



Technical University  
of Denmark

---

# OFFSHORE INTERMITTENCY RISK

Recent methods in the cross-section of time series,  
machine learning, and dynamical systems

Nicolaj Hans Nielsen, s184335

BSc Mathematics and Technology

---

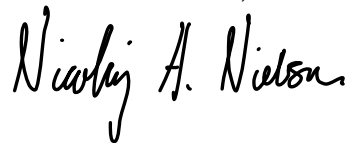
Supervisor: Pierre Pinson, Professor

December 23, 2021

## Preface

This B.Sc. thesis was prepared at the Technical University of Denmark, in partial fulfillment of the requirements for acquiring a B.Sc degree in Mathematics and Technology. Professor Pierre Pinson supervised this project, and I am thankful for the involvement in the project, the excellent guidance, and the interesting discussions. I would also like to thank *Ørsted A/S* for providing the data for this project.

Bachelor Thesis, Autumn 2021

A handwritten signature in black ink, reading "Nicolaj A. Nielsen". The script is cursive and fluid, with the first name "Nicolaj" and last name "Nielsen" clearly distinguishable.

Nicolaj Hans Nielsen

Initials: s184335

Mail: s184335@student.dtu.dk

## Abstract

Renewable power generators are subjected to increasing price risk due to the intermittency of their power production along with the *merit order effect* on the market. This means that their production weighted prices are often far from the average market prices. The difference between the two is the *intermittency risk*. It is of interest as many contracts are based on simple daily averages hence predictions of this risk factor are essential for efficient hedging strategies. We focus on the daily *offshore intermittency risk* three days ahead using the N2EX spot prices and the predicted wind power production in the UK with data from *Ørsted*. In our approach, we construct a model for the price of each hour using an ARIMA model, local linear models, LightGBM, and two baseline models. The LightGBM has superior performance with an MAE of 0.392 GBP/MWh, a dramatic improvement from the baseline of 0.692 GBP/MWh. Using quantile regression, we issue a probabilistic forecast for each hour and account for the interdependence between hours using a Gaussian copula. Using LightGBM, we obtain a CRPS score of 0.306 GBP/MWh on the validation data. In the test data from 2021, we see unprecedented volatilities and obtain an MAE of 1.29 GBP/MWh and a CRPS of 0.895 GBP/MWh. To improve the model, we suggest a new update mechanism to adapt to market shifts and propose a set of new modeling strategies that could be used to improve the probabilistic forecasts.

# Contents

<b>Introduction</b>	<b>5</b>
<b>1 Energy Market</b>	<b>6</b>
1.0.1 Volume risk . . . . .	6
1.0.2 Price risk . . . . .	6
1.0.3 Forward contracts . . . . .	6
1.0.4 Intermittency Risk . . . . .	8
<b>2 Forecasting and modeling</b>	<b>9</b>
2.1 Point Forecasts . . . . .	9
2.2 Probabilistic Forecasts . . . . .	10
2.2.1 Non-parametric Quantile Regression . . . . .	10
2.2.2 Baseline Quantile Forecast . . . . .	11
2.2.3 Local Linear Model . . . . .	11
2.2.4 A Generic Local Polynomial Model . . . . .	12
2.2.5 Gradient Boosting . . . . .	13
2.2.6 Forward Stage Additive Modeling . . . . .	14
2.2.7 Gradient boosting . . . . .	14
2.2.8 Gradient Tree Boosting . . . . .	15
2.3 Evaluation of Probabilistic Forecasts . . . . .	16
2.3.1 PIT diagrams . . . . .	17
2.3.2 CRPS Scores . . . . .	17
<b>3 Machine Learning Strategies for Time Series Forecasting</b>	<b>18</b>
3.1 A Generic Supervised Learning Problem . . . . .	18
3.1.1 Intuition from Time Series Models . . . . .	18
3.2 Interpretation of Lag Value Proposition . . . . .	18
3.3 Strategies for Multi-Step Time Series Forecasting . . . . .	19
<b>4 Generation of Statistical Scenarios</b>	<b>20</b>
4.1 From Predictive Distribution to a Multivariate Gaussian . . . . .	20
4.2 From Multivariate Gaussian to Statistical Scenarios . . . . .	20
<b>5 Methodology</b>	<b>22</b>
5.1 Split labeled data into <b>training</b> , <b>validation</b> , and <b>test</b> sets. . . . .	22
5.1.1 Time Based Cross-Validation . . . . .	23
5.1.2 Preprocessing . . . . .	23
5.1.3 Choose some ML algorithm . . . . .	23
5.1.4 Train ML model with various hyperparameter settings . . . . .	23

5.1.5	Evaluate prediction functions on validation set . . . . .	24
<b>6</b>	<b>Results</b>	<b>25</b>
6.1	Price Forecasting . . . . .	25
6.2	Probabilistic calibration . . . . .	26
6.3	Generation of Statistical Scenarios . . . . .	27
6.4	Intermittency Risk . . . . .	28
6.5	Evaluation on Test Data . . . . .	29
<b>7</b>	<b>Discussion</b>	<b>30</b>
7.1	Adaptivity . . . . .	30
7.2	Weak Learners and the Strength of an Ensemble . . . . .	30
7.3	Uncertainty of Predictions . . . . .	30
7.4	Feature Importance . . . . .	31
7.5	New Strategies for Prediction of the Intermittency Risk . . . . .	31
7.6	Hybrid Models . . . . .	31
<b>8</b>	<b>Conclusion</b>	<b>32</b>
	<b>References</b>	<b>33</b>
	<b>Appendices</b>	<b>36</b>
A	UK Power Market . . . . .	36
B	Proof: Minimization of Pinball Loss Leads to Best Estimate of Quantile	37
C	Created Features . . . . .	38
D	Important Hyperparameters for LightGBM . . . . .	39
E	The Available Input Data . . . . .	40
F	ZCA-cor and the Implications for Feature Importance . . . . .	41
G	Time-base cross validation . . . . .	42
H	Supplementary Figures to Assess Model Performance on Validation Data . . . . .	42
I	Changes in the PIT After a Redefinition $\hat{q}^{(0)}$ and $\hat{q}^{(1)}$ . . . . .	44
J	Supplementary Figures to Assess Model Performance on Test Data .	44

## Introduction

Renewable energy generators in the UK have relied heavily on subsidies and feed-in tariffs to ensure that projects would materialize. Now that these initiatives start to phase out [1], the generators are more exposed to the volatile energy market. Due to the *intermittency* of renewable power generation and the *merit order effect*, the generation weighted prices are often lower than average prices. The difference between the average price and the generation weighted prices is the *intermittency risk* and it is a crucial quantity for a renewable generator to hedge against price risks in the forward market.

We issue forecasts for the three days ahead daily *intermittency risk* and use the difference between the average daily *N2EX* day-ahead price and the weighted average using national wind power predictions provided by *Ørsted*. As we have the predicted wind power generation, our approach is to predict the *N2EX* prices for each hour and weight the prediction by the predicted power penetration. We first make a seasonal ARIMA model for each hour and compare it with a baseline. To increase the amount of valuable information for decision-making, we issue probabilistic forecasts using quantile regression. For each hour we fit a *local linear model*, a gradient tree boosting model using *LightGBM* [2], and compare with a probabilistic baseline model. We model the interdependency between the probabilistic forecasts using a Gaussian copula. This also allows us to generate price scenarios for each hour that we can directly weigh by the predicted wind power. We can then generate a probabilistic forecast for the daily intermittency risk using the scenarios.

To imitate the environment of the model at deploy time, we will issue all forecasts as rolling forecasts and select the model that has the best performance on our validation data using MAE and CRPS. Using the best model, we will predict the intermittency on the test set and assess the performance. We discover how volatile and non-stationary the market is, especially under recent events as COVID-19 and a global gas crisis. It makes an interesting challenge with many possible paths of which we have only uncovered a handful and can suggest a couple more.

# 1 | Energy Market

Wholesale electricity in the UK is traded in multiple ways to include small and large market participants, stimulate the installment of more renewable energy generation, and ensure stability on the grid [3]. Most of the trade volume is facilitated via bilateral contracts between the retailers and generators directly; however, an increasing amount of energy is also traded in the day-ahead markets [4]. With current technologies, electricity is not storable or prohibitively expensive to store, which means demand and supply will have to be in balance to ensure the stability of the power system. The transmission system operator, TSO, ensures stability and knows about all grid interactions. If a market participant cannot meet their traded volumes, then the TSO runs auxiliary markets where volumes can be finetuned in real-time. With this market construction, we can locate two fundamental risk factors for market participants; *volume risk* and *price risk*.

## 1.0.1 Volume risk

On the retailer side, the volume risk is the uncertainty in consumer demand. Demand can be hard to predict as it will change with intra-day business and everyday activities, with weekend activity, and with annual temperature shifts. The volume risk for the generators stems from uncertainty in the volumes they can generate. There could be an outage, or there will be a global shortage of gas or fuel. For renewable energy generators, the volume risk is huge because their generation depends hugely on meteorological variables that are highly stochastic, which makes the production *intermittent* because the energy is not available at all time [5].

## 1.0.2 Price risk

The volume risk is highly linked to the price risk as demand and supply affects the realized prices. The electricity market is highly volatile and has large price movements, which is undesirable to most parties. The generators will not know the price they will obtain for the generated electricity, and the consumers cannot be sure about their production cost or cost of living. In figure 1.1a, we see how the distribution of each hour differs with the number of outliers. Therefore, price forecasts are essential for power portfolio managers to construct sustainable trading strategies. A viable way to hedge price risk on the day-ahead market is to make forward contracts.

## 1.0.3 Forward contracts

Forward contracts are bilateral contractual agreements and are of enormous interest and make up the majority of the traded volumes in the UK, see appendix A. The trades are over the counter, OTC, which means they most often happen outside the order book of an exchange.

## Long-term agreements

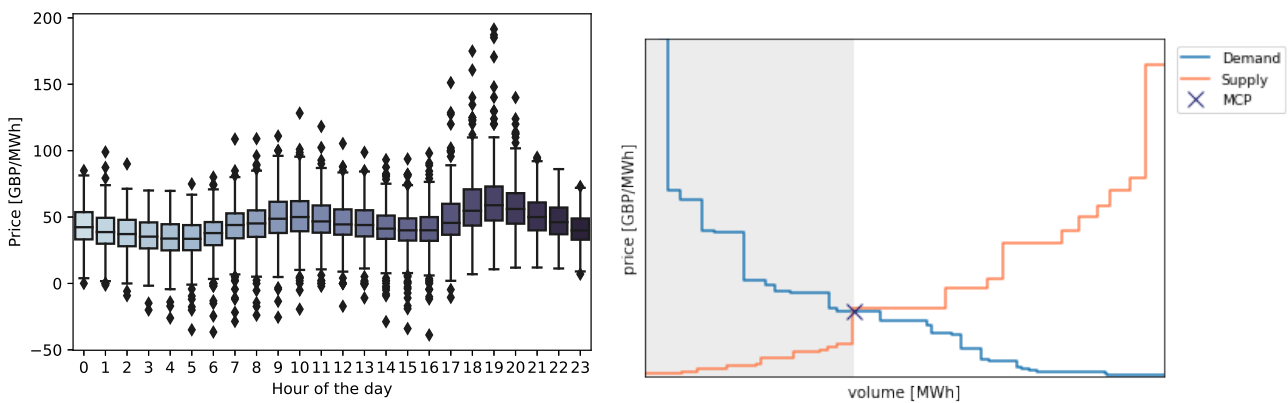
We focus on renewable energy suppliers. For a long time, they could rely on fiscal incentives, which in the UK would be feed-in tariffs, FiTs, and Renewable Obligations, ROCs. It meant that renewable energy suppliers could enter short-term agreements as the support payments were sufficient to demonstrate a stable revenue stream to investors and lenders. As these initiatives have started to phase out, there has been a considerable rise in corporate power purchase agreements, CPPAs, which are long-term bilateral agreements often with a large credit-worthy corporate [1]. They are essential for wind projects to materialize; however, the transaction costs are high, and the tendering process can be lengthy, which might not be beneficial for short time horizons and finetuning of contracted energy volumes.

## Standardized OCT

The standardized forward contracts offer shorter time horizons, trades at lower volumes, and lower transaction costs, which can be very beneficial. The contracts are traded on e.g. the European Energy Exchange, EEX. Because many CPPAs are directly linked to the day-ahead market through simple daily averages, these standardized products can be used to hedge price risk.

## Day-ahead Market

Participants can trade energy for each hour of the following day in a day-ahead market. We focus on the UK day-ahead market *N2EX* exchange operated Nord Pool Spot and Nasdaq OMX Commodities. Here the agents must submit their bids before 12:00 CET (as of 1. January, it is changed to 10:50 CET [6]) for each hour of the following day. Participants can submit multiple bids with different prices and volumes with boundary price thresholds at -500 and 3000 GBP/MWh [7]. The role of the power exchange is to determine the *market clearing price*, MCP. In a broad sense, it is determined as the intersection between the ordered, aggregated supply and demand curves for each hour [8], see figure 1.1b.



(a) Hourly distribution of the N2EX price from 2018-01-01 to 2020-10-01

(b) Depiction of the clearing process. The shaded bids will all obtain the MCP.

**Figure 1.1** – An constructed illustration of the way to determine the market clearing figure in 1.1b. In figure 1.1a, we see the hourly distribution of N2EX prices. Note how the distribution changes for each hour.

The N2EX is a marginal auction market where all sellers with bids below or equal to the MCP and all buyers with bids above the MPC will obtain the same price. This way



of determining the MCP using the *merit order* of the bids gives an advantage to the renewable energy generators. Given the correct weather conditions, the marginal cost of one unit more of energy is essentially zero. On the other hand, for a gas plant, the marginal cost is far from zero as the gas plant would have to burn the corresponding unit of gas. In markets that use the *merit order* principle, it means that coal and oil plants could be pushed out of the energy mix if the weather conditions are correct and the prices would be much lower. This effect of renewable energy sources on marginal auction-based markets is called the *merit order* effect [5]. This is beneficial for renewable energy generators as they would always be in the power mix and obtain the MCP. The downside is that the renewable energy generators will get a lower generation weighted price than the average price because they would push the oil and gas plants out of the energy mix in hours with a lot of wind. This is a great risk as a great deal of PPAs and subsidies are based on simple daily averages [1].

#### 1.0.4 Intermittency Risk

We define the daily *intermittency* risk three days ahead as

$$\mathcal{I}_d = \frac{\sum_{h=0}^{23} p_{d,h} w_{d,h}}{\sum_{h=0}^{23} w_{d,h}} - \frac{\sum_{h=0}^{23} p_{d,h}}{24} \quad (1.1)$$

Where  $p_{d,h}$  is the realized power price at the N2EX at day  $d$  and hour  $h$  and  $w_{d,h}$  is the off-shore wind power forecast with a lead-time of three days. The wind power forecast on a country level in the UK and are provided by Ørsted<sup>1</sup>.

---

<sup>1</sup>Estimating this specific intermittency risk was raised during an internal challenge, *HackØrthon*, at Ørsted created by Edoardo Simioni. *Head of Short Term Trading Optimization*, Ørsted

## 2 | Forecasting and Modeling

The intermittency risk with a lead time of 3 days,  $\mathcal{I}_{d+3}$ , can be predicted using different strategies. From [8] and figure 1.1a we recognize that the price dynamics differ for each hour and it would be hard to capture the underlying price dynamics for each hour with one model. To reduce the model complexity and to be able to investigate and gain insights about the model differences between hours, we make a model for each hour to predict the prices three days ahead. We can then use the predicted wind power to obtain the intermittency risk as defined in 1.1. In the following, we will focus on the prediction of day-ahead power prices. We let  $P_t$  denote the stochastic process of the hourly prices and let  $p_{t+k}$  be the realization of  $P_{t+k}$  where  $k$  is the hours from time  $t$ . Using the price history, we will issue point forecasts. A point forecast is just one summary statistics for  $P_{t+k}$ . It would be useful to know more about the conditional distribution in a volatile energy market to make more informed decisions. In section 2.2, we will focus on non-parametric quantile forecasts using local linear models, gradient tree boosting with *LightGBM* [2], and compare with a constructed baseline model.

