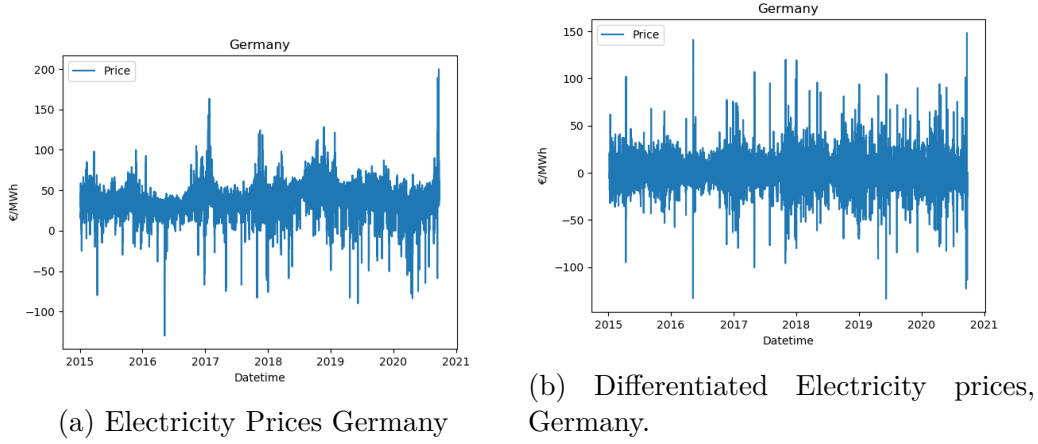


Figure 4.1: Electricity prices in Germany, before and after differentiating the time series.

Linear Model 1

The first model will be simple, and used to to replicate the research KYOS made to see whether using renewable energy sources, or more exactly, to use the photovoltaics and wind power production as main driving covariates for the energy price changes and see whether or not it explains the behaviours we note. To do that, the same integration done in Equation (4.1) was carried out on the two time-series containing combined wind on and offshore production and solar production. The idea is then to predict the change ΔP_h in price based on the previous days price P_{h-24} , the change in wind production ΔW_h and the change in solar production ΔS_h , where Δ denotes a 24 hour change, e.g. $\Delta S_h = S_h - S_{h-24}$. Written out in mathematical form this becomes

$$\Delta P_h = \beta_0 + \beta_P P_{h-24} + \beta_{\Delta W} \Delta W_h + \beta_{\Delta S_h} \Delta S_h \quad (4.2)$$

, where β_0 denotes the intercept. For further notations sake, in shorthand with symbols it becomes

$$\Delta P_h \sim P_{h-24} + \Delta W_h + \Delta S_h, \quad (4.3)$$

where all β s and the intercept are omitted from the equation for readability, as future linear regression systems grows in number of covariates.

The model is then fitted using sci-kit learn's linear regression framework [SKLLM], and yielded the following results:

β	lower	est	upper
β_0	-	8.44	-
P_{h-24}	-0.248	-0.245	-0.243
ΔS_h	-0.878	-0.837	-0.796
ΔW_h	-0.895	-0.883	-0.872

Table 4.1: Slopes for the different covariates in linear model 1.

As seen in the model, all covariates are significant, and shows that increasing production from wind and solar production sources links to a negative price change, while a decrease of said production would link to an increase in price, which is what one would expect. The price 24 hours ago merely sets a baseline on the next days price, along with the intercept of 8.45. The model explains about 43% ($R^2 = 0.426$) of the total variance and has an adjusted $R^2 = 0.426$. It can be seen in the residuals in Figure 4.2 that the spikes are narrower, although they are still there.

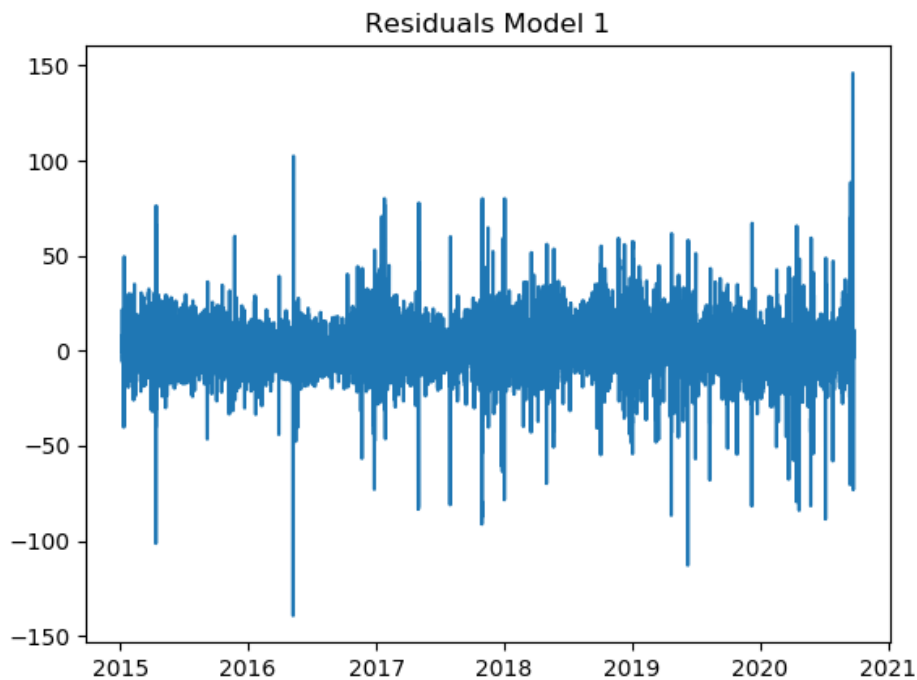


Figure 4.2: Residuals from linear model 1.

Linear Model 2

As the previous model had its flaws, the next model is going to expand on the inputs and examine whether running certain power plants at high output levels affects the prices notably, and this being a good predictor. This theory was tested on the fossil brown coal and fossil hard coalpower plants, as they are both burning fuel and burning more fuel corresponds to more output. The power production limit from fossil hard coal and fossil brown coal are about 25 GWh and about 20 GWh respectively, and a high output level was set to approximately half of that, to 12 GWh and 10 GWh respectively. The operation on the datasets then became:

$$\text{FHCHO}_i = \max(\text{FHC}_i - 12000, 0), \quad (4.4)$$

$$\text{FBCHO}_i = \max(\text{FBC}_i - 10000, 0), \quad (4.5)$$

where FHCHO means Fossil Hard Coal High Output and FBCHO means Fossil Brown coal High Output. The new model then becomes

$$\Delta P_h \sim P_{h-24} + \Delta W_h + \Delta S_h + \text{FHCHO}_h + \text{FBCHO}_h. \quad (4.6)$$

The results from the model fitted using scikit-learns framework were

β	lower	est	upper
β_0	-	8.09	-
P_{h-24}	-0.326	-0.322	-0.318
ΔW_h	-0.802	-0.762	-0.723
ΔS_h	-0.793	-0.781	-0.769
FHCHO_h	0.965	1.018	1.071
FBCHO_h	0.735	0.787	0.839

Table 4.2: Slopes for covariates in linear model 2.

With the high coefficients for the slopes for the newly introduced variables, as well as a higher R^2 value this time in 0.46, it can be concluded that high usage of fuel consuming power plants also affects the change in price. However, the adjusted R^2 stays the same on 0.426, thus while adding more parameters to the model, the model itself did not necessarily become better. The residuals looks very similar to linear model 1, but with slightly lower variance than before.

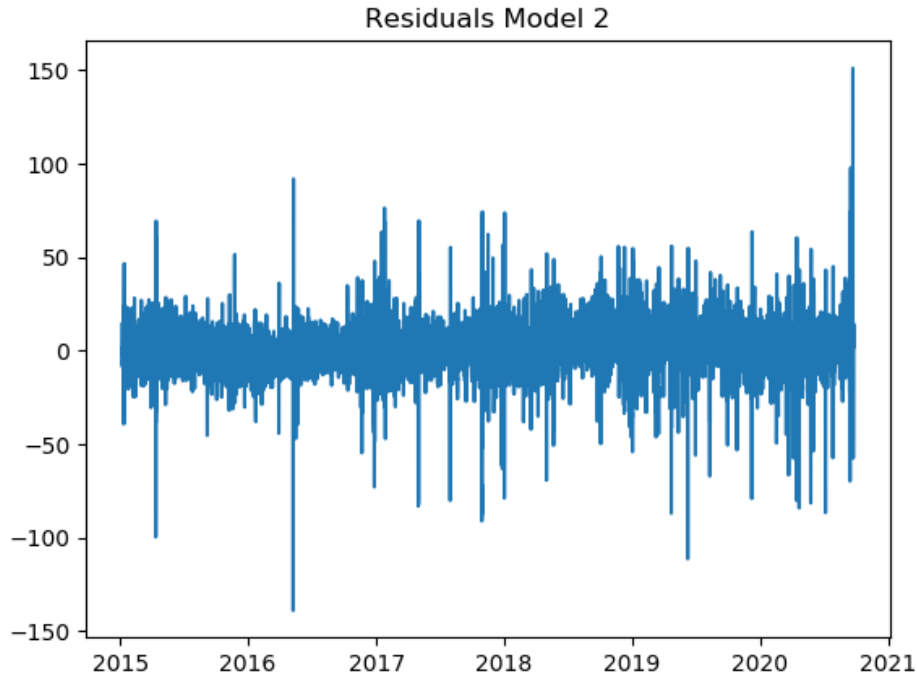


Figure 4.3: Residuals from linear model 2.

Final Linear Model Testing

To finalize this section, a model using all available possible explanatory variables was created, with the exception of other renewables and other conventionals. The variables were used in their normalized form, thus with inputs ranging from 0 to 1, depending on the load on the production type. Furthermore, the covariates used in model 2 will be added as well, which takes the total number of covariates up to 15. While some variables are used multiple times, they are used in different ranges, for example the solar production is used in a scale between 0 to 1 relative to its production capacity and the differentiated solar power production, which could both be useful for the model. If they are not, they will be deemed not significant and thus sorted out, and in the end leaving a model with only the useful explanatory variables.

The final model then had the following results:

β	lower	est	upper
β_0	-	6.44	-
P_{h-25}	-0.611	-0.605	-0.598
ΔS_h	-0.409	-0.376	-0.343
ΔW_h	-0.419	-0.407	-0.394
FHCHO _h	-0.518	-0.447	-0.376
FBCHO _h	-1.588	-1.502	-1.415
n_Solar _h	-1.308	-0.824	-0.339
n_Wind offshore _h	1.314	1.650	1.986
n_Wind onshore _h	-7.662	-6.950	-6.237
n_Hard Coal _h	20.794	21.773	22.753
n_Brown Coal _h	32.760	33.792	34.824
n_Biomass _h	-27.513	-26.353	-25.193
n_Gas _h	46.138	47.203	48.267
n_Hydro _h	-7.150	-5.999	-4.848
n_Hydro Storage _h	21.647	22.278	22.910
n_Nuclear _h	0.893	1.487	2.081

Table 4.3: Slopes for covariates in linear model 3.

with an intercept of 6.44, R^2 of 0.65 and adjusted R^2 of 0.426. The residuals can be seen in Figure 4.4 below:

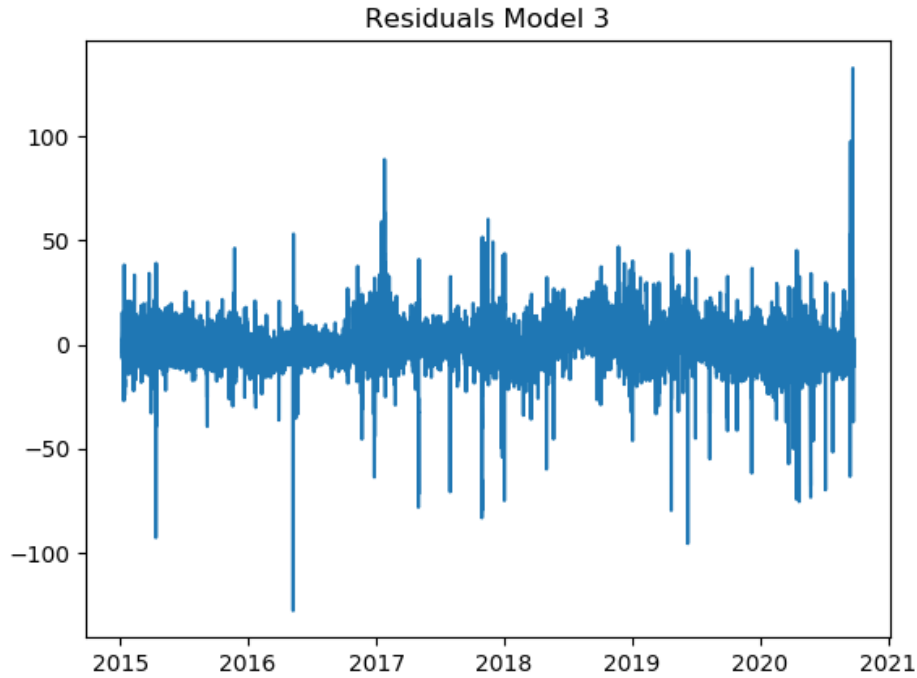


Figure 4.4: Residuals from linear model 3.

From Table 4.3 one can see an interesting result, that higher wind production offshore seems to correlate with increasing electricity prices, which is unexpected since wind power has no marginal increase in production cost. Overall, the last model seems to fit the data the best, with a visibly variance compared to the previous residual plots, however the overall quality of the prediction has not increased if one look at the adjusted R^2 . All in all, it is clear that since electricity prices are not linear and with the weakness of integrated data having sequential errors piling up when reconstructing data, a non-linear method should be considered.

4.2 Naive Gaussian Process Regression Approach

As noted in the previous section, a linear regression approach on a clearly non-linear time series is not ideal. While the integrated time series seems to be stationary enough for linear regression to work, it still has the weakness of any integrated process, namely that the errors are sequential and adds up over time, creating a bigger and bigger error margin as we predict further in time. Thus leading to a search for more suitable methods, and a first method to investigate is Gaussian Process Regression (GPR). While introducing new concepts, a transformation on the price data P should be made as well.

As noted in the previous section, the data was possibly heteroscedastic, which can be explained by inflations and the general price level of electricity changing. Thus to prevent the inflations and other unknown factors affecting the results by changing the average price level, a transformation called "Hour-to-Month Ratios", or H2M-ratios for short, is introduced. It is defined as the hourly price divided by the monthly average price:

$$\text{H2M}_i = \frac{P_i}{\frac{1}{|J|} \sum_{j \in I} P_j} \quad (4.7)$$

where I denotes the set of indices belonging to the same month as entry i .

The problem with varying price levels for each year can be seen in Figures (4.5) and (4.6) below, where the first figure shows the electricity spot prices for March in Germany for each year in the dataset, while the second figure with subfigures shows the prices and H2M ratios over the entire dataset. In Figure 4.1b one can note that the price level swings up and down over the entire time period, while in Figure 4.6b it stays far most stable around 1 to 1.5.

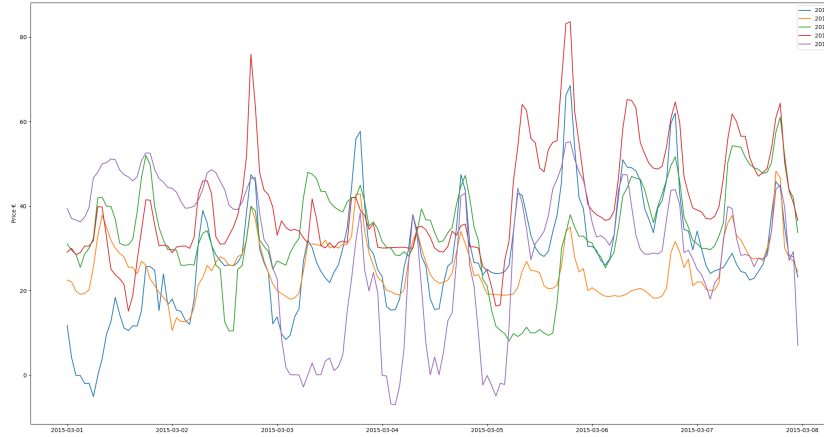
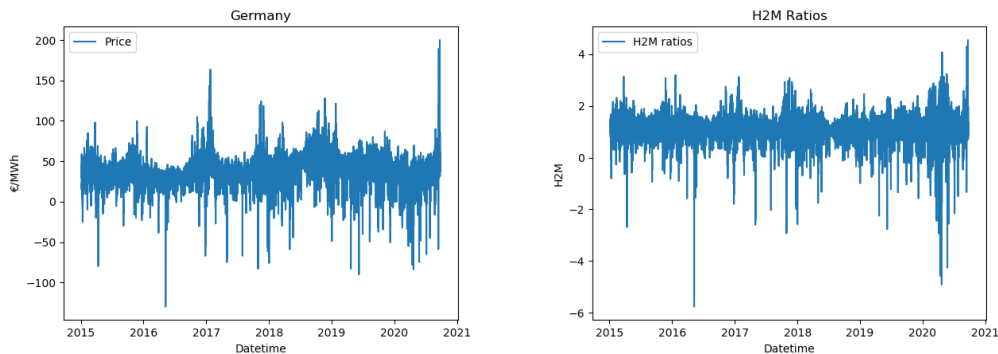


Figure 4.5: Electricity spot prices for March in Germany, for years 2015-2019



(a) Electricity Prices Germany

(b) H2M ratios, Germany.

Figure 4.6: Electricity prices in Germany, before and after transforming the time series to H2M ratios.

Using this new price level independent representation of electricity prices, a naive approach was made to directly fit all data using the GPR framework in scikit-learn. The output was set to $H2M_h$ and inputs as nW_h and nS_h , the normalized total wind production and normalized solar production, and the kernel is initialized as a product of a simple RBF kernel and three periodic kernels spanning a week, a month and a year as well as an additive noise kernel to allow for modelling with noise. However, this approach quickly failed

as the covariance matrix grew too large to be stored in the memory, which led to some workarounds being necessary. Instead of taking the whole dataset at once, instead a shorter time period from year 1 was given to the model to train on. And then the predictions would be on the same time period at later years in the data. On the first 3000 datapoints of each year, corresponding to the period January-April, the following results were obtained (Figure 4.7):

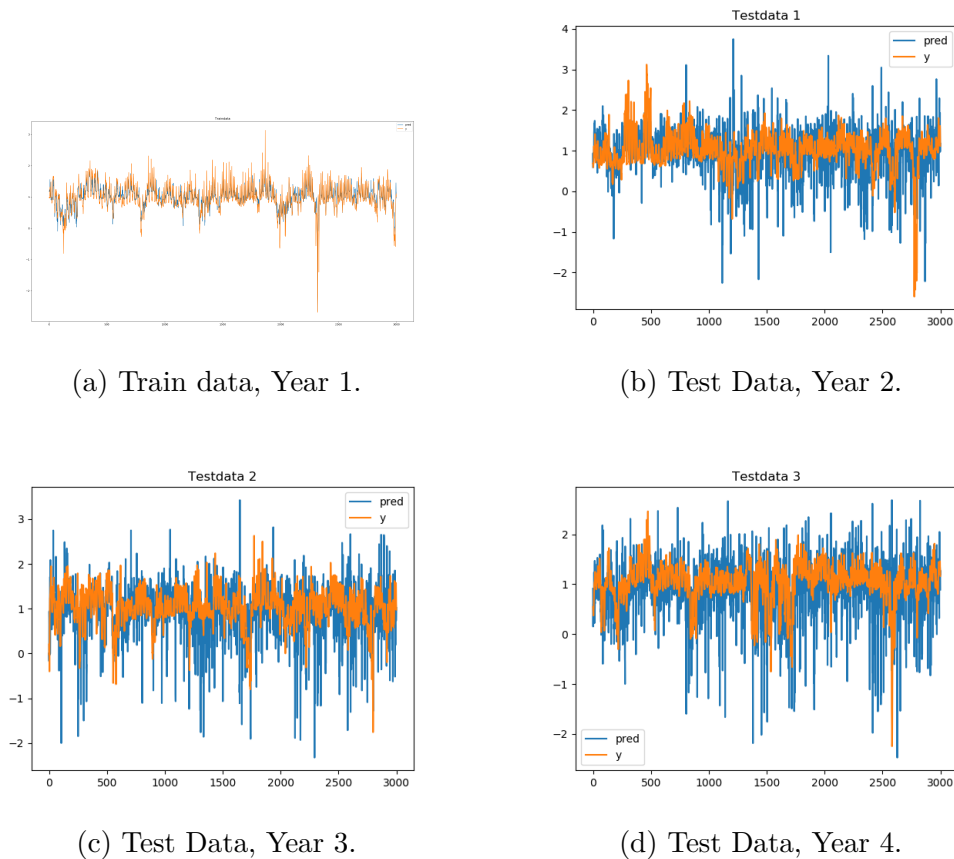


Figure 4.7: Train data and test results for the naive GPR.