### 2.1 Point Forecasts

In the following, we introduce a baseline model and a SARIMA model for point forecasting the energy prices.

#### Benchmark model

A common baseline for price prediction is the *similar-day* model [8]. We will combine this with the idea of a seasonal naive method [9] to construct a baseline model. If we are to predict a Monday, then we will take the average of the prices of the last 3 Mondays. For ease of notation let  $p_{d,h}$  be the price at day  $d$  and hour  $h$ . Introduce  $\mathbf{p}_{d+3,\cdot}$  as the prices 3 days ahead,  $\mathbf{p}_{d+3,\cdot} = [p_{d+3,0}, p_{d+3,1}, \dots, p_{d+3,23}]^\top \in \mathbb{R}^{24}$ . Then we want to predict  $\mathbf{p}_{d+3,\cdot}$  given data to day  $d$ . For the prediction of a specific hour  $h$ , we use the last 3 similar week days such that

$$\hat{p}_{d+3,h|d} = \frac{p_{d-4,h} + p_{d-(4+7),h} + p_{d-(4+14),h}}{3} \quad (2.1)$$

#### Seasonal ARIMA Model

We predict the prices three days ahead with a model for each hour. We choose to do this with a seasonal ARIMA model [9] to avoid the accumulation of errors that would happen if we predicted three days ahead with a standard one-step ARIMA model. Consider the model  $\text{ARIMA}[P, D, Q]_m$  where  $m$  is the number of seasonal lags,  $P$  is the order of the seasonal AR part,  $d$  is the degree of difference, and  $Q$  is the number of seasonal moving average terms. Using only the available information, we restrict the models to be those with

$m \geq 3$ . Using the ACF and PACF, we can determine the number of needed seasonal terms and if the order of difference is correct using the procedures in [10]. One assumption for ARIMA models is that the series is weakly stationary [10]; however, this is not the case for our series. Therefore, we will try with log-transformation, a box-cox transformation, reduce the number of data points and adaptively update the parameters. The best SARIMA fit will be the one that minimizes the square loss. It can be shown that this corresponds to finding the conditional mean [10].

### Evaluation of Point forecast

We use the mean absolute error, MAE, to evaluate the performance of the point forecasts because this is the metric of interest from *Ørsted*:

$$\text{MAE} = \frac{1}{N_d} \sum_{d=1}^{N_d} |\mathcal{I}_d - \hat{\mathcal{I}}_d| \quad (2.2)$$

where  $N_d$  is the number of days in the period we evaluate.

## 2.2 Probabilistic Forecasts

In the following, we briefly cover the core elements of a non-parametric quantile regression [11]. Then we introduce a baseline model, local linear models, and then a gradient tree boosting model using *LightGBM*.

### 2.2.1 Non-parametric Quantile Regression

Introduce the cumulative distribution function  $F_{t+k}$  for  $P_{t+k}$  and let  $q_{t+k}^{(\alpha_i)}$  be the quantile with nominal level  $\alpha \in [0, 1]$ . We define  $q_{t+k}^{(\alpha_i)}$  of  $P_{t+k}$  to be the unique value  $x$  such that

$$\mathbb{P}(P_{t+k} < x) = \alpha_i \quad \text{or equivalently} \quad q_{t+k}^{(\alpha_i)} = \inf \{x : F_{t+k}(x) \geq \alpha_i\} = F_{t+k}^{-1}(\alpha_i) \quad (2.3)$$

where  $F_{t+k}^{-1}(\alpha)$  is the inverse cumulative function.

Given all information up to time  $t$ , denoted  $\Omega_t$ , we can make a conditional estimate of the quantile  $\hat{q}_{t+k}^{(\alpha_i)}$ . If we assume nothing about the target distribution, then we can predict each  $\hat{q}_{t+k}^{(\alpha_i)}$  using non-parametric methods. This enables us to make a point-wise estimate of conditional distribution, i.e. the density  $\hat{f}_{t+k|t}$  of this distribution can be characterized using a finite set of  $m$  quantiles.

$$\hat{f}_{t+k|t} = \left\{ \hat{q}_{t+k}^{(\alpha_i)} \mid 0 \leq \alpha_1 < \alpha_2 < \dots < \alpha_m \leq 1 \right\} \quad (2.4)$$

we use  $m = 19$  with nominal levels equidistantly on the unit interval such that  $\alpha_i \in \{0.05, 0.10, \dots, 0.90, 0.95\}$ .

### Estimation of Quantiles Using Pinball-loss Function

An estimate of the quantile  $\hat{q}_{t+k|t}^{(\alpha)}$  can be obtained by minimizing the  $\alpha$ -pinball-loss [11]. For some estimate of the quantile  $x \in \mathbb{R}$  and a generic response variable  $y \in \mathbb{R}$ , we can define  $\ell_\alpha : \mathbb{R} \times \mathbb{R} \mapsto [0, \infty)$  as:

$$\ell_\alpha(y, x) = \begin{cases} (y - x)(\alpha - 1) & y < x \\ (y - x)\alpha & y \geq x \end{cases} \quad (2.5)$$

In appendix B, we show that if we minimize the expectation of  $\ell_\alpha$ , then we find a best estimate of the quantile with nominal level  $\alpha$ :

$$q_{t+k|t}^{(\alpha)} = \arg \min_{\hat{q}_{t+k|t}^{(\alpha)}} \mathbb{E}_p \left[ \ell_\alpha \left( P_{t+k}, \hat{q}_{t+k|t}^{(\alpha)} \right) \right] \quad (2.6)$$

We use  $\ell_\alpha$  for each nominal value of  $\alpha$  to construct the ensemble of quantiles. When we estimate the ensemble, then  $\hat{q}_{t+k}^{(\alpha_i)}$  should ideally be monotonically increasing with the nominal level. In many cases, this will not hold for empirically estimated quantiles. Therefore, we order the predicted quantiles each time to enforce this condition. It is shown in [12] that the rearranged quantiles are closer to the actual quantiles in finite samples; hence we rearrange all our quantile forecasts.

### 2.2.2 Baseline Quantile Forecast

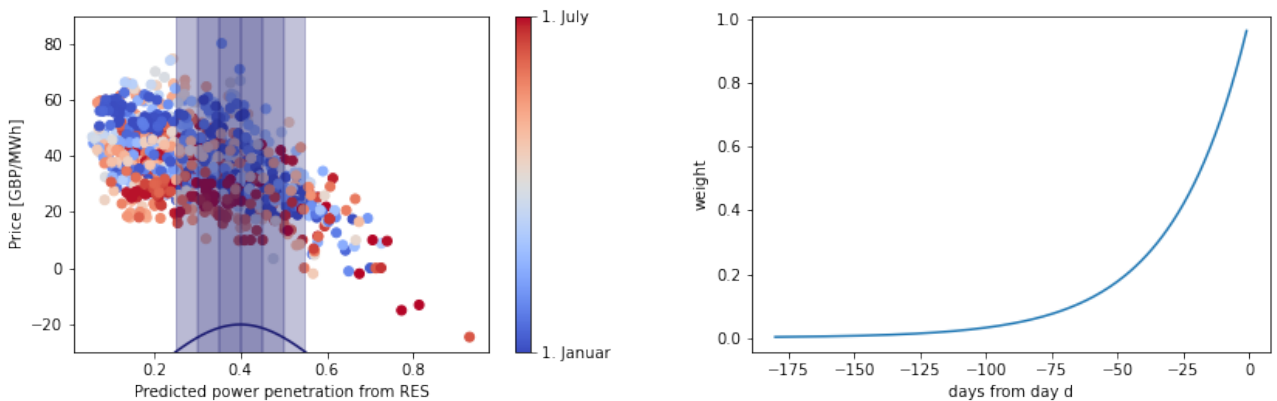
As the energy prices are highly non-stationary, it would be appropriate to make baseline model that is also local and not based on advanced quantile regression. Let be  $p_{d,h}$  the realized price for at specific hour  $h$  at day  $d$  and introduce  $\mathbf{p}_{d,h}^{(\Psi)} \in \mathbb{R}^{20}$  as the vector of the last 20 observation of that hour,  $\mathbf{p}_{d,h}^{(\Psi)} = [p_{d,h}, p_{d-1,h}, \dots, p_{d-19,h}]^\top$ . Now sort  $\mathbf{p}_{d,h}^{(\Psi)}$  to obtain  $\mathbf{p}_{d,h}^{(s)} = [p_{s,h}^{(1)}, p_{s,h}^{(2)}, \dots, p_{s,h}^{(20)}]^\top$  such that  $p_{s,h}^{(i)} \leq p_{s,h}^{(i+1)}$  for  $i = 1, 2, \dots, 19$ . The local baseline forecast for each quantile will simply be the average of consecutive values in  $\mathbf{p}_{d,h}^{(s)}$ :

$$\hat{q}_{t+3|t}^{0.05} = \frac{p_{s,h}^{(1)} + p_{s,h}^{(2)}}{2}, \quad \hat{q}_{t+3|t}^{0.1} = \frac{p_{s,h}^{(2)} + p_{s,h}^{(3)}}{2}, \quad \dots, \quad \hat{q}_{t+3|t}^{0.95} = \frac{p_{s,h}^{(19)} + p_{s,h}^{(20)}}{2} \quad (2.7)$$

In this way, we predict the quantiles based on the immediate price history.

### 2.2.3 Local Linear Model

Due to the merit order effect, the predicted renewable energy penetration is a good predictor of energy prices. The relationship is complex [5] hence a linear regression is not appropriate, and therefore, we use a local linear regression. The idea is to make a linear regression that weighs past data points with a similar predicted renewable power penetration using a Gaussian kernel. Consider a day with predicted power penetration of 0.4, then we would weigh data around 0.4 as depicted in figure 2.1a.



(a) Predicted power penetration and price for hour 6 from 2019-10-01 to 2020-10-01.

(b) Used exponential decay to weight immediate past. In this case, the half time is 20 days,  $d_{1/2} = 20$ .

**Figure 2.1** – In figure 2.1a, we see the Gaussian kernel for a fitting point at 0.4. We also see a need to account for seasonal effects. Figure 2.1b shows the proposed exponential weights.

In figure 2.1a, we see the apparent relationship, but the prices seem to be at a higher level during the winter. To try to account for this, we will weigh the immediate past using an exponential decay as in figure 2.1b.

In the following, we will cover how we can define the local linear problem, introduce the Gaussian kernel, and the exponential weightings. In appendix C we show the features we use to define the predicted renewable energy penetration.

### 2.2.4 A Generic Local Polynomial Model

Consider a generic problem with  $N$  realizations of pairs  $\{(X, Y)\}_{i=1}^N$  where  $x_i$  is the predictor variable of  $y_i$ . We want to find a function  $f : \mathbb{R} \mapsto \mathbb{R}$  that minimizes the square loss

$$f^* = \arg \min_f \sum_{i=1}^N (y_i - f(x_i))^2 \quad (2.8)$$

Let  $g$  be a local parametrization of  $f$  around each point  $x_i$ . Consider a Taylor approximation of  $g$  around a specific data point  $x_i$ :

$$g(x_i) \approx g(x) + g'(x)(x_i - x) + \frac{g''(x)}{2}(x_i - x)^2 + \dots + \frac{g^{(p)}(x)}{p!}(x_i - x)^p \quad (2.9)$$

with some finite  $p$ . Substitute the approximation of 2.9 into equation 2.8:

$$g^* = \arg \min_g \sum_{i=1}^n \left( Y_i - \sum_{j=0}^p \frac{g^{(j)}(x)}{j!} (x_i - x)^j \right)^2 \quad (2.10)$$

The trick is now to reformulate this as a linear regression problem. Here stipulate  $\theta_i := \frac{g^{(j)}(x)}{j!}$  for all  $j$  such that we have a vector  $\theta = [\theta_0, \theta_1, \dots, \theta_{p+1}]^T \in \mathbb{R}^{p+1}$ . Then equation 2.10 simple is a least square problem

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^n \left( Y_i - \sum_{j=0}^p \theta_j (x_i - x)^j \right)^2 \quad (2.11)$$

In our specific case, we use a linear model hence we set  $p = 1$ . Consider some new input  $x$ , we want  $\theta^*$  to be local hence we, we weight each datum  $(x_i, y_i)$  according to the proximity of  $x_i$  to  $x$  as in figure 2.1a.

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^n \left( Y_i - \sum_{j=0}^p \theta_j (x_i - x)^j \right)^2 \mathbf{w}_x \quad (2.12)$$

where  $\mathbf{w}_x \in \mathbb{R}_{>0}^n$  is the weight for each point  $i = 1, 2, \dots, n$  of our data. We decompose  $\mathbf{w}_x$  into assigned weight from the Gaussian kernel and the exponential weights  $\mathbf{w}_x = \mathbf{w}_k \mathbf{w}_e$  and study them separately in the following.

### Gaussian Kernel

The Gaussian kernel is a way to estimate the conditional density of the fitting point  $x$  and in this case we use it as a proximity measure. Let  $w_{k,i}$  be the weight assigned to a point  $i$  when we have the fitting point  $x$ :

$$w_{k,i} = K_h \left( \frac{x - x_i}{h} \right) \quad (2.13)$$

where  $h \in \mathbb{R}_{>0}$  is the bandwidth [13] that controls the size of the neighborhood around  $x$ . The  $K_h : \mathbb{R} \mapsto \mathbb{R}_{>0}$  is the kernel that could have an arbitrary density but is mostly symmetric and unimodal around zero [13]. We use a Gaussian kernel which is the PDF of a standard Gaussian:

$$K(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \quad (2.14)$$

to find the optimal bandwidth  $h^*$ , we use *leave-one-out* cross validation.

### Exponential decay

For the  $w_e$ , we use an exponential decaying function as the one in figure 2.1b. We use the generic formula  $N(d) = \exp -\lambda d$  where  $\lambda$  is the exponential decay constant. Introduce  $\mathbf{d}_{away} = [N, (N-1), \dots, 1, 0]^T \in \mathbb{Z}^{N+1}$  as the vector days each data pair is from day  $d$ . Then we can control the number of time points to include using the half time,  $d_{1/2}$  and define the vector of exponential weights  $\mathbf{w}_e$ :

$$d_{1/2} = \frac{\ln 2}{\lambda} \quad \mathbf{w}_e = \exp\left(-\mathbf{d}_{away} \frac{\ln 2}{d_{1/2}}\right) \quad (2.15)$$

We will allow  $d_{1/2} \in [20, 180]$  and use cross validation to find the best  $d_{1/2}$ . In figure 2.1a, we also see a clear seasonal pattern hence we introduce an indicator function to indicate if the point is in the same season as day  $d$ :

$$\mathbf{1}_{sea} = \begin{cases} 1 & \text{if } x_{d_i} \in \{x_{d-1}, x_{d-2}, \dots, x_{d-183}, x_{d-365}, x_{d-366}, \dots\} \\ 0 & \text{otherwise} \end{cases} \quad (2.16)$$

Then we define the vector of seasonal weights as where  $w_{sea} = w_e \mathbf{1}_{sea}$ . We call this a local linear model with seasonal weights.

### 2.2.5 Gradient Boosting

Boosting is a learning method where simple basis functions are added iteratively to an ensemble of basis functions such that the added basis function minimizes the error of the ensemble so far [14].

**Sequential Ensemble Method** Consider some chosen basis functions  $h_1, h_2, \dots, h_M$  with  $h_i : \mathbf{X} \mapsto \mathbb{R}$ , then we can construct an ensemble as a linear combination of the basis functions

$$f_M(x) = \sum_{m=1}^M v_m h_m(x) \quad (2.17)$$

where  $v_m \in \mathbb{R}_{>0}$  and  $v_m, h_m$  are found using in-sample data. In short, we define our ensemble as:

$$\mathcal{F}_M = \left\{ \sum_{m=1}^M v_m h_m(x) \mid v_m \in \mathbb{R}_{>0}, h_m \in \mathcal{H}, m = 1, \dots, M \right\} \quad (2.18)$$

Where  $\mathcal{H}$  is the space of basis function we consider which could be e.g. linear functions or trees ([15], lecture 7).