As seen, the naive approach with periodic kernels to capture the weekly, monthly and yearly electricity price behaviours did not work well at all. While the predictions works fairly well on the training set, it clearly does not come close to giving a proper prediction on the same time period at later years. This can be explained by the fact that the model only gets each hour of each day of the year once, e.g. April 4th 22:00 only appears once in the dataset, leading to insufficient information to capture the behaviour of that

said hour. Thus a whole different approach is necessary if Gaussian Process Regression is to be used.

4.3 Gaussian Process Regression Using Dummy Variables

As found in the previous section, one simply cannot use the full dataset at once, it simply isnt feasible to construct a covariance matrix that big. Thus instead, another way to attempt to feed the model multiple input-output pairs of the same hours was done, this time using dummy variables. Two different kinds of one-hot encoding was done on the data, to represent the hour, day, month and year of the datapoint. The first one is a regular sharp 0-1 one-hot encoding, where it is simply encoded as a 1 if it belongs to the category or 0 if not. An example would be categorizing the years using one-hot encoding:

Date	2015	2016	2017	2018
2016-04-13	0	1	0	0
2018-11-03	0	0	0	1

Table 4.4: Example of one-hot encoding. Where the entry in the column is 1 if it belongs to the category (year) and 0 otherwise.

For the second kind of encoding, used only for categorizing months, the wish to incorporate information from the previous month to get a gauge of the price (or H2M ratio) development spawned a need of a smoothed version of one-hot encoding, where low but non-zero values would represent the entries leading in to the month of interest, while quickly raising to 1 during the month. This was accomplished by the function

$$\text{SOHE Month}_i = \max \left(0, 1 - \left(\frac{i - c}{w} \right)^4 + s \right) \quad (4.8)$$

$$\text{SOHE Month}_i = \min (\text{SOHE}_i, 1) \quad (4.9)$$

where c is the center index of the month, w is how many datapoints the month spans and a constant s denoting how far we shift the curve, which determines how many datapoints to be included from the neighbouring months. In the experiments, s was set to 0.5, which lets it add 64 datapoints, approximately

2 and a half day, from the neighbouring months. The two equations together limits the final result to a range between 0 and 1.

The outputs for all models in this section using dummy variables will be H2M ratios, to be easily compared to each other. The inputs will vary between the models, but all of the models will use the explanatory variables normalized solar production nS_h , normalized total wind production nW_h and normalized residual load nr_h . The kernels will vary slightly as well between models in the same model group, with the purpose of testing how major the difference between a Matérn kernel and RBF kernel is in practical use. The kernels in each model group will be constructed as follows:

$$k_{\text{Model a}} = k_{\text{RBF}} + k_{\text{Constant}} + k_{\text{noise}} \quad (4.10)$$

$$k_{\text{Model b}} = k_{\text{Matérn 1.5}} + k_{\text{Constant}} + k_{\text{noise}} \quad (4.11)$$

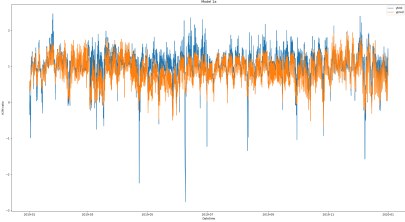
$$k_{\text{Model c}} = k_{\text{Matérn 2.5}} + k_{\text{Constant}} + k_{\text{noise}} \quad (4.12)$$

By using the same data but with three slightly different kernel constructions, the impact of the kernel choice can be directly compared within each model group.

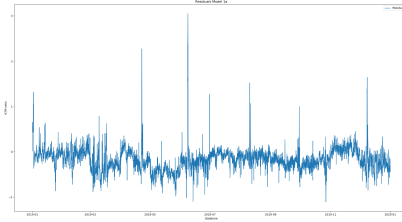
Throughout this section, the models are trained in scikit learns GPR framework, by inputting and training the model one month at a time, and predicting the results month by month, to work around previously found limitations. The predictions are then concatenated and added together to span an entire year. For each month, the training set will consist of all data in the time period 2015-2018 and the test set, where the model will predict the future price, will be on 2019. The prediction errors are then presented as an average over the entire year, and detailed results for each individual month can be found in the Appedix (B) in the end.

4.3.1 GPR Model Group 1

In model group 1, the inputs will be the default inputs and use the standard one-hot encoding as dummy variables. These models will serve as a baseline and reference for comparison against changes in inputs in the other model groups. The resulting predictions and residuals can be seen in Figures (4.8), (4.9) and (4.10). The mean squared error (MSE) and mean absolute error for the models can be found in Table 4.5.

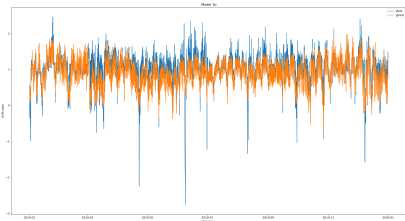


(a) Predictions and actual values.

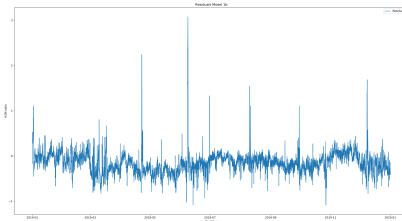


(b) Residuals

Figure 4.8: Model 1a predictions and residuals.



(a) Predictions and actual values.

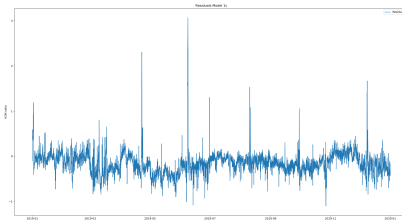


(b) Residuals

Figure 4.9: Model 1b predictions and residuals.



(a) Predictions and actual values.



(b) Residuals

Figure 4.10: Model 1c predictions and residuals.

Model	MSE	MAE
1a	0.095	0.228
1b	0.083	0.212
1c	0.087	0.215

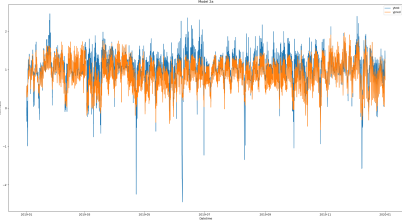
Table 4.5: MSE and MAE between predictions and actual outputs for Model 1a, 1b and 1c.

As seen in the residuals, these residuals are clearly not i.i.d., however that is to be expected. Since the models has been trained one month a time, and thus every month will have its own model structure, its own volatility and variance.

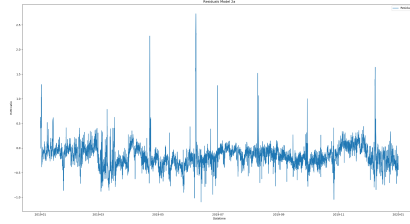
4.3.2 GPR Model Group 2

In model group 2, the output y is modified slightly. As seen in the data, there are a few very high price spikes and some far negative price drops that are difficult to predict, and as thus an attempt was made where extreme outliers were set to the 99th and 1st quantile values. The idea is to get a better prediction on the H2M ratios in a normal scenario. While this does not remove all outliers, especially the negative price levels, it should not do that either, as negative electricity prices is fairly common in Germany at night. However, important to note is that for these models to work in practice, another system to predict price spikes and drops is necessary, although not researched in this thesis.

The predictions against the real data and residuals can be seen in Figures (4.11), (4.12) and (4.13) below, and the prediction errors can be found in Table 4.6.

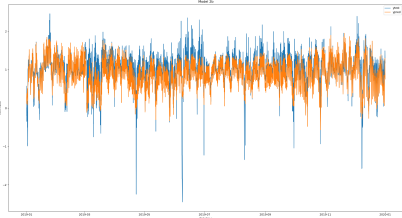


(a) Predictions and actual values.

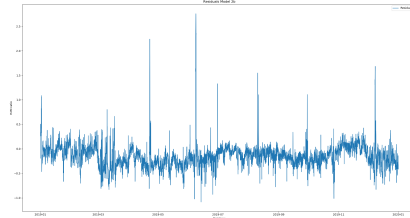


(b) Residuals

Figure 4.11: Model 2a predictions and residuals.

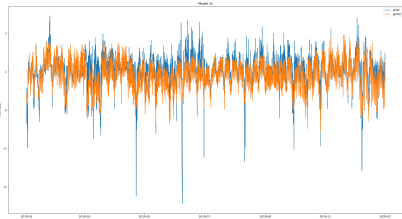


(a) Predictions and actual values.

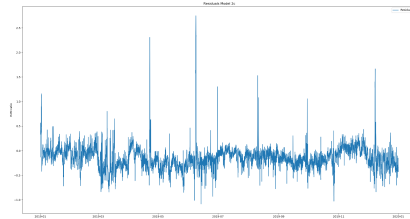


(b) Residuals

Figure 4.12: Model 2b predictions and residuals.



(a) Predictions and actual values.



(b) Residuals

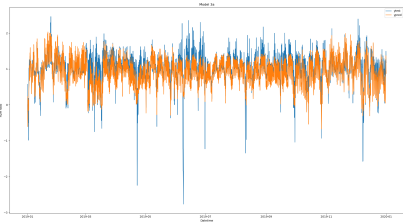
Figure 4.13: Model 2c predictions and residuals.