### 2.2.6 Forward Stage Additive Modeling

Consider a generic dataset,  $\mathcal{D} = \{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \dots, (\mathbf{X}_n, y_n)\}$ , and some loss function  $\ell$ . We initiate the ensemble with  $f_0(x) = 0$  and after stage  $m - 1$  the model is  $f_{m-1} = \sum_{i=1}^{m-1} v_i h_i$ . In the next round, we find some basis function  $h_m$  and weight  $v_m$  and expand the ensemble  $f_m = f_{m-1} + v_m h_m$ . At iterations we find  $(v_m, h_m)$  such that

$$(v_m, h_m) = \arg \min_{v \in \mathbf{R}, h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell \left( y_i, f_{m-1}(\mathbf{x}_i) + \underbrace{v h(\mathbf{x}_i)}_{\text{new piece}} \right) \quad (2.19)$$

We can now repeat this process until the error is within an acceptable level or until some predefined maximum number of iterations. We now see that  $v$  can be understood as a learning rate for each iteration. Algorithm 10.2 and section 10.2 in [14] covers the process in great detail. This algorithm works for simple loss functions [14]; however, it is unable to generalize to a lot of loss functions. In addition, the slightly more advanced gradient boosting algorithm has superior performance [16].

### 2.2.7 Gradient boosting

The additive modeling process can be extended to fit all problems with a differential loss function by analogy to numerical optimization [14]. Let  $f_m : M_{N \times p}(\mathbb{R}) \mapsto \mathbb{R}$  denote the current ensemble of  $m$  basis functions and let  $\ell$  be a differentiable loss function. The objective of gradient boosting [17] is to minimize

$$J(f) = \sum_{i=1}^n \ell(y_i, f_m(\mathbf{x}_i)) \quad (2.20)$$

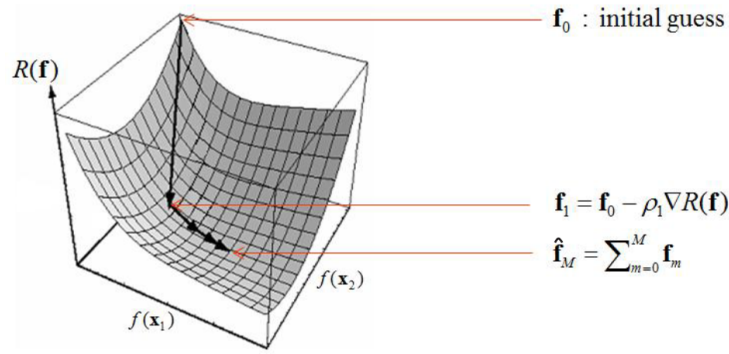
we want to minimize  $J$  with respect to  $f_m$  evaluated on the training points. Therefore introduce  $\mathbf{f}_m = [f_m^1, f_m^2, \dots, f_m^n]^\top \in \mathbb{R}^n$  as the current ensemble evaluated on each of the  $n$  training points and now consider doing gradient descent. At each iteration, we take a step in the negative gradient step direction. Let  $-\mathbf{g} \in \mathbb{R}^n$  be the negative gradient for each data point

$$-\mathbf{g} = -\nabla_{f_m} J(f_m) = - \left[ \frac{\partial \ell(y, f)}{\partial f} \right] \bigg|_{y=y_i, f=f_m^i} \quad \text{for all } i = 1, 2, \dots, n \quad (2.21)$$

In the particular case where  $\ell$  is the square loss function  $\ell_2$  then  $\frac{\partial \ell_2}{\partial f} = \frac{(y-f)^2}{\partial f} = 2(y-f)$  and then plug in  $y_i$  and  $f_m(x_i)$  such that  $-\mathbf{g}_i = 2(y_i - f_m(x_i))$  for all  $i = 1, 2, \dots, n$ . With this gradient at each point, we now find the basis function  $h_m$  that is the closest approximation to the negative gradient.

$$h_m = \min_{h \in \mathcal{H}} \sum_{i=1}^n (g_i - h(\mathbf{x}_i))^2 \quad (2.22)$$

A graphical representation of this procedure is presented figure 2.2, adopted from [16]. Here they consider the predictions on the dataset  $\mathbf{f} = (f(\mathbf{x}_1), f(\mathbf{x}_2))$ , they use  $R(\mathbf{f})$  for their objective function, and  $\rho$  for the learning rate.



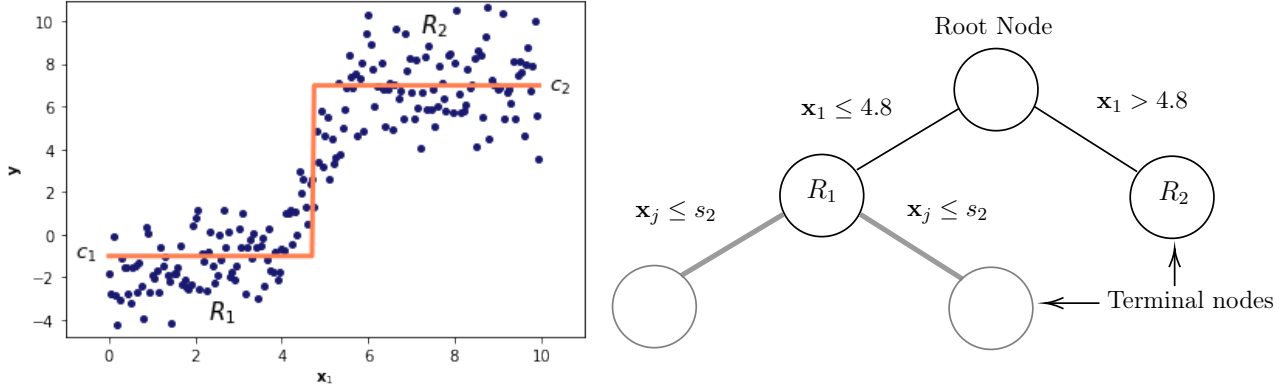
**Figure 2.2** – An unconstrained gradient step in function space here the  $f(\mathbf{x}_i)$  is the function evaluated the vector  $\mathbf{x}_i$ . Each step corresponds to adding a new basis function to the ensemble. The figure is adapted from figure B.1, p 98 [16].

### 2.2.8 Gradient Tree Boosting

In gradient tree boosting, the basis functions are regression trees,  $\mathcal{H} := \{\text{regression trees}\}$ . We will cover how a tree is grown and how it fits the framework of gradient boosting.

#### Building a Generic Regression Tree

To grow a tree, we start from the root node with all  $k$  input features. We now split on a feature. In figure 2.3a, we split on feature  $\mathbf{x}_1$  with a value around 4.8.



(a) One split on  $\mathbf{x}_1$  with split value around 4.8.

(b) Depiction of tree building procedure.

**Figure 2.3** – Figure 2.3a shows how one split divides the data into the regions  $R_1$  and  $R_2$ . The mean value of the  $y_i$ s in regions  $R_1$  is denoted  $c_1$ . Likewise for  $R_2$ .

We can depict this split as a tree, which corresponds to the top nodes of figure 2.3b. After this initial split, we can split ones again on some feature  $x_j$  and grow the tree. In this way, we partition the data into regions like  $R_1$  and  $R_2$ . When we want to predict a new price, we just let the input variables fall down the tree until it reaches a *terminal node*. We can then take the mean of the other response variable in that terminal node to get our prediction. In figure 2.3a, the means are denoted  $c_1$  and  $c_2$

The question is now how to determine the regions in an optimal way. Consider a dataset with  $s$  features and  $N$  samples such that we have a feature matrix  $\mathcal{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k]^T \in M_{N \times k}(\mathbb{R})$  with corresponding outputs  $\mathbf{y} \in \mathbb{R}^N$ . We now partition  $\mathcal{X}$  into two disjoint



regions  $\{R_1, R_2\}$

$$\mathcal{X} = R_1 \cup R_2 \text{ and } R_1 \cap R_2 = \emptyset \quad (2.23)$$

We want to find  $R_1, R_2$  such that given some new input  $\mathbf{x}_{N+1,\cdot} \in \mathbb{R}^k$ , we can predict the response  $\hat{p}_{N+1}$  only by determining which region the  $\mathbf{x}_{N+1,\cdot}$  belongs to. Introduce the prediction function  $f : \mathbb{R}^k \mapsto \mathbb{R}$

$$f(x) = c_1 \mathbf{1}_{\{\mathbf{x}_{N+1,\cdot} \in R_1\}} + c_2 \mathbf{1}_{\{\mathbf{x}_{N+1,\cdot} \in R_2\}} \quad (2.24)$$

where  $c_1, c_2 \in \mathbb{R}$  are chosen such that they minimize the loss function. For the loss function  $\ell_2(y_{N+1}, \hat{y}_{N+1}) = (y_{N+1} - \hat{y}_{N+1})^2$ , we would take the average of the  $y_i$ s in each region. This is of course only an optimum if  $R_1$  and  $R_2$  are chosen optimally. We have to find the best splitting variable  $j$  and splitting value  $s$  to obtain the regions  $R_1(j, s) = \{\mathbf{x} | x_j \leq s\}$ ,  $R_2(j, s) = \{\mathbf{x} | x_j > s\}$  with average responses of  $c_1(j, s), c_2(j, s)$  that minimize the loss

$$(j^*, s^*) = \arg \min_{(j,s)} \sum_{i: x_{i,\cdot} \in R_1} (p_i - c_1(j, s))^2 + \sum_{i: x_{i,\cdot} \in R_2} (p_i - c_2(j, s))^2 \quad (2.25)$$

We can now try all the combinations of splits between all input features and data points to find the optimum. After this split we can repeat the procedure for each of the regions  $R_1(j^*, s^*), R_2(j^*, s^*)$  to build the tree as indicated in 2.3b.

### Gradient Tree Boosting and Essential Hyperparameters

When we do gradient boosting and use trees as the basis function, then we have to be careful as this combination is very prone to overfitting [14]. For that reason, we need to regularize the trees sufficiently and control the learning rate and the number of boosting iterations. We will use *LightGBM* [2], in short LGBM, as our gradient boosting framework. We will introduce the hyperparameters we found essential in our application. To find the optimal parameters, we use cross-validation, see chapter 5. Initially, we adopted a generic hyperparameter tuning framework from top practitioners [18]; however, these parameter intervals were not optimal for quantile regression as the forecasts were far from probabilistically calibrated. Using ideas from online threads with the developers [19] and trial and error, we found that the learning rate should be higher than usual, the number of iterations for the boosting should be lower than usual, and the number of terminal nodes should be enforced above 600, see appendix D for further details. To make the model adapt to recent changes in the energy market, we also used local exponential weighting as in section 2.2.4.

### Feature importance

In the construction of the tree, we can take out information value information that makes the trained model more interpretable. This is called feature importance, and we use the 'gain' [20]. If we split on a feature  $j$ , and the split hugely increases the accuracy of the tree, then we would assign more importance to that feature. As we retrain the model over time, we can track the feature importance over time and look for shifts in the market and show this to experts to see if the model picks up the right trends, see e.g. figure 6.5b.

## 2.3 Evaluation of Probabilistic Forecasts

We want a forecast that is *reliable* and of good *quality*. The forecast is *reliable* if the forecast probabilities coincide with the observed prices. To check this, we use PIT diagrams. To

evaluate if the probabilistic forecast is of good quality, we introduce the CRPS score, which is the probabilistic generalization of the deterministic MAE.

### 2.3.1 PIT diagrams

To assess if the probabilistic forecasts are reliable, we use the computed CDFs for each lead time  $\hat{F}_{d+3,h}$ . To be probabilistically calibrated, we require [21]

$$\hat{F}_{d+3,h}(p_{d+3,h}) \sim U[0, 1] \quad (2.26)$$

where  $p_{d+3,h}$  denotes the observed prices. This amounts to asking if the forecasted probabilities are consistent with the observed long-run relative frequencies. In practice, we can assess this using the probability integral diagram [22], *PIT* diagram. If the PIT diagram of  $\hat{F}_{d+3,h}(p_{d+3,h})$  looks uniform, then the forecast is assumed reliable. We should note that  $\hat{F}_{t+d,h}(p_{t+d,h})$  of a series of  $N$  points, will at best follow a uniform distribution sampled with  $N$  points [23]. Because there still might be serial correlation in the residuals, then the series of  $\hat{F}_{t+d,h}(p_{t+d,h})$  is not i.i.d. hence we will expect the PIT to be even less uniform [24]. We will take a pragmatic approach and accept that the PIT diagrams might not be perfect but look for obvious biases.

### 2.3.2 CRPS Scores

With a quantile forecast, we want a metric that evaluates the forecast as a whole. The quantile forecast is an ensemble of  $M$  predictions, each focused on one part of the distribution. We want each quantile to cover its nominal level i.e. we want it to be *elicit* [25]. Therefore, consider the observation  $y_{t+k}$  and the predicted densities  $\hat{F}_{t+k|t}(x)$ , then the *continuous ranked probability score* is defined as:

$$\text{CRPS}(k) = \int_x \left[ \hat{F}_{t+k|t}(x) - H(x - y_{t+k}) \right]^2 dx, \quad H(x) := \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (2.27)$$

Here  $H$  is the Heaviside step function. An intuitive understanding and decomposition of the CRPS can be found in [26]. The CRPS is negatively oriented and is 0 if probabilistic predictions are perfect hence it is a proper score [25]. It takes the same unit as the variable hence in our case, GBP/MWh. It is shown in [25] that the CRPS is the generalization of the MAE, and the CRPS reduces to the MAE for points forecasts. Therefore, we can directly compare MAE and CRPS to check if our probabilistic forecast contains more valuable information than the point forecast.

## 3 | Machine Learning Strategies for Time Series Forecasting

In the following, we will consider how we can rephrase a time series problem as a supervised learning problem. We will also use theory from dynamical systems to interpret what it means to reformulate the problem in this way. To understand why we need to reformulate the problem, we first consider a generic supervised learning problem.

### 3.1 A Generic Supervised Learning Problem

Consider a generic supervised learning problem with samples of input-output pairs  $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$  with  $\mathbf{x}_i \in \mathbb{R}^{d_x}$  and  $\mathbf{y}_i \in \mathbb{R}^{d_y}$  for  $i = 1, 2, \dots, N$ . Most machine learning algorithms are built to find a relationship between input-output pairs without explicitly considering the order of the pairs. If the pairs in  $\mathcal{D}$  is a time series, then there is most likely a temporal dependency between consecutive pairs in  $\mathcal{D}$ . If we naively feed some model  $M$  with  $\mathcal{D}$ , then  $M$  will not suspect any dependency between consecutive pairs and we as model builders will essentially collapse the temporal dimension in  $\mathcal{D}$ . Therefore, we need to present the data in a way that takes the temporal dimension into account.

#### 3.1.1 Intuition from Time Series Models

ARIMA models include the temporal dimension with ease. Consider a one step prediction using ARIMA,  $\hat{\mathbf{y}}_{t+1}$ . It takes as input all observed values at favorable lags,  $\mathbf{y}_{t-1}, \mathbf{y}_{t-2}, \dots, \mathbf{y}_{t-k}$ , and the errors it made a previous predictions  $\epsilon_{t-1}, \epsilon_{t-2}, \dots, \epsilon_{t-k}$ . The temporal dimension is sustained by including information of the past series at the prediction of  $\mathbf{y}_{t+1}$ . Let  $\Omega_t$  denote the information up to time  $t$ , then to improve our model building procedure, we should include  $\Omega_t$  in the input-output pair such that

$$\mathcal{D} = [(\{\mathbf{x}_t, \Omega_t\}, \mathbf{y}_t), (\{\mathbf{x}_{t+1}, \Omega_{t+1}\}, \mathbf{y}_{t+1}), \dots, (\{\mathbf{x}_{t+N}, \Omega_{t+N}\}, \mathbf{y}_{t+N})] \quad (3.1)$$

In practice, we simply include lagged values as new features [27].

### 3.2 Interpretation of Lag Value Proposition

There are two main approaches to describe time-evolving system [28]. The system can be described as a realization of a stochastic process where the stochasticity in the system is a consequence of independent degrees of freedom acting in a way we cannot fully capture. The system can also be seen as a deterministic dynamical system where a few degrees of freedom interact in a nonlinear, complicated way which leads to deterministic chaos, which

generates the apparent random behavior [28]. Consider the deterministic interpretation and let  $s_t \in \mathbb{R}^{n_s}$  denote a time-dependent state vector that evolves according to

$$s_t = \mathcal{F}^t(s_0) \quad (3.2)$$

where  $\mathcal{F} : \mathbb{R}^{n_s} \mapsto \mathbb{R}^{n_s}$  is the flow map that evolve the states at time  $t$ . Then a time series can be interpreted as the observable part of the states above. Let  $y_t \in \mathbb{R}^{n_d}$  denote the states of the observable time series. Now in the absence of noise, the states of the time series  $y_t$  are related to the states of the dynamical system,  $s_t$ , through the measurement function  $\mathcal{G} : \mathbb{R}^{n_s} \mapsto \mathbb{R}^{n_d}$

$$y_t = \mathcal{G}(s_t) \quad (3.3)$$

For all real-world systems  $\mathcal{F}$  and  $\mathcal{G}$  are unknown hence from the measurement we will never be able to correctly reconstruct the state space only from the measurements at time  $t$ . However, for a great variety of deterministic dynamical system, we can reconstruct the state space using delay coordinates of the observations [28] [29]. It follows from Takens theorem [30] that there exists a diffeomorphic mapping,  $\phi : \mathbb{R}^{n_s} \mapsto \mathbb{R}^n$ , between the states of the system and delay observations time series such that

$$\begin{aligned} \phi(s_t) &= \left[ \mathcal{G} \left[ \mathcal{F}^{-k}(s_t), \dots, \mathcal{F}^{-k-n+1}(s_t) \right] \right]^\top \\ &= [y_{t-d}, y_{t-d-1}, \dots, y_{t-d-n+1}]^\top \end{aligned} \quad (3.4)$$

here  $k$  is the lag time and  $n$  is number of past values that we place in the delay vector  $\mathbf{y} = [y_{t-d}, y_{t-d-1}, \dots, y_{t-d-n+1}]^\top$ . Taken showed that if  $n \geq 2n_s + 1$ , then  $\phi$  exists and has a smooth inverse,  $\phi^{-1}$ . Using this results, we can introduce  $F$

$$F^t(\mathbf{y}) = \phi \circ \mathcal{G}^t \circ \phi^{-1}(\mathbf{y}) \quad (3.5)$$

$F$  describes the induced dynamics from the embedded coordinates. In [31] they describe the implication of presence of noise and state that in practice, only a few favorable lags will contain all valuable information. We can understand the inclusion of lagged values as a way to formulate a learning problem to approximate the induced dynamics of equation 3.5. So when we are including  $\Omega_t$  to the data pairs, we can see it as a way to learn the induced dynamics.

### 3.3 Strategies for Multi-Step Time Series Forecasting

This insight still leaves us with a couple of ways to construct the pairs of equation 3.1. In our project, we want a multi-horizon forecast. This gives rise to different strategies [28] [32] to try to approximate the induced map. We think that the underlying process for each hour is best modeled separately hence we adopt the direct strategy [28].

$$\hat{p}_{d+3,h} = f_h(X_{d,h}, X_{d-1,h}, \dots, X_{d-N,h}) + \omega_{d+3,h} \quad (3.6)$$

where  $f_h : X_d \mapsto \mathbb{R}$  for each hour  $h \in \{0, 1, \dots, 23\}$  and we use price inputs from different days  $X_{d_i,h}$ .  $N$  is the number of days back we include in our case we set  $N = 14$ . We strongly considered the *MIMO* strategy [33] [28]. Then we would only need one model for each day and it would take the interdependency between hours into account. The approach would increase the model complexity dramatically. For the probabilistic forecasts, we can instead couple predictive densities for different lead times using a Gaussian copula and explicitly track the dependency over time [34]. We will introduce this technique in the following.

## 4 | Generation of Statistical Scenarios

In the following, we will show how we can couple the prediction errors of our probabilistic distribution in a latent space using a Gaussian copula [34].

### 4.1 From Predictive Distribution to a Multivariate Gaussian

Let  $\hat{F}_{d+3,h|d}$  be the predicted cumulative distribution for an hour  $h$  characterized by the quantiles  $\{\hat{q}_{d+3,h}^{(\alpha)}\}$ , see section 2.2.1. Introduce the random variable  $Y_h$  and let  $Y_h^{(d)}$  be the realization of  $Y_h$  at day  $d$  defined such that

$$Y_h^{(d)} = \hat{F}_{d+3,h|d}(p_{d+3,h}) \quad (4.1)$$

where  $p_{d+3,h}$  is the realized price. In practice, we construct  $\hat{F}_{d+3,h|d}$  using linear interpolation between the predicted quantiles,  $\{\hat{q}_{d+3,h}^{(\alpha)}\}$ . If  $\hat{F}_{d+3,h|d}$  is probabilistically calibrated, then we know from section 2.3.1 that  $Y_h^{(d)} \sim U[0, 1]$ . Given  $Y_h^{(d)} \sim U[0, 1]$ , we can directly transform  $Y_h^{(d)}$  into a standard normal distributed variable  $X_k \sim \mathcal{N}(0, 1)$  by using the probit function i.e. the inverse of the CDF for a standard normal distribution:

$$\Phi^{-1} : p \mapsto \sqrt{2}\text{erf}^{-1}(2p - 1) \quad (4.2)$$

where  $\text{erf}^{-1}$  is the inverse error function then

$$X_k^{(d)} = \Phi^{-1}(Y_k^{(d)}) \quad \text{for all days } d \text{ and } h = 0, 1, \dots, 23 \quad (4.3)$$

Consider now the vector for the entire forecast horizon,  $\mathbf{X}^{(d)} = [X_0^{(d)}, X_1^{(d)}, \dots, X_{23}^{(d)}]^\top \in \mathbb{R}^{24}$ . As  $X_k \sim \mathcal{N}(0, 1)$ , then  $\mathbf{X} \in \mathcal{N}(\mathbf{0}, \Sigma)$  where  $\mathbf{0}$  is a vector of zeros and  $\Sigma$  is the covariance matrix ideally with ones in the diagonal. We will update  $\Sigma$  each day according to the rule

$$\Sigma_d = \lambda \left( \frac{d-2}{d-1} \right) \Sigma_{d-1} + \left[ 1 + \lambda \left( \frac{1}{d-1} - 1 \right) \right] \mathbf{X}^{(d)} \mathbf{X}^{(d)\top} \quad (4.4)$$

where  $\lambda \in [0, 1)$  is a forgetting factor.  $\lambda$  can be adjusted such that  $\Sigma_d$  follows the shifts in the interdependence structure over time [34].

### 4.2 From Multivariate Gaussian to Statistical Scenarios

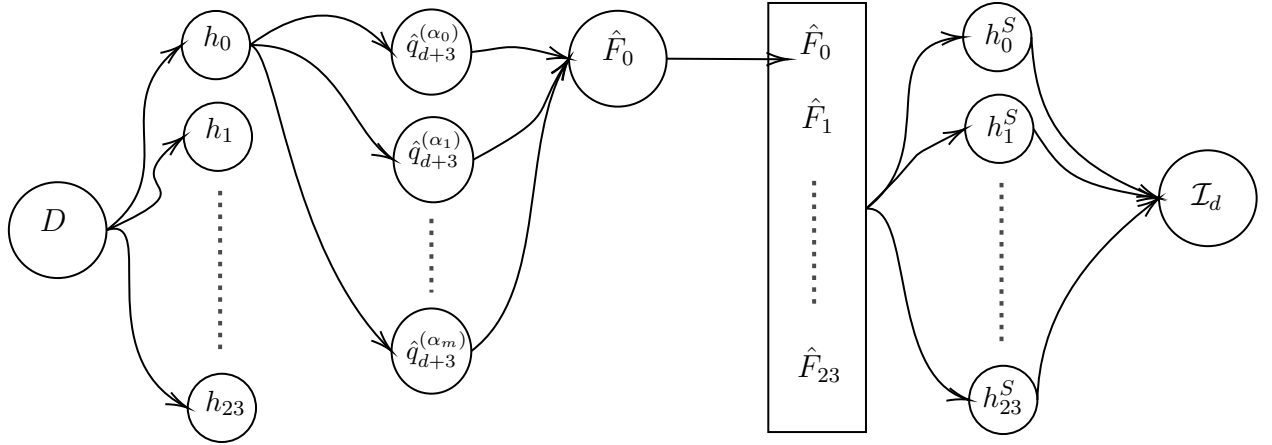
At a given day  $d$ , we can now sample from  $\mathbf{X} \sim (0, \Sigma_d)$ . Let  $\mathbf{X}^{(s)}$  denote one realization of  $N_s$  computed samples and let  $X_h^{(s)}$  be the scenario for a specific hour  $h$ . We can now apply the inverse probit function to transfer back to a uniform random variable:

$$Y_h^{(s)} = \Phi \left( X_h^{(s)} \right) \quad \text{for all } h = 0, 1, \dots, 23 \text{ and } s = 1, 2, \dots, N_s \quad (4.5)$$

To obtain the price for a scenario  $s$  with of hour  $h$ , we apply the inverse cumulative distribution,  $\hat{F}_{d+3h|d}$ . We also use linear interpolation between the estimated quantiles to approximate the whole distribution.

$$\hat{p}_{d+3,h}^s = \hat{F}_{d+3h|d}^{-1} (Y_k^s) \quad (4.6)$$

Now we have generated scenarios that are consistent with the probabilistic forecast and take the interdependence structure of the forecast horizon into account. It also allows us to use directly the formula 1.1 to calculate the intermittency risk, which would otherwise not be straightforward for the probabilistic forecast. We can see this as a Monte Carlo simulation framework that can be used to generate prices scenarios of which we can then take quantiles to get a probabilistic distribution of the daily intermittency risk. Below is a figure for obtaining the probabilistic forecast of the daily intermittency risk.



**Figure 4.1** – The model building procedure to make the probabilistic forecast for the intermittency risk. First, we split the day into each hour. Then we split each hour into  $m$  quantiles used to characterize the CDF  $\hat{F}_h$ . Using The CDFs for each hour, we use a Gaussian copula to couple the hourly distributions. We now generate hourly prices scenarios and weigh them by the predicted power production to obtain the intermittency risk.

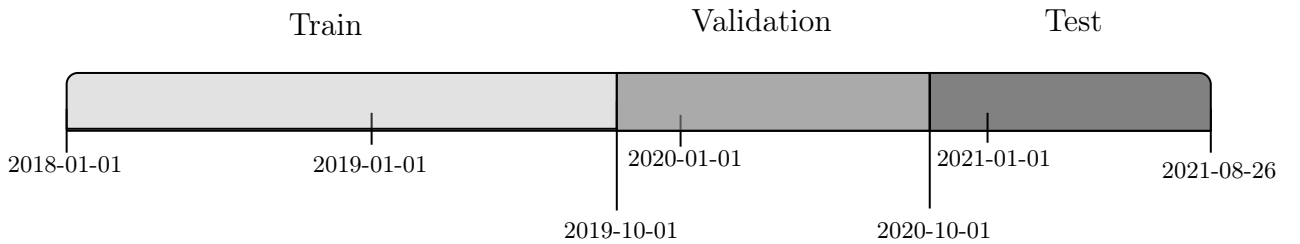
## 5 | Methodology

To have a firm framework of our workflow, we adapt that of a generic model building procedure [15]

1. Split labeled data into **training**, **validation**, and **test** sets.
2. Repeat until satisfied with performance on validation set:
  - (a) Build or revise your preprocessing and feature extraction methodology.
  - (b) Choose some ML algorithm.
  - (c) Train ML model with various hyperparameter settings.
  - (d) Evaluate prediction functions on validation set.
3. Retrain model on train and validation data
4. Evaluate performance on test set.

### 5.1 Split labeled data into training, validation, and test sets.

As the order of the data constitutes the temporal dimension, we split the data as depicted in figure 5.1. We do this for all the available input features, both the ones we have on an hourly and a daily basis, see appendix E. Because we consider the hours separately, we construct datasets for each hour and append the daily data to each of them.



**Figure 5.1** – The used train, validation and test split

The split in figure 5.1 corresponds approximately to the 50%-25%-25% for training-validation-test split suggested in [14]. We have a slightly larger validation set because we want a model that can work during summer and winter. We want a model that would perform best on unseen data according to some score metric,  $S$  [14]. As we work with time series, the cross-validation should be treated with care to avoid data leakage and sustain the temporal dimension [14] [15]. Therefore we use time based cross-validation.

### 5.1.1 Time Based Cross-Validation

1. Construct  $k$  new series  $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_k$  of consecutive elements from data,  $\mathcal{D}$ . For each  $\mathcal{D}_i$  take the first 80% of  $\mathcal{D}_i$  for train,  $\mathcal{D}_i^t$ , and leave the last 20% for validation  $\mathcal{D}_i^v$
2. For  $i = 1, 2, \dots, k$ :
  - (a) train model  $M_\gamma$  on  $\mathcal{D}^t$
  - (b) calculate the score  $S_{i,\gamma}$  of  $M_\gamma$  on  $\mathcal{D}_i^v$
3. calculate  $\hat{S}_\gamma$  where  $\hat{S}_\gamma = \sum_{i=1}^k S_{i,\gamma}$  is the mean

We can construct  $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_k$  either using *Forward chaining* or *sliding window* approach [15] [35]. With forward chaining, we gradually increase the amount of data we train on as we progress through time, and for the sliding window, we consider a fixed window size of time. For the SARIMA model, we use the sliding window as we could construct much better and simpler models using this approach. With the local linear model and LGBM, we weight the data of the immediate past using exponential decay, which corresponds to a smooth sliding window approach as the impact of data points far away is diminishing. This ensures that we do not feed too much data, which would make our model very complex as the model would try to learn complex long-term trends. As we use cross-validation to find  $d_{1/2}$ , we believe the procedure can better balance the trade-offs between the approaches. Graphics and further details on *Forward chaining* and *sliding window* can be found in G.

### 5.1.2 Preprocessing

For the point forecasts, the models can be used directly on the price data. For the probabilistic forecast, the baseline only uses the price data, the local linear model uses the predicted renewable power penetration defined in appendix C, and the *LightGBM* is fed with all input features. The following will mostly be relevant for the machine learning pre-processing pipeline. As a start, only a couple of features for the interconnector capacities e.g. UK-FR are removed because they contain missing data or are constant. To decorrelate the input features, we use ZCA-cor [36]. Details on this transformation can be found in appendix F.

## Feature Engineering

In section 3.1, we introduced the lagged feature values as a way to create new features. We include the latest 14 days of the N2EX prices as new features to the feature matrix. We also introduce a couple of extra features such as aggregated available gas capacities and the predicted renewable energy penetration, see appendix C.

### 5.1.3 Choose some ML algorithm

We use *LightGBM* [2] but tried other boosting machines, however, as already suggested in [2], *LightGBM* is much faster than XGboost [37] and the scikit-learn GradientBoostingRegressor [38].

### 5.1.4 Train ML model with various hyperparameter settings

For the local linear model, we finetune the bandwidth and the halftime  $d_{1/2}$  for each model for all lead times and all quantiles using cross-validation. In section 2.2.8, we covered the



essential parameters to tune for the LGBM model. We could dramatically speed up the cross-validation using *Optuna* [39] which is build-on optimization library designed for ML projects.