Model	MSE	MAE
2a	0.085	0.216
2b	0.079	0.207
2c	0.079	0.205

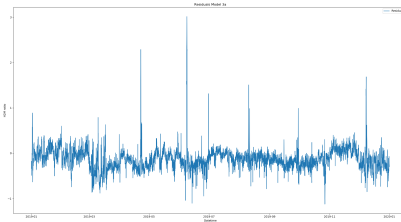
Table 4.6: MSE and MAE between predictions and actual outputs for Model 2a, 2b and 2c.

4.3.3 GPR Model Group 3

In model group 3, the same inputs and outputs from model group 1 was used, with the exception of the dummy variables for which month the input and output belongs to. Instead of using the regular one-hot encoding the smoothened version in Equation (4.9) was used. The resulting predictions can be shown below, shown in Figures (4.14), (4.15) and (4.16). The different models MSE and MAE can be seen in Table 4.7.

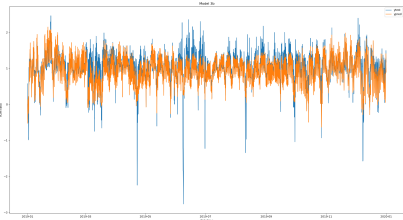


(a) Predictions and actual values.

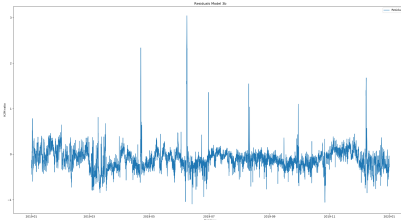


(b) Residuals

Figure 4.14: Model 3a predictions and residuals.

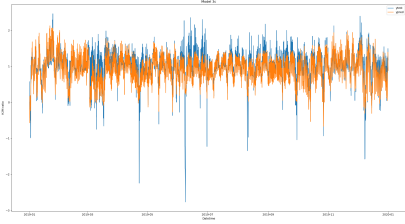


(a) Predictions and actual values.

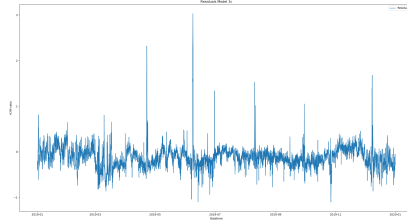


(b) Residuals

Figure 4.15: Model 3b predictions and residuals.



(a) Predictions and actual values.



(b) Residuals

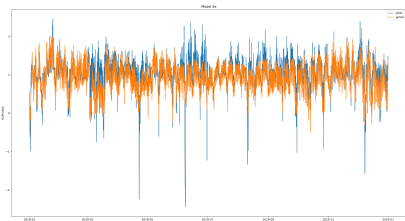
Figure 4.16: Model 3c predictions and residuals.

Model	MSE	MAE
3a	0.074	0.198
3b	0.071	0.190
3c	0.072	0.193

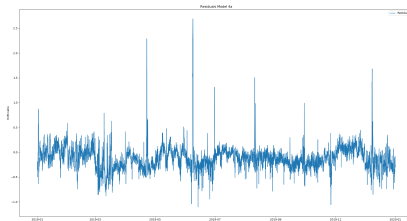
Table 4.7: MSE and MAE between predictions and actual outputs for Model 3a, 3b and 3c.

4.3.4 GPR Model Group 4

In model group 4 both the changes applied to model group 2 and 3 were applied, primarily as a test to see if both using the outlier fixed dataset and the smoothening of the dummy variables for each month might be too much post processing on the data. The resulting predictions can be seen below in Figure 4.17, (4.18) and (4.19). The errors of the model can be found in Table 4.8.

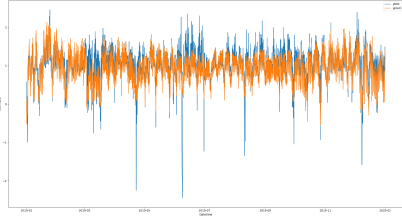


(a) Predictions and actual values.

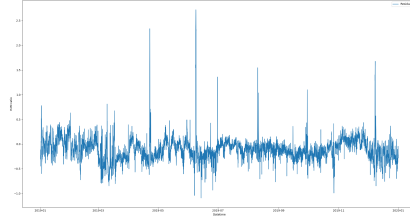


(b) Residuals

Figure 4.17: Model 4a predictions and residuals.

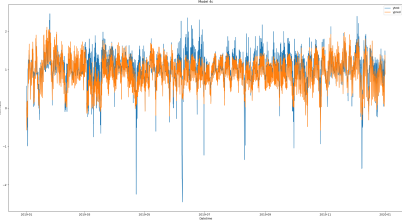


(a) Predictions and actual values.

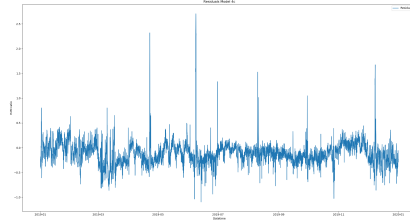


(b) Residuals

Figure 4.18: Model 4b predictions and residuals.



(a) Predictions and actual values.



(b) Residuals

Figure 4.19: Model 4c predictions and residuals.

Model	MSE	MAE
4a	0.073	0.196
4b	0.070	0.189
4c	0.071	0.192

Table 4.8: MSE and MAE between predictions and actual outputs for Model 4a, 4b and 4c.

4.4 XGBoost

In this section, the results for the models created with the XGBoost framework are presented. These models are trained with root mean square error (rmse) as loss function, $\eta = 0.3$ and max depth = 6, and otherwise default settings in the XGBoost framework. The inputs and outputs for

XGBoost model 1 will be identical to GPR Model Group 1, and for XGBoost Model 2 it will use the input and outputs from GPR Model 2. The smoothed dummy variables will not be used in this section, as the model framework support inputting the entire training set at once, and thus the extra datapoints leading into each month is not necessary to gauge the price level.

In Figures (4.20) and (4.21) the resulting predictions with the two datasets and their residuals are shown, and the errors can be found in Table 4.9.

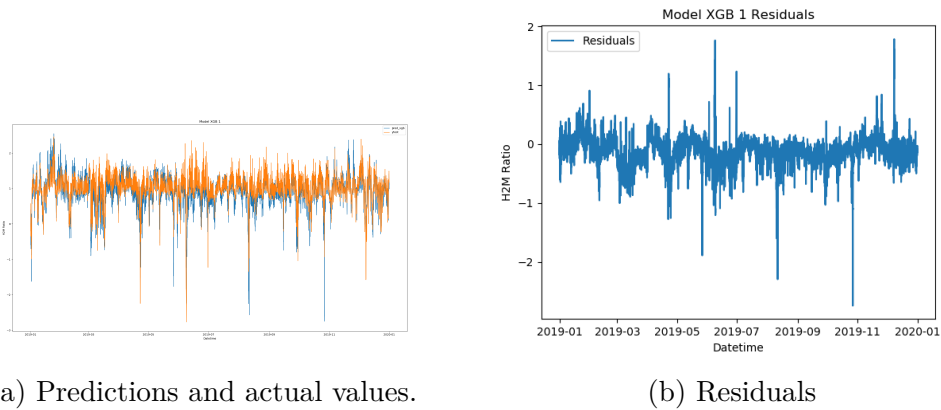


Figure 4.20: Model 1 predictions and residuals.

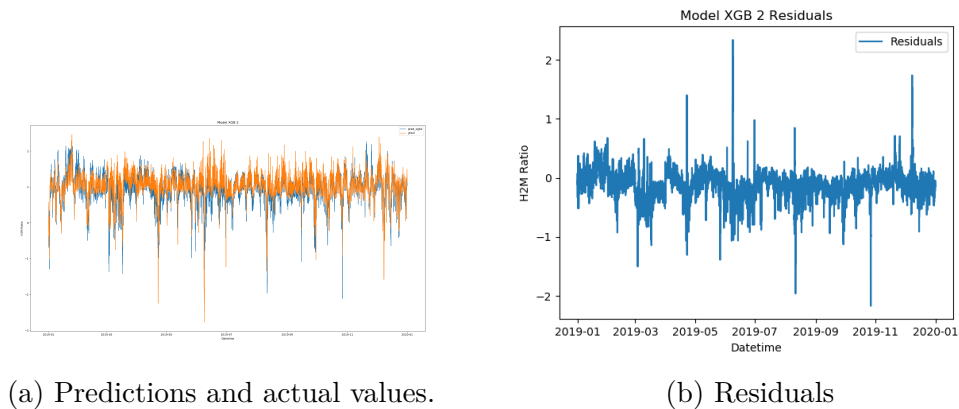


Figure 4.21: Model 2 predictions and residuals.

Model	MSE	MAE
XGBoost 1	0.077	0.214
XGBoost 2	0.075	0.196

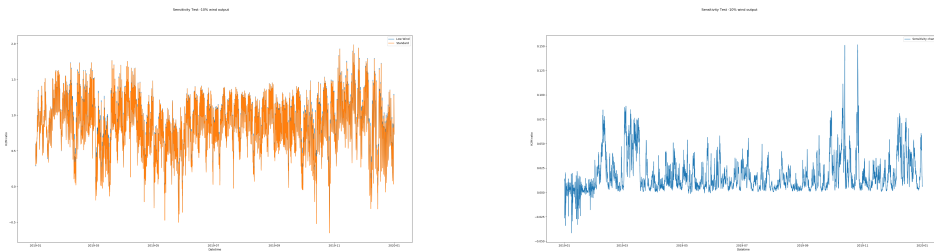
Table 4.9: MSE and MAE for the two XGBoost Models.

4.5 Sensitivity Testing of the Models

Before discussing and evaluating the suitability of the model frameworks for the task, a test needs to be made to ensure that the model reacts the correct way to unexpected changes in inputs. In other words, if one month suddenly has a lot lower electricity production from renewables, say wind production, one would expect the prices to rise as a consequence as the residual load has to increase to match the demand. This will be tested in this section with GPR Model 1a and XGB model 1, to make sure the models behaves as one would expect.