### Adaptivity

We do rolling forecasting [9] for all models which means we will update the models regularly to make them adaptive. We do this to best imitate how the model would be used if deployed. The local linear model is adaptive by construction as we include new data points when we estimate the parameters in equation 2.12. For the LGBM, we refit the model each week to adapt to recent market changes. As we only have limited computer capacity, we only do cross-validation to find hyperparameters on the first of October and the first of April of each year.

#### 5.1.5 Evaluate prediction functions on validation set

For graphical assessments we display the *nominal coverage rate* [40] i.e. the price range in which the price is expected to be in with a probability  $(1 - \alpha)$ . For a nominal coverage rate of  $(1 - \alpha)$ , the price is expected to be between the quantiles with nominal levels  $\alpha/2$  and  $(1 - \alpha/2)$ . A numeric score is the ultimate model selection criterion. As *Ørsted* defines the best model as the one with the lowest MAE of the daily intermittency risk. We adopt this criterion and select the final model based on the MAE on the validation set. For the probabilistic forecast, we will take out the median of the ensemble and use that to calculate the MAE. If the best CRPS score is lower than the best MAE, then we will also include the model with the best CRPS score because it means that the probabilistic forecast carries more valuable information that could be relevant for more informed decision-making.

## 6 | Results

In this section, we will see how well the models can predict energy prices and report interesting model findings on the validation data. Then we will present the performance of the prediction of intermittency risk. We will often refer to table 6.1 with the scores for each model on the validation period from 2019-10-01 to 2020-10-01.

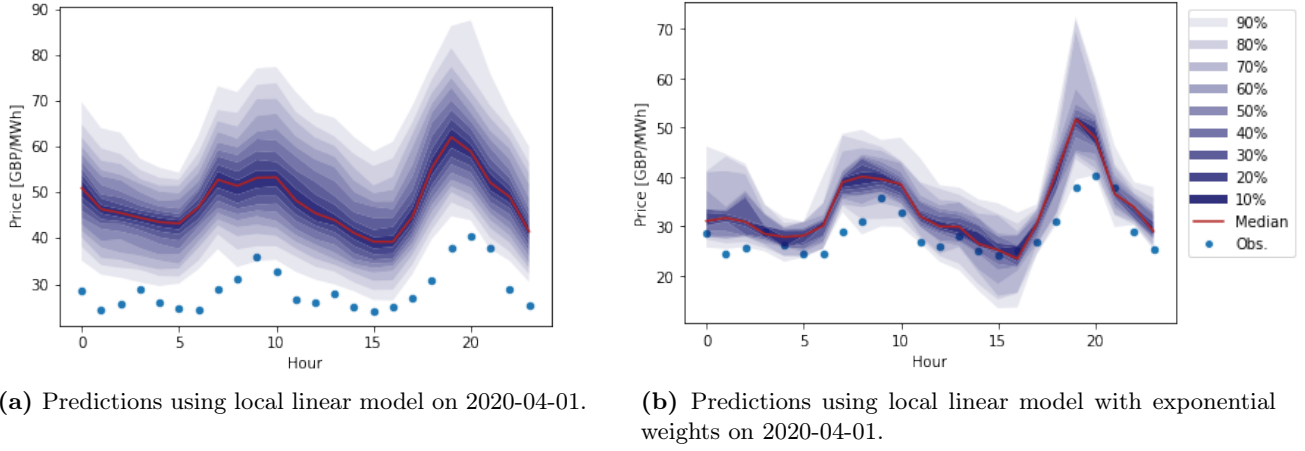
### 6.1 Price Forecasting

In the first columns of table 6.1, we see the performance of the price predictions. The baseline performs slightly better than the ARIMA model for the point forecast. This suggests that a simple ARIMA model is inappropriate for this application. We use the procedures in [10] to fit the ARIMA model, find that variance stabilizing transformation only helps a little, but the model performs much better if we limit the data to only be the windows of the last 180 days.

Models	Price		Intermittency Risk		
	MAE	CRPS	MAE	CRPS	MAE <sub>S</sub>
Point Baseline	5.886		0.707		
ARIMA Model	5.954		0.695		
Probabilistic baseline	5.634	4.272	0.629	0.480	0.645
Local Linear	11.487	8.316	0.536	0.359	0.506
Local Linear Exp weights	4.878	3.579	0.430	0.307	0.419
Local Linear Sea weights	4.882	3.581	0.430	0.309	0.420
LightGBM	3.608	2.785	0.392	0.306	0.403

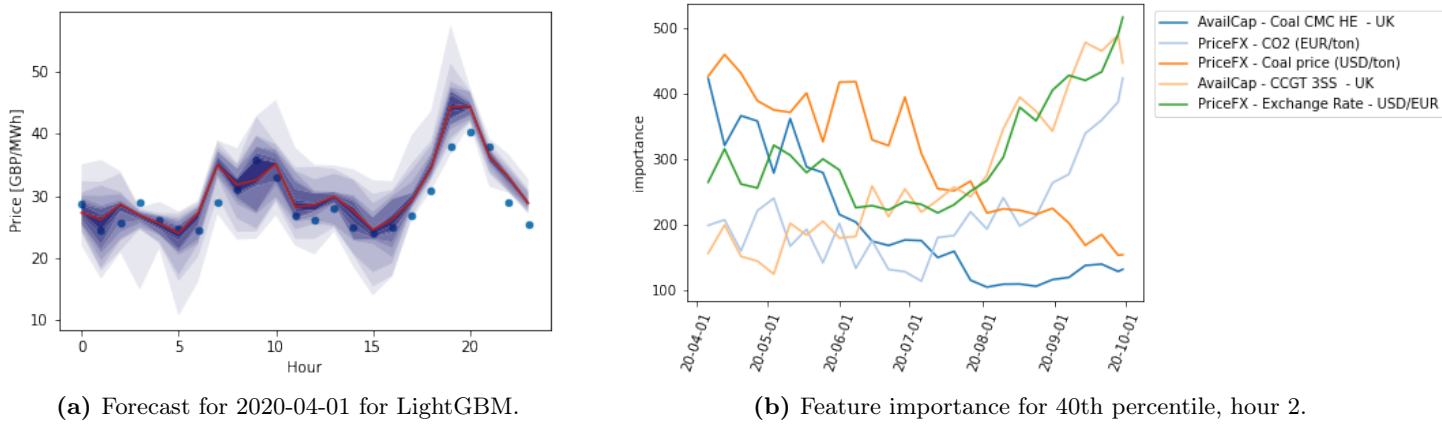
**Table 6.1** – Results for the prediction of the price and intermittency risk. All scores have the unit GBP/MWh.

For the probabilistic price forecasts, we see that in general the CRPS is lower than the MAE which means that the probabilistic forecasts add more valuable information. Note also that the models with the best MAE scores also have the best CRPS scores. The MAE of the probabilistic baseline is 5.634 which is better than the ARIMA and way better than the local linear model without the exponential weights with an MAE of 11.487. The local linear models with exponential weights and seasonal weights perform way better with MAE on 4.878 and 4.882 respectively. This indicates a clear need for the weights on the data points of the immediate past. Without the weights, the local linear model has a huge positive bias during the summer and spring months, see figure 6.1a and compare with 2.1b. All quantiles are displayed in terms of nominal coverage rates, see section 5.1.5. The optimal  $d_{1/2}$  for the weights for the local linear model is surprisingly low which indicates just how non-stationary the time series is. We search for  $d_{1/2} \in [20, 180]$  and most often,  $d_{1/2}$  is very close to 20.



**Figure 6.1** – In figure 6.1a, there is an obvious positive bias for the local linear model. When we use exponential weights, the bias disappears as in figure 6.1b.

We find that seasonal weights do not improve our model. We suspect that the seasonal weights become redundant because  $d_{1/2} \approx 20$ , hence data points more than 180 days away from  $d$  already have a diminishing impact on the parameter estimation. The found bandwidths increased slightly for hours with ranges of sparse data, e.g. predicted power penetrations above 0.7 in figure 2.1a. In figure 6.2a, we see the strong performance of the LGBM. We searched extensively in different ranges to find the optimal hyperparameters and obtain solid performance when we regularize sufficiently and lower the learning rate as described in 2.2.8. For a graphical illustrations of the entire period see appendix H. In figure 6.5b, we see how the importance of features changes over time for quantile 30, hour 2. Indeed coal had a huge impact on commodity prices in the EU in the summer of 2020 [41] and energy largely was produced by CCGT towards the end of the [4]

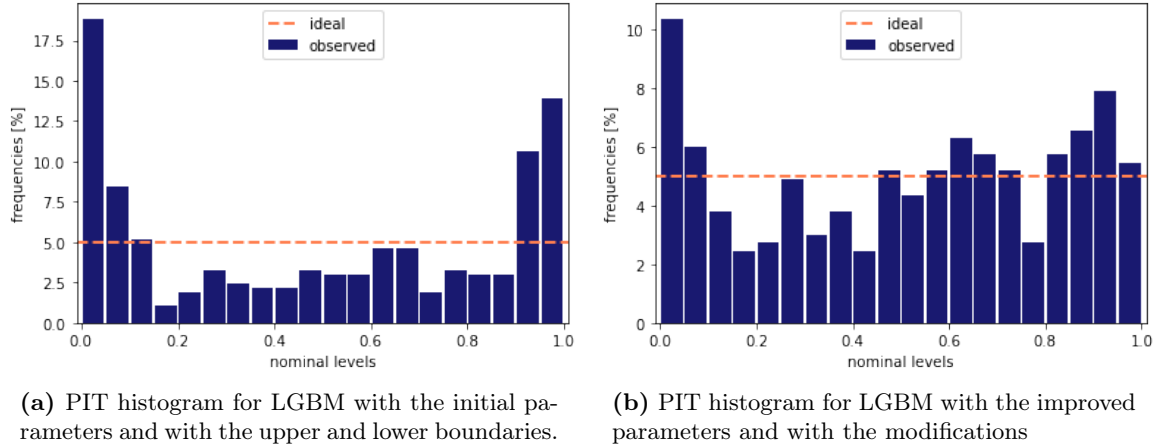


**Figure 6.2** – In figure 6.2a, we see a graphical example of the LGBM. In 6.5b, we see how the feature importance changes over time as the market changes.

## 6.2 Probabilistic calibration

Initially, we set the lower and upper quantiles to be that of the minimum and maximum of the allowed market bids i.e.  $q^{(0)} = -500$  and  $q^{(1)} = 3000$  GBP/MWh. This proved

very insufficient because the prices are mostly around 50 GBP/MWh hence both a slight deviation from  $q^{(0)}$  and a huge deviation from  $q^{(0)}$  would be assigned the same nominal level. This made many of the probabilistic forecasts biased. An example of this can be seen in figure 6.3a. This is the PIT histogram of the LGBM with parameters that made the forecast too optimistic combined with the effects of the above, which made us reconsider the modeling framework.



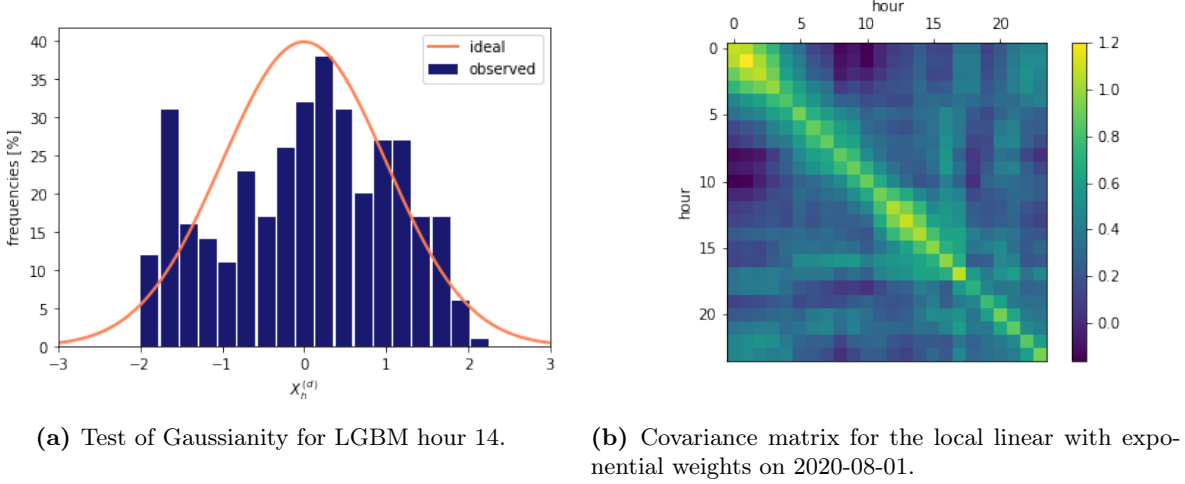
**Figure 6.3** – PIT histograms before we included the modifications

To improve the calibration of our forecasts, we introduce a way to handle outliers. We want to model the normal behavior of the market and therefore we redefined the upper and lower quantiles such that  $\hat{q}^{(0)} = \frac{2}{3}\hat{q}^{(0.05)}$  and  $\hat{q}^{(1)} = \frac{4}{3}\hat{q}^{(0.95)}$ . If some price  $p_i$  with  $p_i < \hat{q}^{(0)}$ , then we classify it as an outlier and assign it some density  $Y_{p_i,0}$  where  $Y_{p_i,0}$  is sampled uniformly in the interval  $(0, 0.3]$ . Likewise, if  $p_i > \hat{q}^{(1)}$  then we sample  $Y_{p_i,1}$  from a uniform distribution on the interval  $[0.7, 1)$ . To show the combined effect of this and using better hyperparameters for the LGBM, see figure 6.3b. To see the isolated effect of the change mention above, see appendix I where the differences are displayed for the probabilistic baseline. Indeed, figure 6.3b is not perfectly calibrated but it is within what we accept. Figure 6.3b represents the calibration of most hours and applies also applies roughly for the local linear with exponential weights though it tended to have its highest frequencies for nominal levels close to 1. For hour 20 and hour 21, the PIT histograms reached frequencies up to 16% for all models, which is not ideal. Of course, the local linear model without exponential weights is far from calibrated.

### 6.3 Generation of Statistical Scenarios

We now follow each of the introduced steps to check if our assumptions hold as described in chapter 4. We start with the transformation from uniform to Gaussian. As the probabilistic forecasts are not perfectly calibrated, the transformed variable  $X_k$  of the marginal distribution in equation 4.3 are not perfectly Gaussian. For a calibration level of figure 6.3b, the distribution is reasonably Gaussian see figure 6.4a. Our models cannot fulfill the ideal conditions as presented in chapter 4; however, we take a pragmatic view and try with the model we have. When we calculate the covariance matrix of equation 4.4 it always is a diagonally dominant matrix, see figure 6.4b, but the off-diagonal elements changes depending on time and the model. We generate 10,000 scenarios for day and use

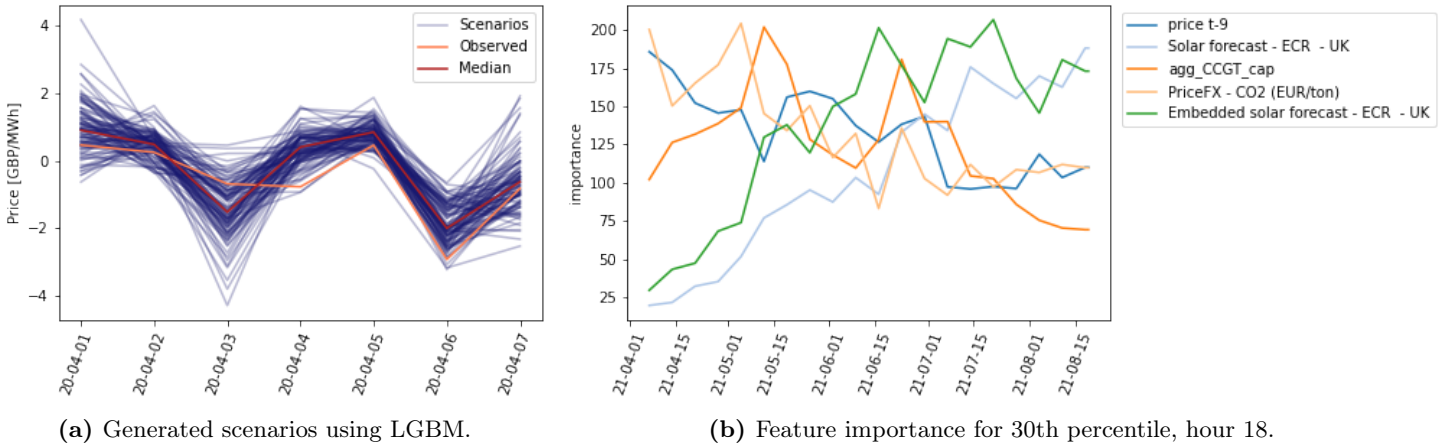
equation 1.1 to obtain 10,000 scenarios of the daily intermittency risk. Using the error of the intermittency risk, we can cross-validate to find the optimal forgetting factor  $\lambda^*$ . We find the optimal forgetting factors to be 0.96, 0.98, 0.98, 0.99 and 0.96 for the models in the order in which they appear in table 6.1.



**Figure 6.4** – In figure 6.4a, we see how Gaussian a calibration level of figure 6.3b. Figure 6.4a displays an example of the diagonal dominant covariance matrix on 2020-08-01.

## 6.4 Intermittency Risk

The intermittency risk can be calculated directly for the point forecasts using equation 1.1 and the performance is evaluated using the MAE from equation 2.2. For the probabilistic forecast, we will see if the coupling of the marginal distribution leads to a better estimate of the intermittency risk. In figure 6.5a we see a 100 of the generated scenarios. Consider 10,000 of these and then take out the quantiles  $\hat{q}_s^{(\alpha_i)}$  with nominal levels  $\alpha_i \in \{0.05, 0.1, \dots, 0.95\}$ . Using the quantiles, we now have a probabilistic forecast for the intermittency risk. The median  $\hat{q}_s^{(0.5)}$  is used to generate the results presented as  $MAE_s$ .



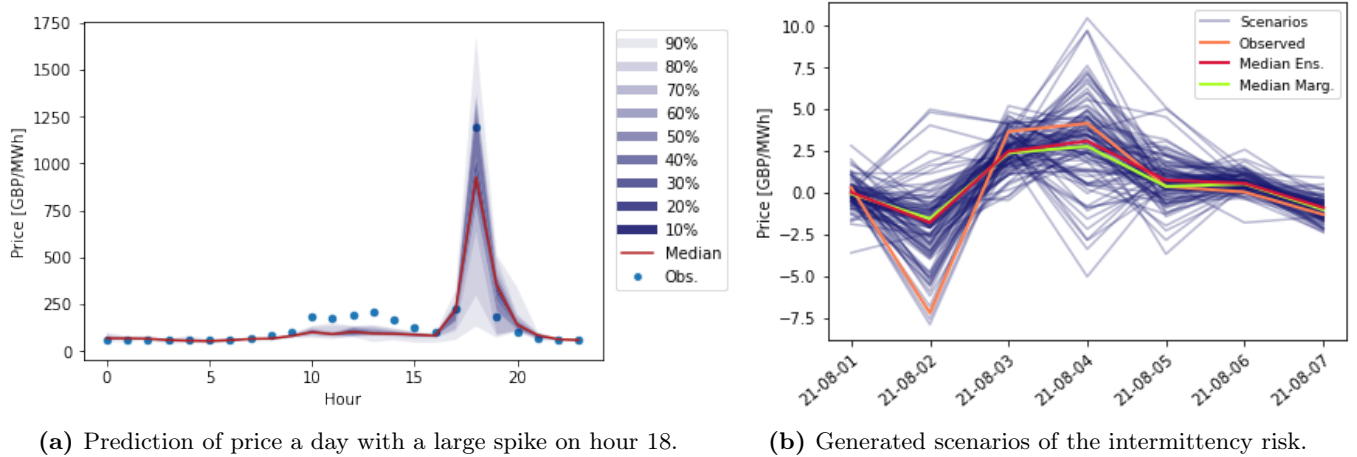
**Figure 6.5** – In figure 6.5a, we see how 100 generated scenarios and  $\hat{q}_s^{(0.5)}$  calculated with 10,000 scenarios. Figure 6.5b, we see the top 5 features of the 40th percentile hour 2.

In the last set of columns in table 6.1, we see some interesting results. It seems that the

MAE of the intermittency risk is slightly lower when we use the median of the marginal hourly distribution instead of the median of the scenarios. However, there is more valuable information in the probabilistic forecast using the generated scenarios because the CRPS is lower. No matter if we evaluate using the MAE,  $MAE_s$  or CRPS, we see that the LGBM is the model we should select. We should also generate scenarios and obtain a probabilistic forecast as the CRPS is lower than the MAE. With the selected model, we can now test on the test data.

## 6.5 Evaluation on Test Data

The power market in 2021 turned out very volatile and reached prices of 1499.62 GBP/MWh in the beginning of the year for hour 18. An example is seen in figure 6.6a. Notice, that the model can actually capture the price spike. This means that the model has adapted to an environment with high spikes, and since the model is very data-driven, this underpins that it has been presented with multiple days with high spikes. This non-stationary and volatile time series is hard to predict, which the metrics in table 6.2 also suggest if we compare it with the results of the validation period. For all scores, we again see that the CRPS score is lower, and figure 6.6b suggests why. In figure 6.6b, we see seven days in August 2021 with 100 scenarios, the median of 10,000 generated scenarios, and the median of the marginal distributions. The median both deviate from the price but the price is always inside the span of the generated scenarios, which gives us valuable information about the price that can be used for decision making. The hourly distribution of the CRPS and an example of the predictions for the test period can be found in appendix J. The increase in MAE and CRPS could also be because the model is unable to generalize well to new market environments. We will discuss improvements to our model next section.



**Figure 6.6** – Results for the test period that turned out very volatile. Figure 6.6b demonstrate how more information is captured in the quantiles computed from the generated scenarios

	Price		Intermittency Risk		
Model	MAE	CRPS	MAE	CRPS	$MAE_s$
LightGBM	7.974	6.267	1.129	0.896	1.160

**Table 6.2** – Results for the prediction of the price and intermittency risk on the test data. All scores have the unit GBP/MWh.

## 7 | Discussion

### 7.1 Adaptivity

In our setup, the local linear models are adaptive by their construction, see section 2.2.3. For the LGBM, we retrain the model every week to adapt to new changes. In both cases, we use exponential weightings of the immediate past to handle the non-stationarity. In [42] they take a different approach and consider how to detect temporal covariance shift in the data and define  $K$  new sub-series of the data. As there could be a shift in the distribution of the test data, they find a model with good, invariant performance on all  $K$  series. Another possibility is to use changepoint analysis [43] to find changes in the volatility of the time series. Then we could make an adaptive time window and train only on the data from the latest change point with the same amount of volatility. The chosen direct strategy also limits our adaptability as we would adapt faster to market changes using data from all hours. The issue is that we would also increase the complexity of the model dramatically. One could also improve the handling of price spikes and train a model specifically for the price spikes and ignore them when we model the normal dynamics of the market.

### 7.2 Weak Learners and the Strength of an Ensemble

In table 6.1, the MAE for the probabilistic forecast using the median of the sorted ensemble using the result in [12] to obtain an MAE of 0.392 GBP/MWh for the LGBM. If we do not sort the ensemble and take out the  $\hat{q}^{(0.5)}$  directly, we obtain an MAE of 0.777 GBP/MWh for the LGBM. This demonstrates the importance of sorting and the strength of an ensemble of relatively weak learners. We set a high learning rate, allow few iterations, and regularize the trees. This makes the individual models for each quantile worse but strengthens the ensemble of quantiles. Each model is more likely to find different signals, which is better for the ensemble to generalize well. We could see this in our models using feature importance plots. There are general trends from low to high nominal levels, but the feature importance for adjacent nominal levels could be quite different, indicating that they pick up different signals.

### 7.3 Uncertainty of Predictions

When we generate the models, we take the predictions of power production and load as ground truth. These predictions could have a bias or be good in some periods and worse in others. A measure of uncertainty of the predictions would be beneficial as we could take this into account when we construct our models.

## 7.4 Feature Importance

The feature importance presented gives some interpretation of the most essential features to include in the model. If the feature importance should be studied in a more general, generic way, one should consider Shapley values [44] instead. There already exists libraries [45] that are built on top of LightGBM. We also encountered a problem of spurious feature importance due to the correlation between input features. To alleviate this, we use the ZCA-cor [36] transformation; however, this also reduces the information in the created features like the aggregated gas capacities because this feature is a direct linear combination of all the individual capacities.

## 7.5 New Strategies for Prediction of the Intermittency Risk

We model the marginal distribution of the hourly prices and then combine them to predict the daily intermittency. Instead, we could predict the daily intermittency directly to see if this strategy is better. We could also adopt different versions of the MIMO [28] to make fewer models. Instead of a separate model for each hour and quantile, we could make one model for each hour that outputs all the quantiles and use then CRPS as our loss function. One could also fix a nominal level  $q^{(\alpha_i)}$  and make one model that predicts this nominal level for all hours. The strategies would reduce the needed number of needed models and could be interesting to study further.

## 7.6 Hybrid Models

To improve the final model, one could make a hybrid model [8] that takes the prediction of the local linear model and use it as input to the LGBM, or make an ensemble of all model structures considered. We use a Gaussian copula which has the valuable property that it is not a complex black-box model and allows us to track the interdependence over time using the covariance matrix. However, we could also construct a non-linear model for the interdependence between hours that could perhaps capture more of the non-linear interdependency between hours.



## 8 | Conclusion

The intermittency risk is essential for renewable energy generators to construct efficient trading strategies. This project focuses on the daily intermittency risk three days ahead using the difference between the average N2EX day-ahead prices and the average production weighted prices. We focus on the marginal distribution of each hour and issue a baseline point forecast and ARIMA model for each hour. On the validation data, the point forecasts are mostly outperformed by the probabilistic forecasts of which we issue a baseline, a local linear model with and without exponential weights on the immediate past, and a LightGBM quantile forecast. On the validation data, the latter has the lowest MAE for the intermittency risk on 0.392 GBP/MWh, while the baseline has one of 0.692 GBP/MWh. Using a Gaussian copula, we can couple the hourly distribution and make a probabilistic forecast for the daily intermittency risk. Using this approach, the LightGBM obtains a CRPS of 0.306 GBP/MWh, which suggests the probabilistic forecast has more valuable information about the intermittency risk. We select the LightGBM due to its superior performance and test on test data where we get an MAE on 1.129 GBP/MWh and CRPS on 0.895 GBP/MWh. The test data has unprecedented price spikes, which we suspect is the main reason we see a worse performance. We propose new update mechanisms and modeling strategies to better adapt to market changes.

## References

- [1] E. Reid and S. Dingenen, “Corporate ppas: An international perspective - 2020/2021 edition,” 2021. [Online]. Available: <https://www.twobirds.com/en/news/articles/2018/global/bird-and-bird-and-corporate-ppas-an-international-perspective>
- [2] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.
- [3] S. Dr Haben and J. Dr Caudron, “Probabilistic day-ahead wholesale price forecasting,” 2021. [Online]. Available: <https://es.catapult.org.uk/reports/probabilistic-day-ahead-wholesale-price-forecasting/>
- [4] (2021) Wholesale market indicators. [Online]. Available: <https://www.ofgem.gov.uk/energy-data-and-research/data-portal/wholesale-market-indicators>
- [5] J. M. Morales, A. J. Conejo, H. Madsen, P. Pinson, and M. Zugno, “Impact of stochastic renewable energy generation on market quantities,” in *Integrating Renewables in Electricity Markets*. Springer, 2014, pp. 173–203.
- [6] Nord pool announces gb auction changes for brexit. [Online]. Available: <https://www.nordpoolgroup.com/message-center-container/newsroom/exchange-message-list/2020/q4/nord-pool-announces-gb-auction-changes-for-brexit/>
- [7] Curtailment, price thresholds and decoupling. [Online]. Available: <https://www.nordpoolgroup.com/trading/Day-ahead-trading/Curtailment-price-thresholds-and-decoupling/>
- [8] R. Weron, “Electricity price forecasting: A review of the state-of-the-art with a look into the future,” *International journal of forecasting*, vol. 30, no. 4, pp. 1030–1081, 2014.
- [9] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.
- [10] H. Madsen, *Time series analysis*. Chapman and Hall/CRC, 2007.
- [11] R. Koenker and G. Bassett Jr, “Regression quantiles,” *Econometrica: journal of the Econometric Society*, pp. 33–50, 1978.
- [12] V. Chernozhukov, I. Fernández-Val, and A. Galichon, “Quantile and probability curves without crossing,” *Econometrica*, vol. 78, no. 3, pp. 1093–1125, 2010.
- [13] H. Madsen and J. Holst, “Modelling non-linear and non-stationary time serie,” *IMM*, 2016.
- [14] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*. Springer New York Inc., 2017.
- [15] D. S. Rosenberg, “Lecture slides,” in *Foundations of Machine Learning*. Bloomberg, 2017. [Online]. Available: <https://bloomberg.github.io/foml/#home>
- [16] G. Seni and J. F. Elder, “Ensemble methods in data mining: improving accuracy through combining predictions,” *Synthesis lectures on data mining and knowledge discovery*, vol. 2, no. 1, pp. 1–126, 2010.

- [17] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Annals of statistics*, pp. 1189–1232, 2001.
- [18] Kaggle’s guide to lightgbm hyperparameter tuning with optuna in 2021. [Online]. Available: <https://towardsdatascience.com/kagglers-guide-to-lightgbm-hyperparameter-tuning-with-optuna-in-2021-ed048d9838b5>
- [19] Problem with quantile regression for lightgbm. [Online]. Available: <https://github.com/microsoft/LightGBM/issues/1109>
- [20] Lightgbm documentation. [Online]. Available: <https://lightgbm.readthedocs.io/en/latest/>
- [21] J. M. Morales, A. J. Conejo, H. Madsen, P. Pinson, and M. Zugno, *Renewable Energy Sources—Modeling and Forecasting*. Boston, MA: Springer US, 2014, pp. 15–56.
- [22] K. Pearson, “On a method of determining whether a sample of size  $n$  supposed to have been drawn from a parent population having a known probability integral has probably been drawn at random,” *Biometrika*, pp. 379–410, 1933.
- [23] J. Bröcker and L. A. Smith, “Increasing the reliability of reliability diagrams,” *Weather and forecasting*, vol. 22, no. 3, pp. 651–661, 2007.
- [24] P. Pinson, P. McSharry, and H. Madsen, “Reliability diagrams for non-parametric density forecasts of continuous variables: Accounting for serial correlation,” *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, vol. 136, no. 646, pp. 77–90, 2010.
- [25] T. Gneiting and A. E. Raftery, “Strictly proper scoring rules, prediction, and estimation,” *Journal of the American statistical Association*, vol. 102, no. 477, pp. 359–378, 2007.
- [26] H. Hersbach, “Decomposition of the continuous ranked probability score for ensemble prediction systems,” *Weather and Forecasting*, vol. 15, no. 5, pp. 559–570, 2000.
- [27] B. Jason. How to convert a time series to a supervised learning problem in python. [Online]. Available: <https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/>
- [28] G. Bontempi, S. B. Taieb, and Y.-A. Le Borgne, “Machine learning strategies for time series forecasting,” in *European business intelligence summer school*. Springer, 2012, pp. 62–77.
- [29] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.
- [30] F. Takens, “Detecting strange attractors in turbulence,” in *Dynamical systems and turbulence, Warwick 1980*. Springer, 1981, pp. 366–381.
- [31] M. Casdagli, S. Eubank, J. D. Farmer, and J. Gibson, “State space reconstruction in the presence of noise,” *Physica D: Nonlinear Phenomena*, vol. 51, no. 1-3, pp. 52–98, 1991.
- [32] S. B. Taieb, “Machine learning strategies for multi-step-ahead time series forecasting,” *Universit Libre de Bruxelles, Belgium*, pp. 75–86, 2014.