Low Wind Scenario

Testing the theoretical scenario where the wind production is 10% lower than normal for the time of the year. For GPR model 1a, the prediction change can be observed in Figure 4.22:

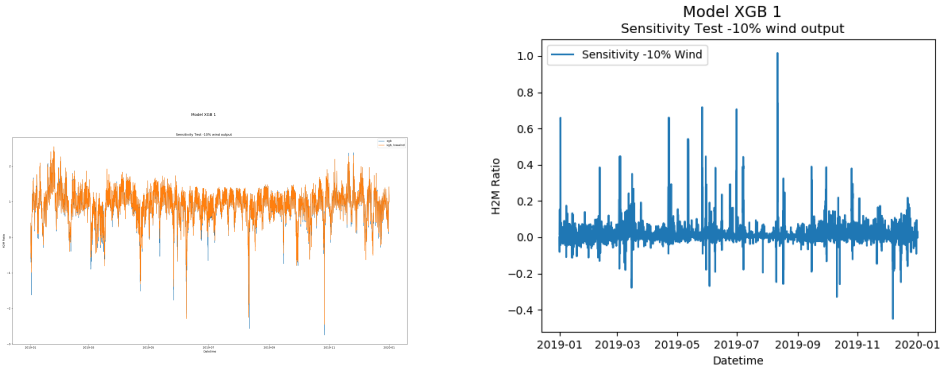


(a) Predictions for low wind and regular wind. (b) Difference between low wind and regular wind predictions.

Figure 4.22: Prediction Results with low wind, GPR.

For GPR Model 1a, the average prediction was changed by 0.018, leading to a shift upwards in predicted H2M ratios.

For the corresponding XGBoost model, a similar shift can be observed in Figure 4.23:



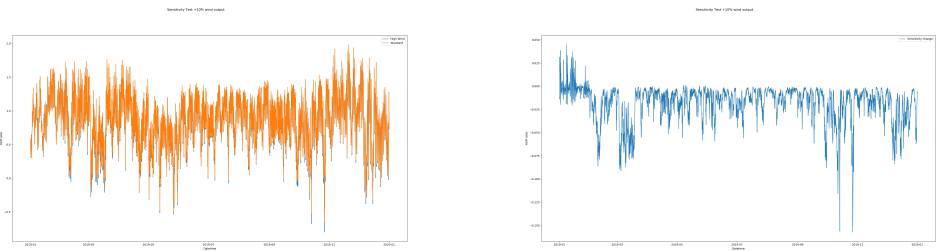
(a) Predictions for low wind and regular wind. (b) Difference between low wind and regular wind predictions.

Figure 4.23: Prediction Results with low wind, XGBoost.

In the XGB model, the corresponding shift equated to 0.016, which is about identical to the GRP model.

High Wind Scenario

To test the corresponding scenario where the wind production instead is increased by 10%, the GPR model 1a yielded the following change in prediction, seen in Figure 4.24.

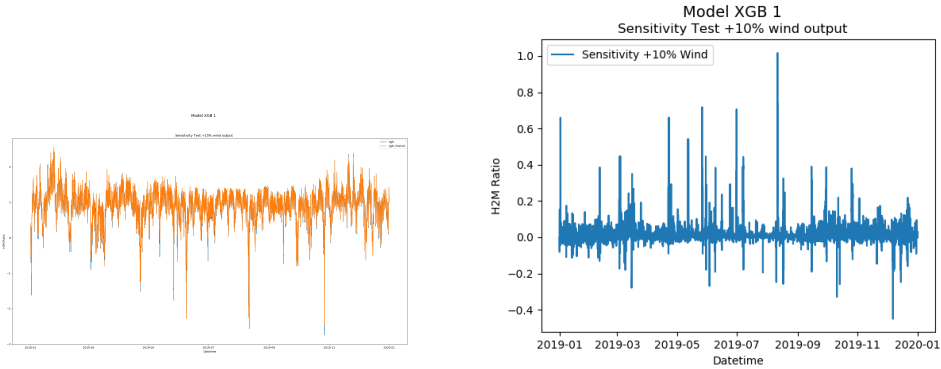


(a) Predictions for high wind and regular wind. (b) Difference between high wind and regular wind predictions.

Figure 4.24: Prediction Results with low wind, GPR.

For GPR Model 1a, the average prediction was shifted by -0.018 , leading to a negative in predicted H2M ratios.

For the corresponding XGBoost model, a similar shift can be observed in Figure 4.25:



(a) Predictions for low wind and regular wind. (b) Difference between low wind and regular wind predictions.

Figure 4.25: Prediction Results with low wind, XGBoost.

In the XGB model, the corresponding shift equated to 0.016 , which lead to an increase in predicted H2M ratios, which contradicts the expectations.

Chapter 5

Discussion

How good are renewables as predictors?

As seen in the testing in section 4.1, one could see how the residuals variance gradually decreased as more features were added to the model. Additionally, the R^2 score increased as more features were added to the model. Thus the models did get better as the models became more complex, which is as to be expected. However, R^2 as a metric is not perfect, as it only explains the goodness of fit for the model to a data, which will always get better, or at worst be the same, as one adds more features to a model. Thus one should also take a look at the adjusted R^2 to get another picture on whether the more complex models provided a better prediction or not, and in which all three models scored the same. The adjusted R^2 aims to better explain when a feature added to the model is a poor fit, and thus decreasing the quality of the model. By such, the adjusted R^2 can at most be the same as the R^2 score. And looking at the adjusted R^2 from the three models, we can see that they scored the same, thus showing that the added features did not explain more than what was already explained by ΔW_h and ΔS_h . Thus it is fair to conclude that wind and solar power are very good explanatory variables for the electricity price in Germany.

RBF or Matérn kernel

Across the board on all the model groups, the models using a Matérn kernel are outperforming their RBF kernel based equivalent. This is likely

connected to what was discussed in the theory section 2.3.2 about RBF and Matérn kernels, that while the RBF kernel has a lot of nice properties in theory, it performs worse in practical scenarios as the conditions are not fulfilled. However, the difference is not as clear between the Matérn kernels with $\nu = 1.5$ and $\nu = 2.5$, thus suggesting that they might both be valid choices for the purpose. The length scales however are not as deterministic across the months, but rather vary between months, suggesting that different months have different patterns of different length scales which it tries to capture.

Underlying Problems with the GPR Approach

While the Gaussian Process Regression should work in theory, as the output is generally smooth and does not fluctuate wildly in short periods of time in a general sense, with the exception of price spikes, it still has its other problems. As seen in section 4.2, using big datasets leads to problems such as singular value decomposition when inverting the covariance matrix, making the method unsuitable for large datasets. This is due to the need to invert the covariance matrix during the optimization, an operation that scales in $\mathcal{O}(n^3)$. The problems with large dataset was then attempted to be bridged by splitting the prediction into month by month predictions instead, which however lead to other problems.

First, by no longer having any memory of the transition between months, the price level going into the month, which might lead to jumps in the output (H2M ratio). While one might be able to mitigate some of that by predicting with overlap between the months and averaging the predictions for each timepoint, that suddenly requires even more computing power, and thus was not done here. Secondly, the month-by-month prediction method also loses out on information. This is due to Mondays appearing over all of the year, but when learning about the behaviours of a Monday, the model only gets 1/12th of the total information, since it only gets fed all the training data for January months when learning about January. This is likely hindering the model from being as good as it possibly can be. Furthermore, there are more information that could be useful for the model, for example fuel price along with the fuel consumption for certain plants could improve the accuracy too. These kind of inputs were, however, left out as the purpose of this thesis is to evaluate the suitability of the framework for the task.

Model Sensitivity

As mentioned in section 4.5, for the model to be useful it needs to react realistically to a new presented scenario. This was tested in the previous mentioned section, by presenting a high wind power production scenario and a low wind power production scenario to the models and gauging their response. The hypothesis is that a decrease of power production from renewables leads to higher electricity prices, as the demand now has to be matched by conventional powerplants which has a running cost associated to them. Or in the worst scenario, not made up for at all, which further leads to higher prices. This response was seen in the GPR model, where the predicted H2M ratios rose when faced with a low wind production scenario and dropped when faced with a high wind scenario. In the XGBoost model, the same response was had regardless of wind scenario, it reacted the same way in both scenarios, leading to the belief that either the framework was used wrong or the reaction from the model being unrealistic.

Suitability of the Frameworks

Thus leading us to the final discussion, the evaluation of the frameworks chosen in this thesis. The linear model will not be discussed here, as it is clear that the price trends are not linear and thus a linear model are not suitable.

The GPR models with dummy variables seems up for the task, to create a model that predicts the price level close to the actual price level. However, as discussed previously, it falls short when the output starts changing rapidly in a short timeframe, e.g. a spike. However, for the big parts of the data it works well and suits the model assumptions well too, and will work great as long as the datasets are kept in reasonable size. The models also react well to changes in scenarios, where the predictions change in the correct way when inputs change. Furthermore, this could be expanded upon with Gaussian Process Approximation models, where scalability of the GPR is in mind for its construction. A further read of this can be found in [GPBD]. Thus the GPR could be a worthwhile framework to look further into.

The XGBoost models however, are a bit different to talk about. While they result in similar prediction errors as the Gaussian models with less time spent on the framework, they fall short in a crucial step: they do not seem

to react well to a change in scenario. While this could be due to the lack of time with the model, it also creates a worry that the framework simply is not suitable for the cause.

Methodology of predictions

In the models, all hours has been sequentially predicted, and thus in reality used a bit more information than should be available should we be predicting day-ahead prices. But since the predictions are more along the lines of a price forward curve, and predictions are made on a medium long timeframe, the issue of having a previous hours value when predicting the next hour should not be a big issue. Another thing to note is that the model does not actually yield a price prediction in the end, but rather a H2M value - a value that needs to be calibrated against peak and average load futures to actually yield price data. These models are of a complete different nature and thus not part of this thesis.

Chapter 6

Conclusion

To conclude the thesis, a wrap around to answer the research questions is in order. As seen in the linear models in chapter (4.1), using the differentiated wind production and solar production data, one can get a decent prediction, which suggests that they are clear driving forces of the prices, and thus suitable predictors. Along with the residual load, those three covariates together gives a reasonable prediction.

As for the model structures, neither model can be said to be perfectly suitable for the task. While the XGBoost framework has no problem with scalability, it has issues with the output and predictions not reacting properly to a change in scenario, as well as binary trees having hard cutoffs in its classification in each leaf can potentially lead to very sharp cutoffs, which is not what we would expect in the pricing. Likewise, the Gaussian Process Regression has its issues too, with poor scalability with big data and poor reactions to spikes, while otherwise fulfilling model assumptions well. Thus, it is likely that other model frameworks, such as Gaussian Process Approximations which retains similar behaviours as GPR but enforcing sparsity in the covariance matrix, or Massively Scalable Gaussian Processes (MSGP), which makes use of the Toeplitz structure of matrices to reduce the complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$ for inference and $\mathcal{O}(n^2)$ to $\mathcal{O}(1)$ for test point predictions, are a better path forward for predicting the future electricity prices.

Bibliography

- [KYOS 2020] C. de Jong <https://www.kyos.com/wp-content/uploads/2020/06/Creating-pr>
KYOS Energy Consulting, June 2020
- [W1] https://en.wikipedia.org/wiki/Supervised_learning
- [W2] https://en.wikipedia.org/wiki/Kullback-Leibler_divergence
- [W3] https://en.wikipedia.org/wiki/Akaike_information_criterion
- [XGBModel] <https://xgboost.readthedocs.io/en/latest/tutorials/model.html>
- [SMARD] <https://www.smard.de/page/en/wiki-article/5884/5840>
- [SMARD CBF] <https://www.smard.de/page/en/wiki-article/5884/6012>
- [SMARD Cons] <https://www.smard.de/page/en/wiki-article/5884/6036>
- [SMARD Prod] <https://www.smard.de/page/en/wiki-article/5884/6124>
- [Pandas] <https://pandas.pydata.org/docs/>
- [KYOS 2014] V. Beolet, C. de Jong, E. Enev. *Improved hourly shaping using renewable production information* KYOS Energy Consulting, 2014
- [KYOS 2013] C. de Jong, H. van Dijken, E. Enev. *How renewables shape the future* KYOS Energy Consulting, 2013
- [GPML] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning* The MIT Press, 2005
- [KRNL] D. Duvenaud *The Kernel Cookbook: Advice on Covariance Functions* <https://www.cs.toronto.edu/~duvenaud/cookbook/>

[GBM] Friedman, Jerome H. *Greedy Function Approximation: A Gradient Boosting Machine* The Annals of Statistics, vol. 29, no. 5, 2001, pp. 1189-1232. JSTOR, www.jstor.org/stable/2699986

[SKLLM] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.L

[GPBD] Liu H, Ong Y.S, Shen X and Cai J. *When Gaussian Processes MEets Big Data: A Review of Scalable GPs* Transactions on Neural Networks and Learning Systems, vol 31, no. 11, pp. 4405-4423. IEEE <https://ieeexplore.ieee.org/abstract/document/8951257>

Appendix A

Supplementary plots

In this section, supplementary plots which might be of interest to the reader are provided. These plots are generally informative but not informative enough to be part of the main body and thus shuffled to this section.

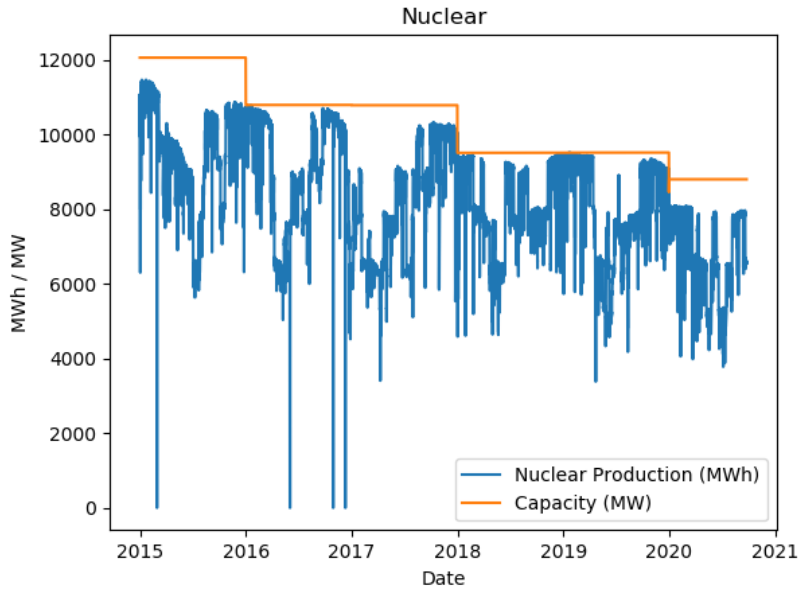


Figure A.1: Nuclear power production in MWh (blue) and the production capacity (orange).

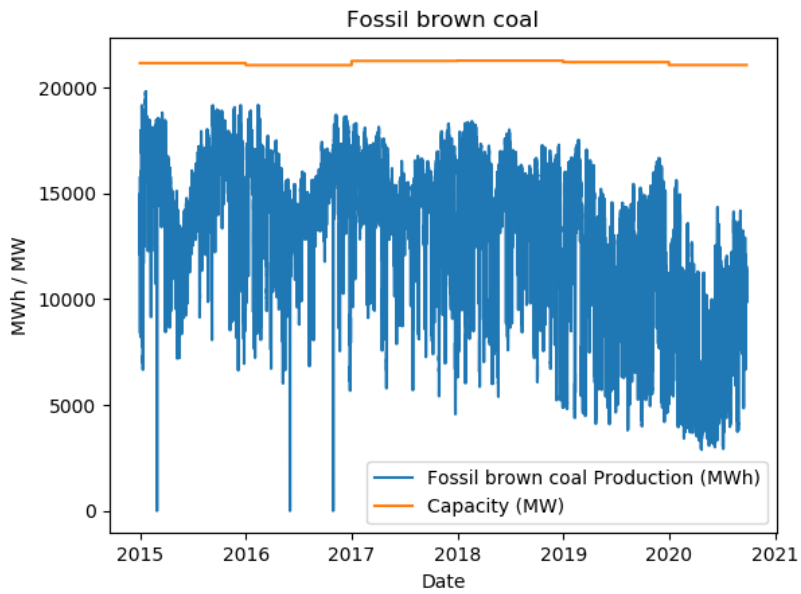


Figure A.2: Fossil brown coal power production in MWh (blue) and the capacity (orange).

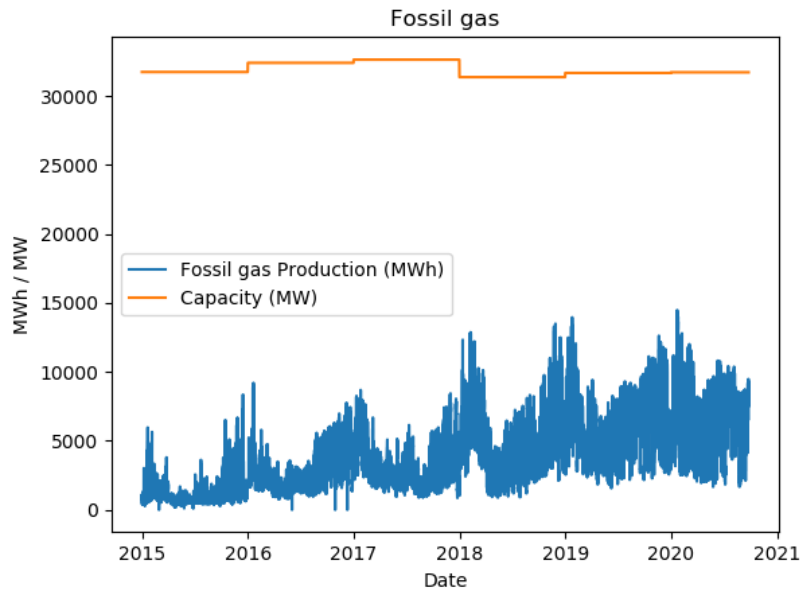


Figure A.3: Fossil gas power production in MWh (blue) and the capacity (orange).

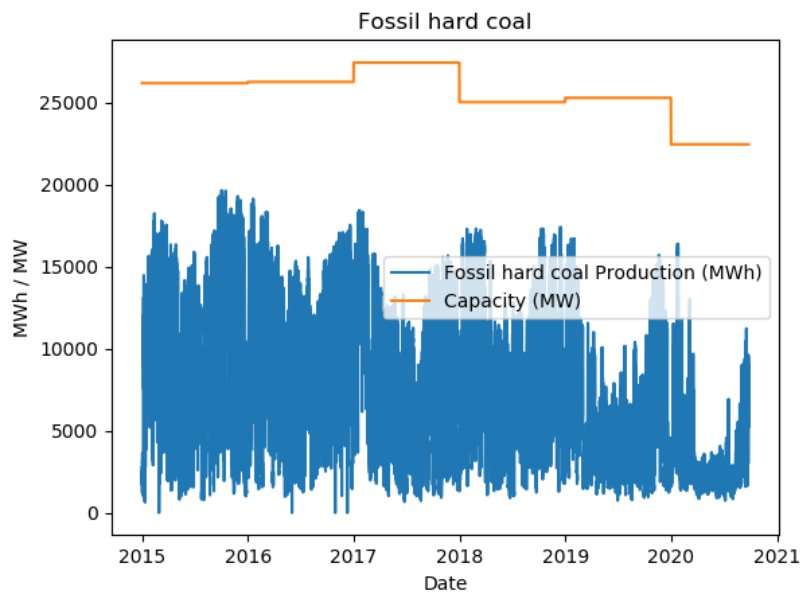


Figure A.4: Fossil hard coal power production in MWh (blue) and the capacity (orange).