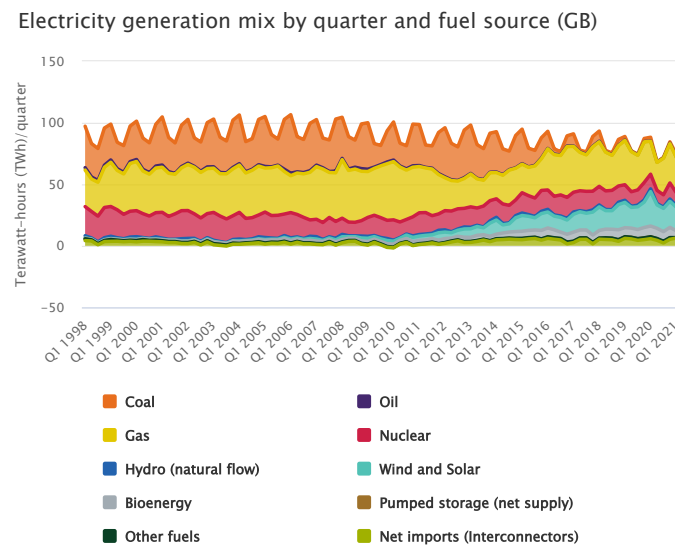
- [33] G. Bontempi, “Long term time series prediction with multi-input multi-output local learning,” *Proc. 2nd ESTSP*, pp. 145–154, 2008.
- [34] P. Pinson, H. Madsen, H. A. Nielsen, G. Papaefthymiou, and B. Klöckl, “From probabilistic forecasts to statistical scenarios of short-term wind power production,” *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 12, no. 1, pp. 51–62, 2009.
- [35] (2020) Time based cross validation. [Online]. Available: <https://towardsdatascience.com/time-based-cross-validation-d259b13d42b8>
- [36] A. Kessy, A. Lewin, and K. Strimmer, “Optimal whitening and decorrelation,” *The American Statistician*, vol. 72, no. 4, pp. 309–314, 2018.
- [37] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
- [38] Gradient boosting regression. [Online]. Available: [https://scikit-learn.org/stable/auto\\_examples/ensemble/plot\\_gradient\\_boosting\\_regression.html](https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_regression.html)
- [39] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.
- [40] P. Pinson, G. Kariniotakis, H. A. Nielsen, T. S. Nielsen, and H. Madsen, “Properties of quantile and interval forecasts of wind generation and their evaluation,” in *Proceedings of the European Wind Energy Conference & Exhibition, Athens*, 2006, pp. 1–10.
- [41] Coal finds a surprising 2020 bright spot in europe. [Online]. Available: <https://www.wsj.com/articles/coal-finds-a-surprising-2020-bright-spot-in-europe-11602763826>
- [42] Y. Du, J. Wang, W. Feng, S. Pan, T. Qin, R. Xu, and C. Wang, “Adarnn: Adaptive learning and forecasting of time series,” in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 402–411.
- [43] R. Killick, P. Fearnhead, and I. A. Eckley, “Optimal detection of changepoints with a linear computational cost,” *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2012.
- [44] A. E. Roth, *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.
- [45] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774.
- [46] (2021) Timeline of uk government coronavirus lockdowns. [Online]. Available: <https://www.instituteforgovernment.org.uk/charts/uk-government-coronavirus-lockdowns>
- [47] A. J. Bell and T. J. Sejnowski, “The “independent components” of natural scenes are edge filters,” *Vision research*, vol. 37, no. 23, pp. 3327–3338, 1997.

## Appendices

### A UK Power Market

In the following, we present a couple of figures that give some idea about the development of the UK power generation and market. For reference, a timeline of the UK lockdowns are provided to compare with the changes in the prediction of energy prices.

#### Electricity generation mix in the UK

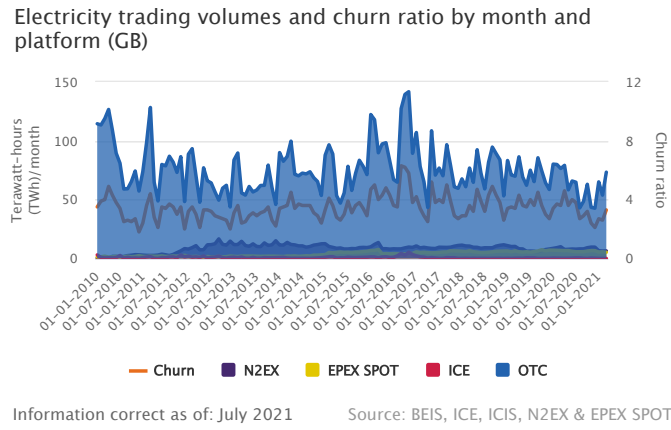


**Figure A.1** – UK power mix over time. Figure is from ofgem [4]

In figure A.1, we see just how important gas is for the market and how renewable energy sources make up an increasing amount of the energy mix.

#### Traded market volumes in the UK power market

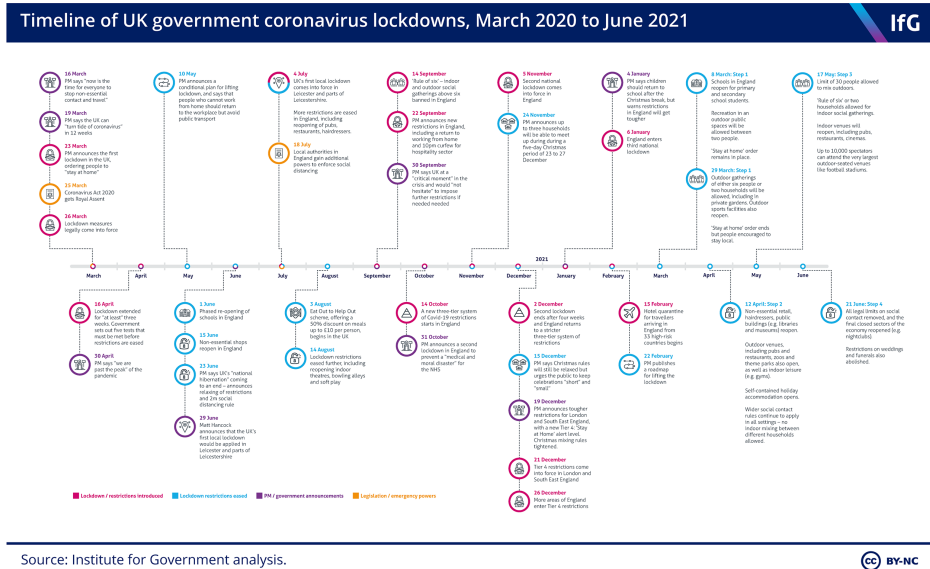
We first see that the largest volumes of electricity are traded over-the-counter, OTC, through a bilateral agreement where both parties are in direct contact.



**Figure A.2** – Volumes traded in the UK power market. Figure is from ofgem [4]

Note that the churn is a metric to assess market liquidity and is a proxy for how many times electricity is traded before it is finally delivered to end-customers. A large churn indicates a healthy, mature market as it allows all participants to finetune their positions and have confidence at the price they pay.

## Timeline of COVID-19 lockdowns in the UK



**Figure A.3** – Timeline of UK government coronavirus lockdowns. The figure is from Institute for Government Analysis [46]

## B Proof: Minimization of Pinball Loss Leads to Best Estimate of Quantile

We will show that if we minimize the conditional expectation of the loss of  $\ell_\alpha$ , then we find a best estimate. For convenience introduce the more generic  $x, y \in \mathbb{R}$ . We want to show:

$$q^\alpha = \arg \min_x \mathbb{E}_p[\ell_\alpha(y, x)] \quad (\text{B.1})$$

First, rewrite the pinball loss function 2.5 using indicator functions

$$\begin{aligned}\ell_\alpha(y, x) &= (\alpha - 1)(y - x) \mathbf{1}_{\{y < x\}} + \alpha(y - x) \mathbf{1}_{\{y \geq x\}} \\ &= \alpha(y - x) (\mathbf{1}_{\{y < x\}} + \mathbf{1}_{\{y \geq x\}}) - (y - x) \mathbf{1}_{\{y < x\}} \\ &= (y - x) [\alpha - \mathbf{1}_{\{y < x\}}]\end{aligned}\tag{B.2}$$

Consider using a decent algorithm where we try to minimize w.r.t.  $x$

$$\begin{aligned}\frac{d}{dx} \mathbb{E}_y [\ell_\alpha(Y, x)] &= \frac{d}{dx} \int \ell_\alpha(y, x) f(y) dy \\ &= \frac{d}{dx} \int (y - x) [\alpha - \mathbf{1}_{\{y < x\}}] f(y) dy \\ &= \int \frac{d}{dx} (y - x) [\alpha - \mathbf{1}_{\{y < x\}}] f(y) dy && \text{by Leibnitz} \\ &= \int_{-\infty}^{\infty} -1 [\alpha - \mathbf{1}_{\{y < x\}}] f(y) dy + \int_{-\infty}^{\infty} (y - x) (-\delta_{y=x}) f(y) dy \\ &= -\alpha \int_{-\infty}^{\infty} f(y) dy + \int_{-\infty}^{\infty} \mathbf{1}_{\{y < x\}} dy \\ &= -\alpha + \int_{-\infty}^x f(y) dy \\ &= -\alpha + P(Y \leq x)\end{aligned}\tag{B.3}$$

hence when  $\ell_\alpha \rightarrow 0$  then  $\frac{d}{dx} \mathbb{E}_y [\ell_\alpha(Y, x)] \rightarrow 0$  which means  $P(Y \leq x) \rightarrow \alpha$  and  $x \rightarrow q^\alpha$ . Note that in the above  $\int_{-\infty}^{\infty} (y - x) (-\delta_{y=x}) f(y) dy$  appears as  $\frac{d}{dx} \mathbf{1}_{\{y < x\}} = \delta(y - x)$  which we denote as  $\delta_{y=x}$ . The term  $\int_{-\infty}^{\infty} (y - x) (-\delta_{y=x}) f(y) dy$  disappears as  $\delta(y - x)$  is zero when  $x \neq y$  and when  $x = y$ , then  $(y - x) = 0$ .

## C Created Features

This section will briefly cover the features we created using the input features.

**Renewable Energy Penetration** The feature name is 'RES prod norm' for renewable energy sources production normalized. Let  $\psi$  denote the renewable energy penetration.

$$\psi = \frac{\sum_{i=1}^5 \gamma_i}{\kappa}\tag{C.1}$$

where  $\kappa$  is the demand forecast 'Demand forecast - ECR - UK' and  $\gamma_i$  are predictions of renewable energy:

- $\gamma_1$  is 'Embedded solar forecast - ECR - UK'
- $\gamma_2$  is 'Embedded wind forecast - ECR - UK'
- $\gamma_3$  is 'Solar forecast - ECR - UK'
- $\gamma_4$  is 'Wind forecast - ECR - UK'
- $\gamma_5$  is the hydro forecast, we define just below

This is the only input feature for the local linear model and is also an input features for the LGBM model.

**Hydro Production** Hydro energy is divided into two features in the dataset from *Ørsted*. One is an hourly hydro index, and the other is the installed hydro capacity that changes on a daily basis. To get the hourly hydro forecast, we take the product of the 'NPSHYD - Index - UK' and 'NPSHYD - Installed capacity - UK' and call it 'Hydro forecast'.

**Aggregated Available Gas Capacities** We have a couple of features with available gas capacities in the UK. We thought that the aggregated capacities could condense the information. We make the feature 'agg\_CCGT\_cap' which is the aggregated available capacities of the combined cycle gas turbine plants. Consult appendix E to find the specific names of the features. Likewise, we created 'agg\_OCGT\_cap' for the Open Cycle Gas Turbines.

## D Important Hyperparameters for LightGBM

We found that the following parameters were vital to keep inside ranges that they would not be in for other applications:

1. **Learning rate** should be higher than usual, around 0.3 to 0.4
2. **n\_estimators** is the number of iterations of the boosting and should be quite low, around 1000.
3. **num\_leaves** is the number of terminal nodes in figure 2.3b. We should enforce this to be above 500.
4. **regularizers** we use both **max\_depth**, **max\_bin**, and **lambda\_l2**. Other regularizers were considered but could cause problems. In the next subsection we will give one such example.

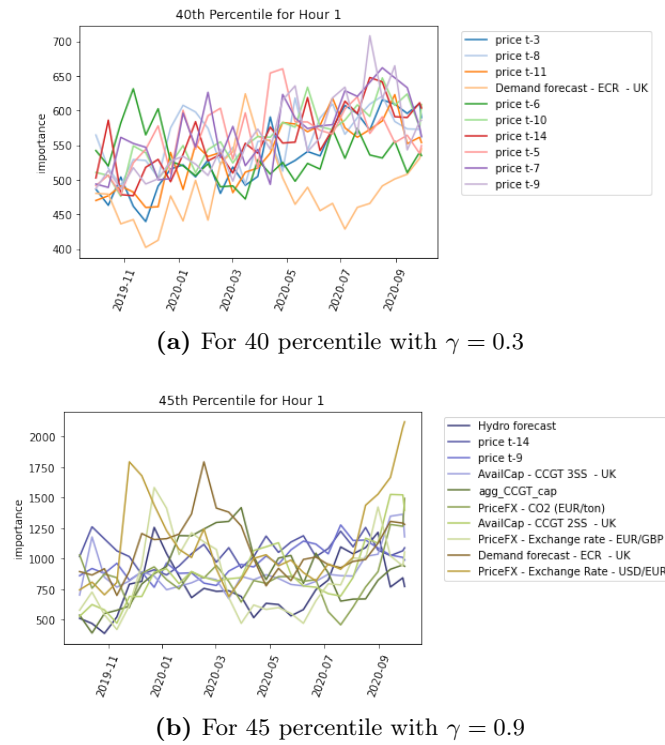
### Exclude feature fraction as regularizer

For gradient boosting methods, the feature fractions are often used as a regularizer to avoid overfitting [15]; however, with this data set with relatively little data, it turns out to have major drawbacks.

### The feature fraction parameter

According to the documentation [20], the feature fraction parameter we here will denote  $\gamma \in [0, 1]$  is the fraction of features included before training of a tree. Before training tree  $i$  with e.g.  $\gamma = 0.2$ , we randomly take out 20% of the features and use only them to construct the tree. In the dataset, we use lagged values of the spot prices as features, and these makeup around  $\approx 5\%$  of all features. These lagged values carry some of the same information about the process and if tree  $i$  only retrieves one of the lagged features, then that prices signal would be very important for that specific tree while others might not have that great importance. As the information gain is calculated on an aggregate basis over all trees, these lagged values will all show up as very important because each subtree shows them as more important while they may carry the same information. This happens persistently, as is very apparent in figure D.1a where we use a low  $\gamma$  and almost only the lagged values seem to be of interest. Then with almost all features included, much different information is displayed as seen in figure D.1b.





**Figure D.1** – Feature gain for hour 1 in rolling forecast period for 40th and 45th percentile

## E The Available Input Data

In this section, we present the names of the available input features. We have data both on hourly granularity and daily granularity. In this first table, we present the names with hourly granularity. We have changed some of the original names of the features to hide the data sources, which was a requirement from *Ørsted* trading support to use the data for this project.