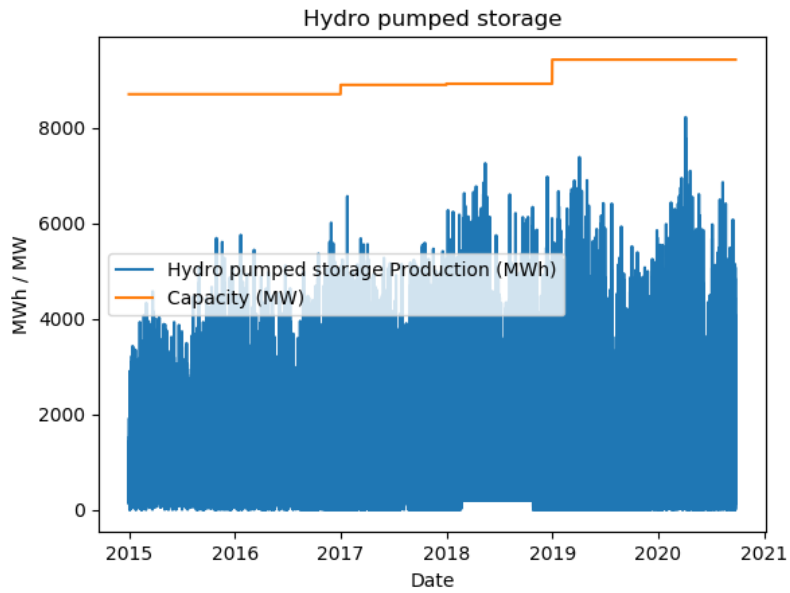


Figure A.5: Hydro-pumped storage power production in MWh (blue) and the capacity (orange).

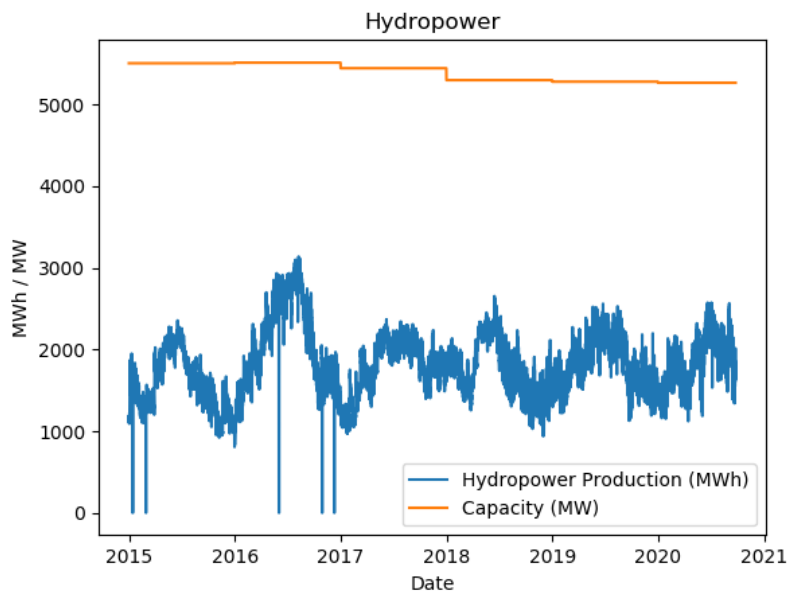


Figure A.6: Hydropower power production in MWh (blue) and the capacity (orange).

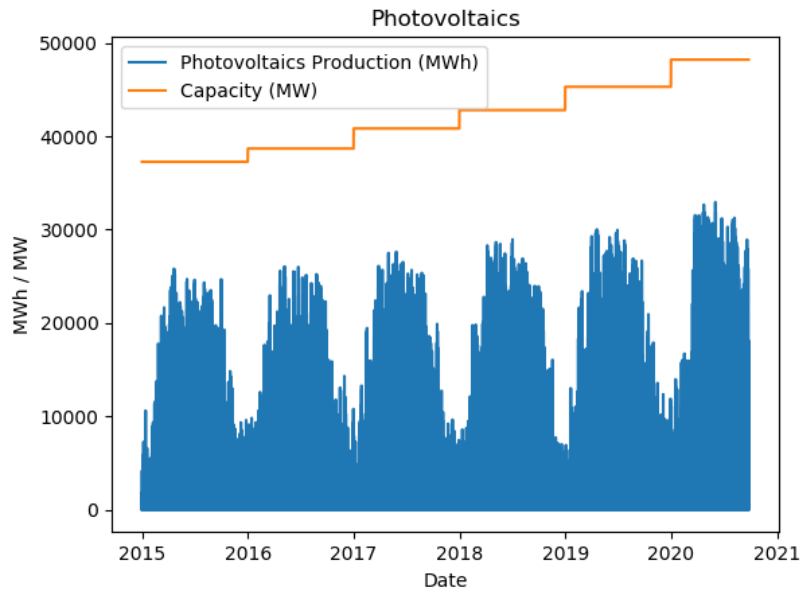


Figure A.7: Solar power production in MWh (blue) and the capacity (orange).

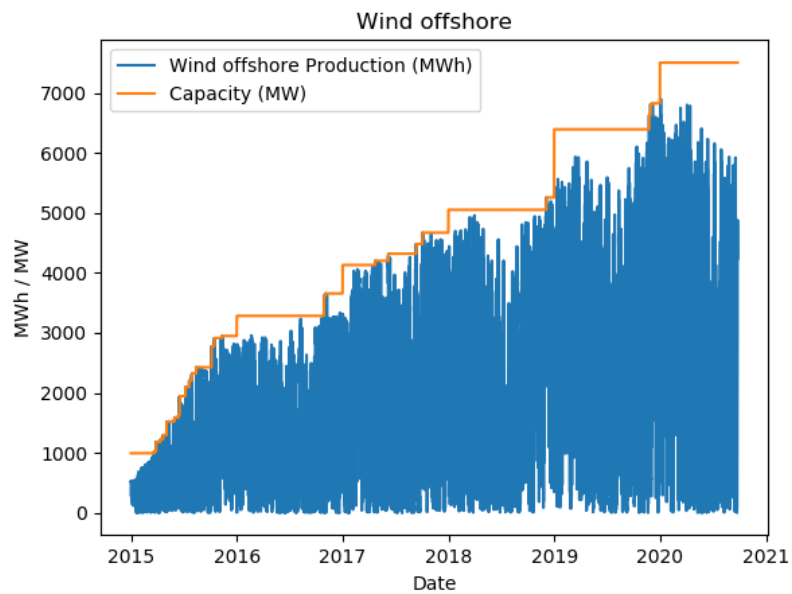


Figure A.8: Wind production from offshore systems in MWh (blue) and the capacity (orange).

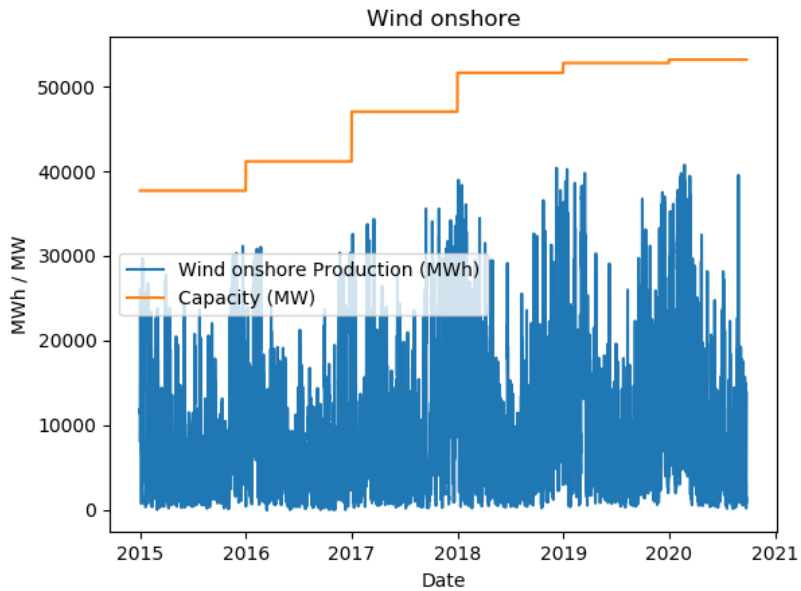


Figure A.9: Wind production from onshore installments in MWh (blue) and the capacity (orange).

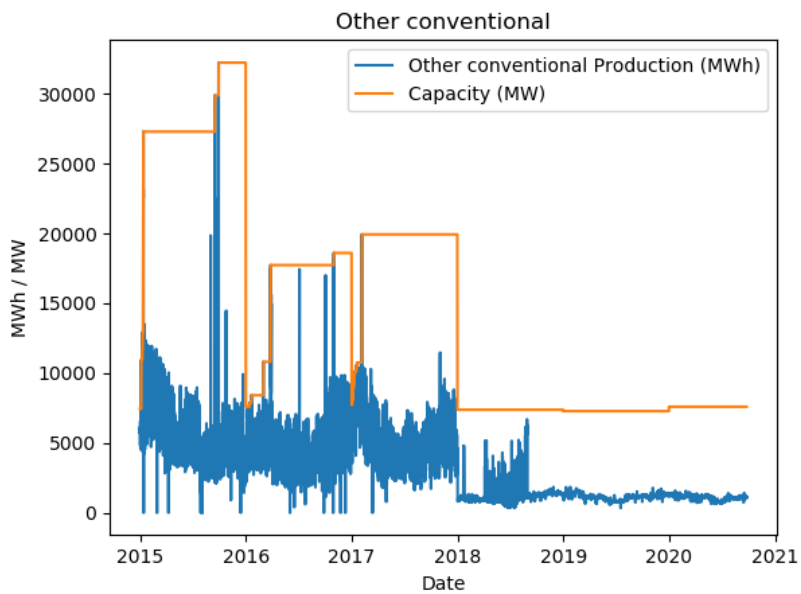


Figure A.10: Power production from other conventional sources in MWh (blue) and the capacity (orange).