### Feature Available on an Hourly Basis

Feature name	Description
BSUoS forecast UK	Forecast for value of energy charge in UK
Demand forecast - ECR - UK	Electricity demand forecast in UK
Embedded solar forecast - ECR - UK	Forecast for generation from behind the meter solar in UK
Embedded wind forecast - ECR - UK	Forecast for generation from behind the meter wind in UK
INTIRL - Index - UK	Interconnector flow to Ireland from UK
NPSHYD - Index - UK	Hydro power production in UK
Wind forecast - ECR - UK	Wind production forecast in UK
Solar forecast - ECR - UK	Solar production forecast in UK

**Table E.1** – Description of features we have on an hourly basis

## Data Available on a Daily Basis

### F ZCA-cor and the Implications for Feature Importance

Using tree-based models, one can obtain the *feature importance* which quantifies how useful any feature is for the model. An issue that can arise is if co-linearity is present in the features as this could display spurious feature importance. Therefore, we first whiten the features. Methods based on principal component analysis, *PCA*, could be of interest as they also reduce the dimension of the input features; however, this would also reduce the interpretability of the feature importance as these would now be in terms of the principal components. We will use methods based on ZCA which is for *zero-phase components analysis* [47]. We can create whitened variables that are maximally close to the original ones with this method

#### Whitening

Whitening or sometimes referred to as sphering essentially is a linear transformation that makes the input variables orthogonal. For our data matrix at time  $t$  with  $N_t$  observations and  $m$  features,  $\mathbf{X} \in M_{N_t \times m}(\mathbb{R})$  the goal is to find a whitening matrix,  $\mathbf{W} \in M_{N_t \times N_t}(\mathbb{R})$ , induced by a transformation with the desired properties such that we obtain the whitened data  $\mathbf{Z} \in M_{N_t \times m}(\mathbb{R})$ :

$$\mathbf{Z} = \mathbf{W}\mathbf{X} \quad (\text{F.1})$$

#### Preliminary definitions and concepts

Let  $\mathbf{x}_j \in \mathbb{R}^{N_t}$  be all observations of feature  $j \in [1, \dots, m]$  and let  $\mu_1, \dots, \mu_m$  be the means of each feature. Introduce the covariance matrix  $\Sigma \in M_{m \times m}(\mathbb{R})$  where  $\sigma_{i,j} = (\mathbf{x}_i - \mu_i)^\top (\mathbf{x}_j - \mu_j)$  for  $i, j = 0, \dots, m$ .

**The mean centered data** is the data subtracted by the mean of the given feature:

$$x_{i,j}^c = x_{i,j} - \mu_j \quad \text{for } i, j = 1, \dots, d \quad (\text{F.2})$$

such that now  $\mu_j^c = 0$  for all  $i$ .

**The standardized data** is where we scale the data to have unit variance. For a given row  $\mathbf{x}_i \in \mathbb{R}^m$  of the datamatrix and with  $\mathbf{V} = \text{diag}(\sigma_{1,1}^2, \dots, \sigma_{m,m}^2)$  the standardized row is:

$$\mathbf{x}_i^s = \mathbf{V}^{-1/2} \mathbf{x}_i \quad (\text{F.3})$$

such that now  $\sigma_i^s = 1$  for all  $i = 1, \dots, m$ . This makes the data have unit variance but will of course not remove any correlation.

**The correlation matrix**  $\mathbf{P} \in M_{m \times m}(\{x \in \mathbb{R} | x \in [-1, 1]\})$  can be defined directly from the  $\mathbf{V}$  and covariance matrix  $\Sigma$  [36]

$$\mathbf{P} = \mathbf{V}^{-1/2} \Sigma \mathbf{V}^{-1/2} \quad (\text{F.4})$$

### ZCA-Mahalanobis whitening

Here we want to find a that whitens the features, is unique, and maximizes the similarity between the whitened  $z_{i,j}$  and the original  $x_{i,j}$  input. The ZCA-whitening maximizes the average cross-covariance between  $\mathbf{Z}$  and  $\mathbf{X}$  by minimizing the squared distance between  $z_{i,j}^c$  and  $x_{i,j}^c$  [36]. In [36] they show that the whitening matrix  $\mathbf{W}^{ZCA}$  will be the induced matrix of the ZCA-whitening:

$$\mathbf{W}^{ZCA} = \Sigma^{-1/2} \quad (\text{F.5})$$

### ZCA-cor whitening

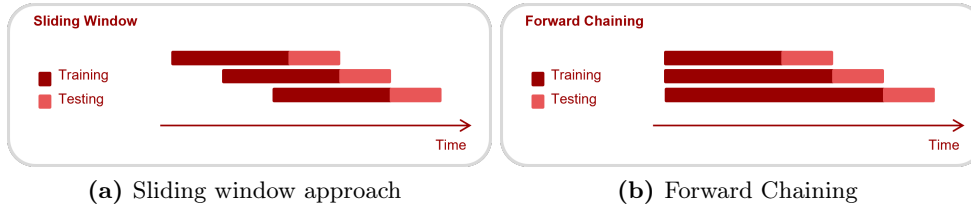
In the ZCA-Mahalanobis whitening, we use cross-covariance as a similarity measure, which means that the individual variance of the features is still a determining factor in the optimality criterion. An alternative scale-invariant version can be obtained by minimizing the squared distance between the centered, standardized features,  $X^{c,s}$ , and the whitened features,  $Z$ . In [36] the desired whitening matrix is denoted  $\mathbf{W}^{ZCA-cor}$  and is defined by

$$\mathbf{W}^{ZCA-cor} = \mathbf{P}^{-1/2} \mathbf{V}^{-1/2} \quad (\text{F.6})$$

This will whiten the features and maximize the correlation with the original variables [36]. These are properties we want for the feature transformation.

## G Time-base cross validation

In figure G.1, we see the two different cross validation



**Figure G.1** – Different time-based cross validation techniques

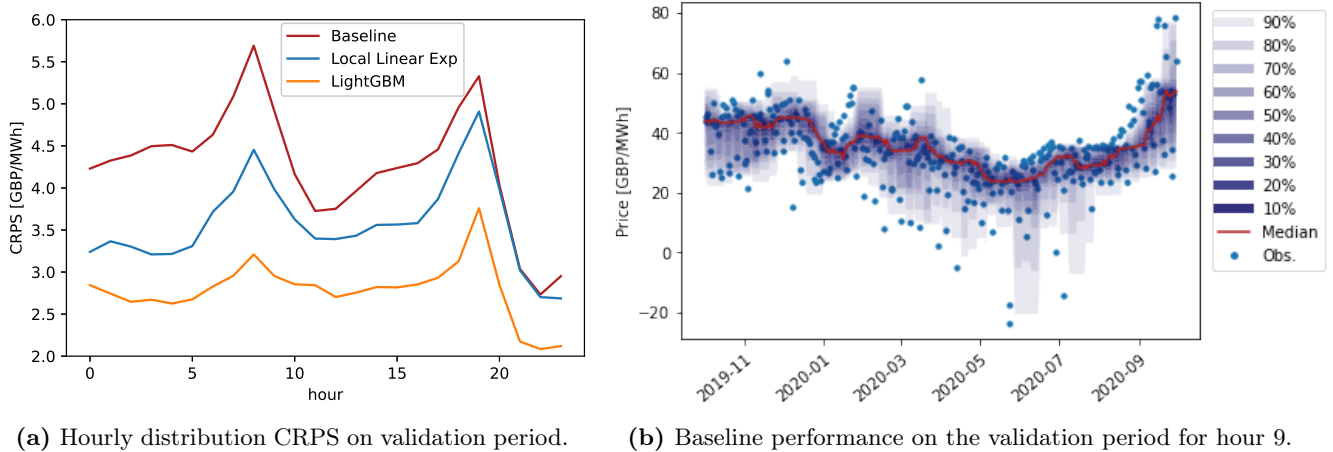
With *forward chaining* we assume that the model will improve as it has more data to train on. This assumption only holds if the series remains stationary. If that assumption does not hold, then we would have to construct a much more complex model to capture shifts in the distribution of the data [15]. On the other hand, using the *sliding window* we think the most valuable training examples are in the immediate past. We will not capture long-term trends or seasonality; however, it makes the assumption of local stationarity [13] more valid. This means we can fit a simpler, more parsimonious model [14].

## H Supplementary Figures to Assess Model Performance on Validation Data

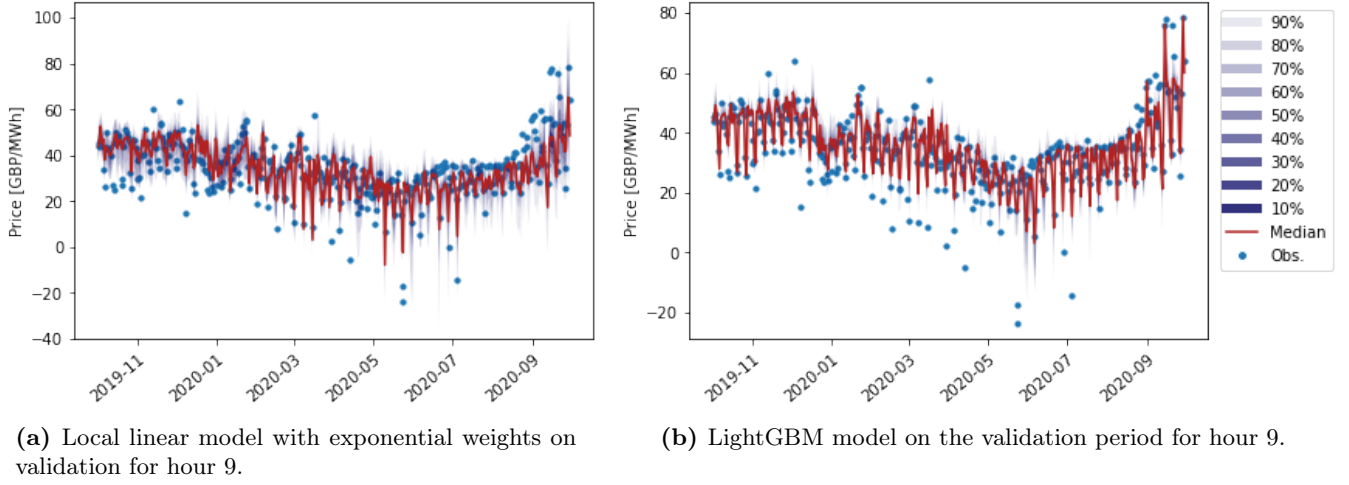
This section provides a couple of extra plots of the results.

## Results For the Validation Period

This section provides some figures to graphically see the performance of the models over the validation period. In figure H.1a we see the CRPS for each hour. We see that for all models, the overall pattern is the same with a bimodal curve on hour 8 and 18 with very low CRPS scores in the very last hours of the day. Compare this with the distribution of the hourly N2EX prices in figure 1.1a. When the spread is high with many outliers, the CRPS is high. In figure H.1b, we see that the baseline is very local and the nominal coverage rates are very susceptible prices far from the median. The case is not the same for the local linear model with exponential weights in figure H.2a that seems unable to cover all the points, especially points above the median. On the other hand, the LGBM in figure H.2b is better to capture high prices but is not as good to capture very low prices. This corresponds to the observations we made about the PIT histograms. The nominal coverage rates in figure H.2 can be hard to see for some of the models, but the median can be tracked easily.



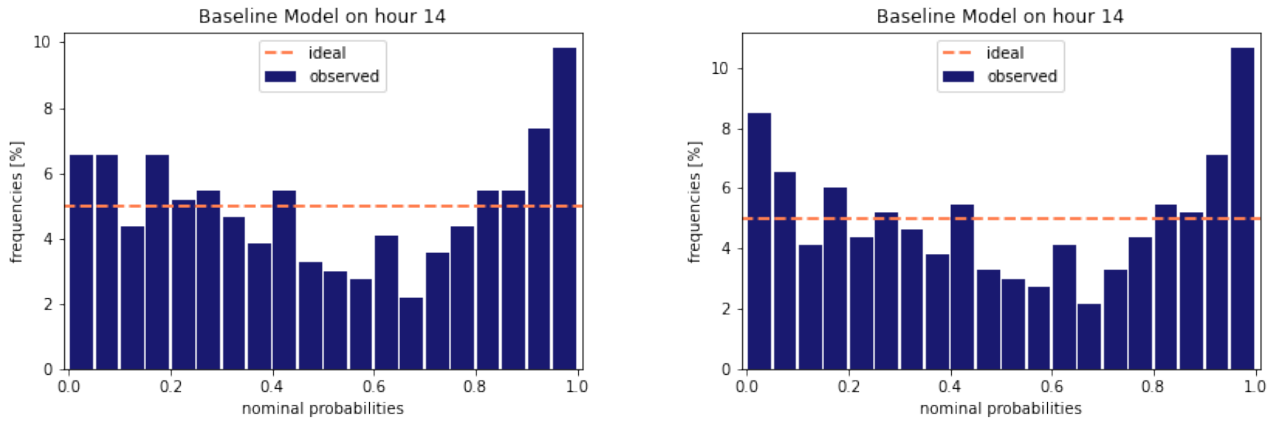
**Figure H.1** – In figure H.1a, we can see the CRPS scores for each hour of the day on the validation data. Figure H.1b shows the performance of the probabilistic baseline on the validation data for hour 9.



**Figure H.2** – The performance of local linear model with exponential weights and the LightGBM for hour 9 on the validation data.

## I Changes in the PIT After a Redefinition $\hat{q}^{(0)}$ and $\hat{q}^{(1)}$

For the baseline in figure I.1, we see some slight modifications; however, as this model was already reasonably calibrated, the changes are small as expected.

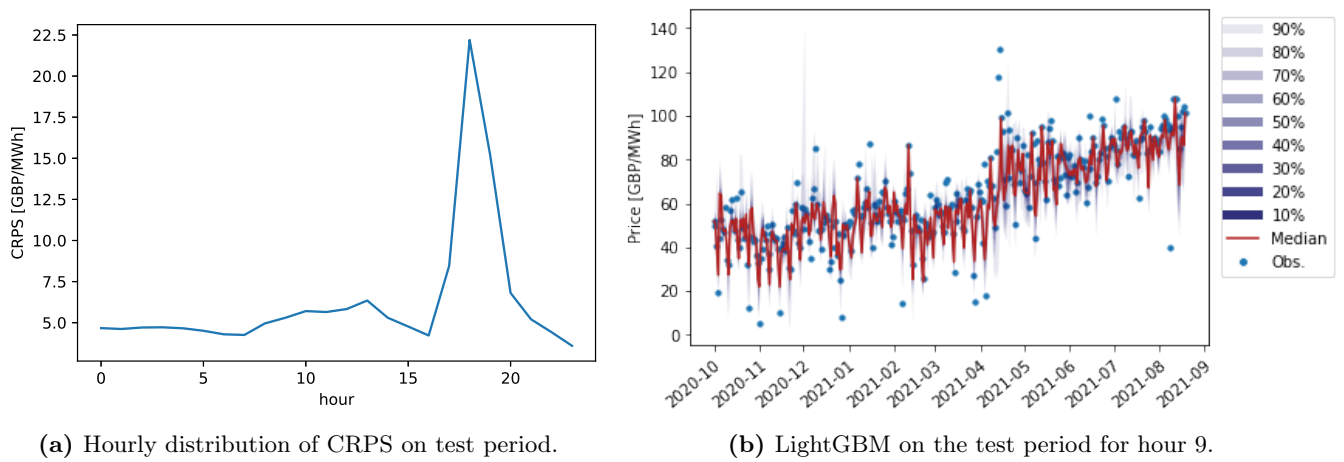


**Figure I.1** – PIT histograms for the baseline model before and after we changed the definition of  $\hat{q}^{(0)}$  and  $\hat{q}^{(1)}$ .

## J Supplementary Figures to Assess Model Performance on Test Data

In this section, we include a couple of supplementary figures to assess the performance on the test data. In figure J.1a we see that the CRPS has increased for all hours when we compare with figure H.1a, but as anticipated, hour 18 turns out to be the hardest hour to predict with a CRPS score above 20. Figure J.1b displays the predictions for the entire period for hour 9. Compare this figure with H.2b with the same hour last year. We see that the prices of 2021 are at a higher level with no negative prices but high positive price

spikes that the model cannot always capture.



**Figure J.1** – In figure J.1a, we see daily distribution of the CRPS scores on the test data. Figure J.1b shows the performance of the LightGBM on the test data for hour 9.