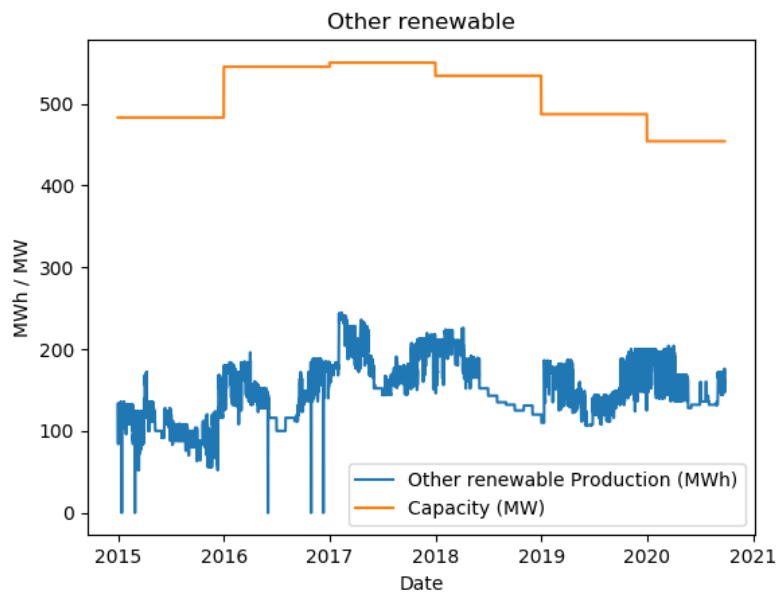


Figure A.11: Power production from other renewable sources in MWh (blue) and the capacity (orange).

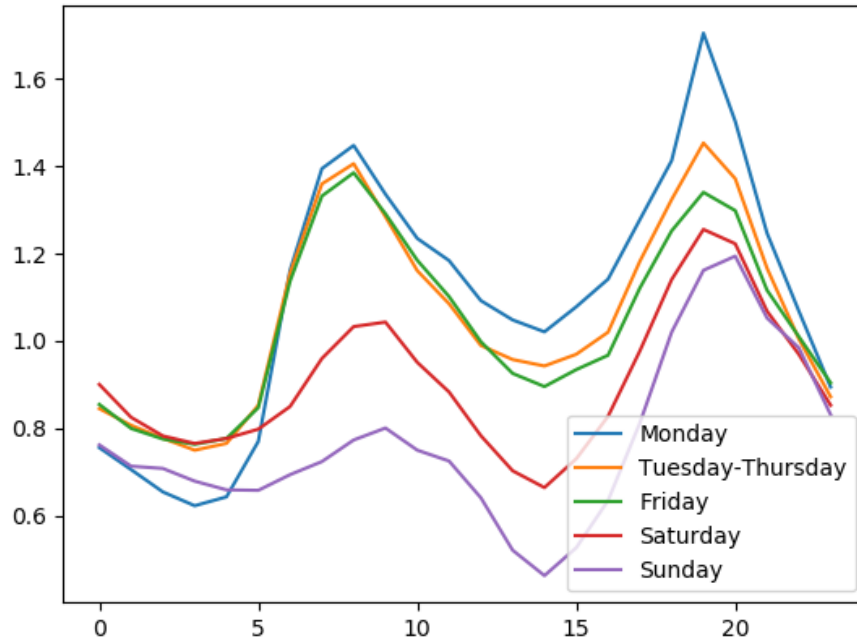


Figure A.12: Average H2M ratio for each day type. Tuesday to Thursday are grouped together as they show similar behaviours.

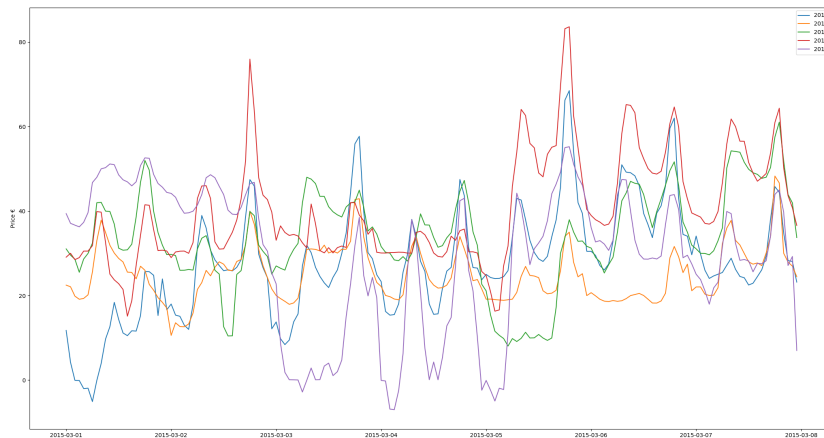


Figure A.13: Price levels for March for each year in the dataset. As seen the patterns looks similar for each day, but the price levels differs.

Appendix B

Detailed Results

In this Appendix, detailed MSE and MAE results per month is provided. This can be interesting to look into as certain months vary more between years than others, e.g. March weather and thus electricity prices depends strongly on when Spring arrives. Detailed results per month, January:

Model	1a	1b	1c	2a	2b	2c
MSE	0.051	0.038	0.040	0.050	0.038	0.039
MAE	0.144	0.138	0.131	0.145	0.139	0.133

Model	3a	3b	3c	4a	4b	4c
MSE	0.043	0.049	0.047	0.043	0.048	0.046
MAE	0.165	0.183	0.178	0.166	0.182	0.177

Table B.1: Detailed January results.

Detailed results per month, February:

Model	1a	1b	1c	2a	2b	2c
MSE	0.032	0.032	0.033	0.032	0.032	0.033
MAE	0.136	0.139	0.139	0.136	0.139	0.139

Model	3a	3b	3c	4a	4b	4c
MSE	0.035	0.034	0.034	0.035	0.034	0.034
MAE	0.139	0.139	0.139	0.139	0.139	0.139

Table B.2: Detailed February results.

Detailed results per month, March:

Model	1a	1b	1c	2a	2b	2c
MSE	0.149	0.140	0.141	0.149	0.141	0.141
MAE	0.349	0.339	0.339	0.350	0.339	0.339
Model	3a	3b	3c	4a	4b	4c
MSE	0.140	0.131	0.134	0.141	0.132	0.135
MAE	0.330	0.318	0.322	0.331	0.319	0.323

Table B.3: Detailed March results.

Detailed results per month, April:

Model	1a	1b	1c	2a	2b	2c
MSE	0.088	0.110	0.087	0.088	0.109	0.087
MAE	0.203	0.254	0.199	0.203	0.253	0.200
Model	3a	3b	3c	4a	4b	4c
MSE	0.108	0.094	0.097	0.106	0.093	0.096
MAE	0.253	0.222	0.228	0.250	0.221	0.227

Table B.4: Detailed April results.

Detailed results per month, May:

Model	1a	1b	1c	2a	2b	2c
MSE	0.244	0.155	0.201	0.147	0.121	0.129
MAE	0.474	0.368	0.426	0.353	0.316	0.326
Model	3a	3b	3c	4a	4b	4c
MSE	0.048	0.042	0.043	0.039	0.039	0.040
MAE	0.164	0.154	0.156	0.148	0.149	0.150

Table B.5: Detailed May results.

Detailed results per month, June:

Model	1a	1b	1c	2a	2b	2c
MSE	0.209	0.208	0.207	0.204	0.202	0.201
MAE	0.285	0.273	0.274	0.284	0.272	0.273

Model	3a	3b	3c	4a	4b	4c
MSE	0.197	0.195	0.196	0.193	0.191	0.191
MAE	0.284	0.271	0.276	0.283	0.270	0.275

Table B.6: Detailed June results.

Detailed results per month, July:

Model	1a	1b	1c	2a	2b	2c
MSE	0.025	0.020	0.021	0.025	0.020	0.021
MAE	0.117	0.102	0.107	0.117	0.102	0.107

Model	3a	3b	3c	4a	4b	4c
MSE	0.023	0.022	0.023	0.023	0.022	0.023
MAE	0.109	0.106	0.108	0.109	0.106	0.108

Table B.7: Detailed July results.

Detailed results per month, August:

Model	1a	1b	1c	2a	2b	2c
MSE	0.039	0.038	0.039	0.039	0.038	0.039
MAE	0.136	0.133	0.136	0.136	0.133	0.136

Model	3a	3b	3c	4a	4b	4c
MSE	0.041	0.036	0.038	0.041	0.036	0.038
MAE	0.144	0.131	0.136	0.144	0.131	0.136

Table B.8: Detailed August results.

Detailed results per month, September:

Model	1a	1b	1c	2a	2b	2c
MSE	0.066	0.054	0.059	0.066	0.054	0.059
MAE	0.231	0.201	0.214	0.231	0.201	0.214
Model	3a	3b	3c	4a	4b	4c
MSE	0.053	0.048	0.050	0.053	0.048	0.050
MAE	0.202	0.188	0.194	0.202	0.188	0.194

Table B.9: Detailed September results.

Detailed results per month, October:

Model	1a	1b	1c	2a	2b	2c
MSE	0.088	0.074	0.076	0.075	0.066	0.066
MAE	0.260	0.224	0.231	0.235	0.210	0.213
Model	3a	3b	3c	4a	4b	4c
MSE	0.076	0.069	0.073	0.070	0.065	0.067
MAE	0.223	0.205	0.213	0.213	0.198	0.203

Table B.10: Detailed October results.

Detailed results per month, November:

Model	1a	1b	1c	2a	2b	2c
MSE	0.016	0.015	0.015	0.015	0.014	0.015
MAE	0.100	0.098	0.099	0.099	0.097	0.098
Model	3a	3b	3c	4a	4b	4c
MSE	0.018	0.017	0.018	0.017	0.017	0.017
MAE	0.107	0.107	0.107	0.107	0.107	0.107

Table B.11: Detailed November results.

Detailed results per month, December:

Model	1a	1b	1c	2a	2b	2c
MSE	0.124	0.114	0.118	0.124	0.114	0.118
MAE	0.289	0.272	0.278	0.289	0.272	0.279
Model	3a	3b	3c	4a	4b	4c
MSE	0.128	0.117	0.122	0.128	0.117	0.122
MAE	0.285	0.270	0.277	0.286	0.270	0.278

Table B.12: Detailed November results.

Master's Theses in Mathematical Sciences 2021:E57
ISSN 1404-6342
LUNFMS-3102-2021
Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden
<http://www.maths.lu.se/>