Quantitative Forecasting & Power Markets

**Devanshu Sharma**
RWTH Aachen University
October 2025

*This report develops a data-driven statistical model for constructing an Hourly Price Forward Curve (HPFC) for the German power market. Using historical day-ahead prices and exogenous indicators, the study forecasts hourly price dynamics for 2024 and scales them to align with an assumed forward market price (Cal-27).*

# 1 Background

Forward contracts are one of the most important means of trading in power markets. Forward contracts are pre-determined agreements for buying or selling electricity at a fixed time in the future. They are similar to futures contracts; however, there is a key difference. Futures contracts are traded on an energy exchange (such as EEX or NYMEX) with a standardized format. Forward contracts, on the other hand, are traded via bilateral agreement through over-the-counter (OTC) trading. This leads to the highly customizable property of forward contracts.

Forward curves are used to visualize the prices of forward contracts corresponding to a certain time in the future. A forward curve shows the price of a future delivery forward contract on a specific day (for example, today). In other words, a forward curve shows what the market believes will be the price of electricity at some point in the future. An Hourly Forward Price Curve (HPFC) is a forward price curve with an hourly frequency.

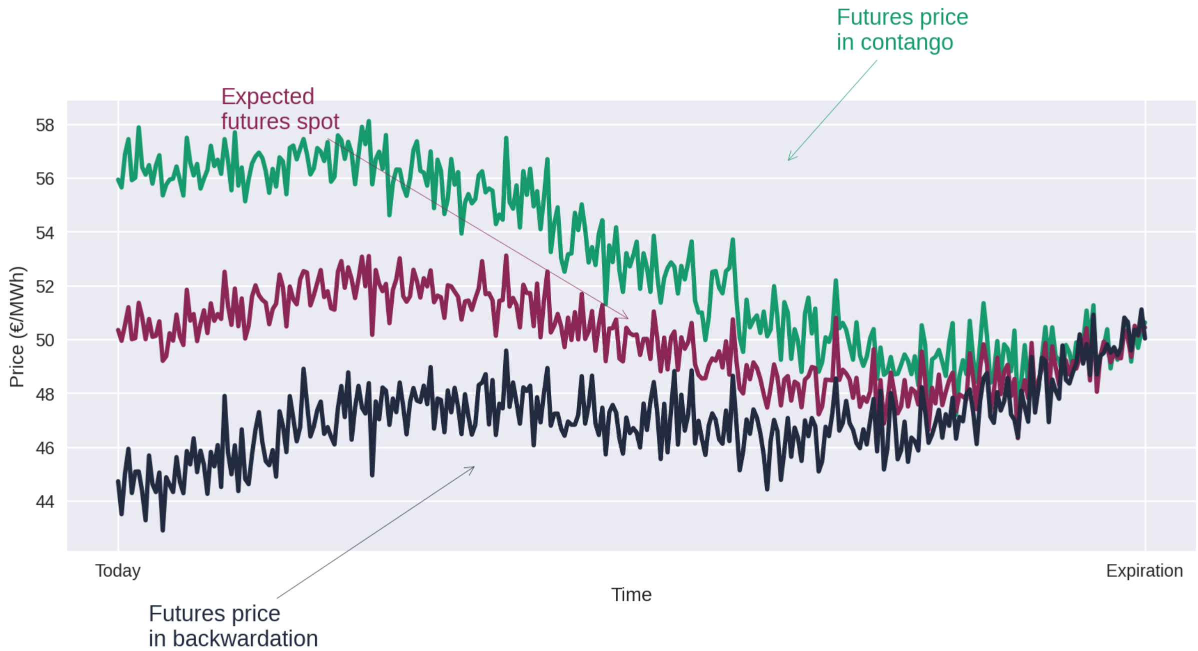



Figure 1: A graphical representation of `Contango` vs `Backwardation` trends in the forward market. As the time of delivery approaches, both curves converge to the spot price at delivery. Note: The plot is not meant to be scaled.

Whenever the electricity prices are expected to increase in the future, the prices of forward contracts become higher than the expected spot prices in the future. This trend is referred to as `Contango`. On the contrary, if prices are expected to fall in the future, the forward contracts become cheaper than the expected future spot prices. Such a trend is called `Backwardation`. We show a visualization of these trends in fig. 1

The primary objective of Market Risk Modelling (MRM) is to identify, quantify, and mitigate potential financial losses arising from adverse market movements. MRM is fundamental to several critical business functions, including portfolio optimization, strategic capital allocation, and the development of robust trading strategies.

A key application within MRM is forward price structuring, which involves constructing forward price curves that project electricity prices hourly, often for periods extending 5 to 15 years into the future. For power companies, forward contracts locked in at these prices represent a guaranteed revenue stream and serve as a vital instrument for hedging against spot price volatility. The price of a forward contract inherently reflects the market's collective sentiment and the perceived risk over the contract's duration, making its accurate structuring a core component of effective market risk management.

For the sake of clarity and brevity in the following sections, the provided dataset of historical hourly electricity prices from the German market will be referred to simply as `data`.

## 2 Data Exploration & Visualizing Patterns

While data can exhibit a complex and seemingly infinite range of patterns, a robust analysis hinges on identifying and focusing on the most economically significant and recurring structures. Based on this understanding, our exploration centers on the following fundamental patterns, each with a direct link to underlying physical and economic drivers:

- Daily patterns (hour-based price frequency),
- Weekly patterns (day-based price frequency),
- Seasonal patterns (month-based price frequency),
- Crisis pattern (year-on-year comparison).

Since electricity is a commodity whose price is governed by its supply (generation and transmission) and demand (consumption), the above patterns can be directly attributed to physical phenomena/events, as we shall see below. Key summary statistics are presented below:

- Row count: 42824, accounting for $(24 * 365 * 5 + 1)$ hours
- Mean: 99.084687 €/MWh
- Standard Deviation: 105.559683 €/MWh
- Global Minima: -500.000000 €/MWh
- Global Maxima: 871.00000 €/MWh

### 2.1 Daily Patterns

The hourly averaged prices are shown in fig. 2. Analyzing data at this granularity reveals insights into the collective rhythm of daily human activity and its impact on electricity markets. The four subplots illustrate the following trends of this daily cycle:

- **Top Left Plot**: The curve has two global maxima inside the peak hour window (08:00 to 20:00), which shows that electricity prices spike in this time period due to enhanced demand and low solar radiation (the suppliers have to borrow power from more expensive sources). The drop in the average price at noon (12:00 to 15:00) is due to large solar radiation producing surplus electricity. Outside the peak hours (00:00 to 08:00 and 20:00 to 00:00), the price drops rapidly due to lower demand. This is also called a "Duck Curve".
- **Top Right Plot**: This figure shows the spread of prices at different hours of the day. The means of this spread are already explained in the top left curve. The distribution seems mostly concentrated towards the lower price cap with occasional spikes. Interestingly, the prices sometimes become negative during the noon hours, thanks to excess supply from the sun. Overall, there is an oscillatory pattern, resembling a sine wave, for the full day.
- **Bottom Left Plot**: This two-dimensional heatmap shows price variation as a function of both hour of the day and the day itself. The prices are higher on weekdays compared to

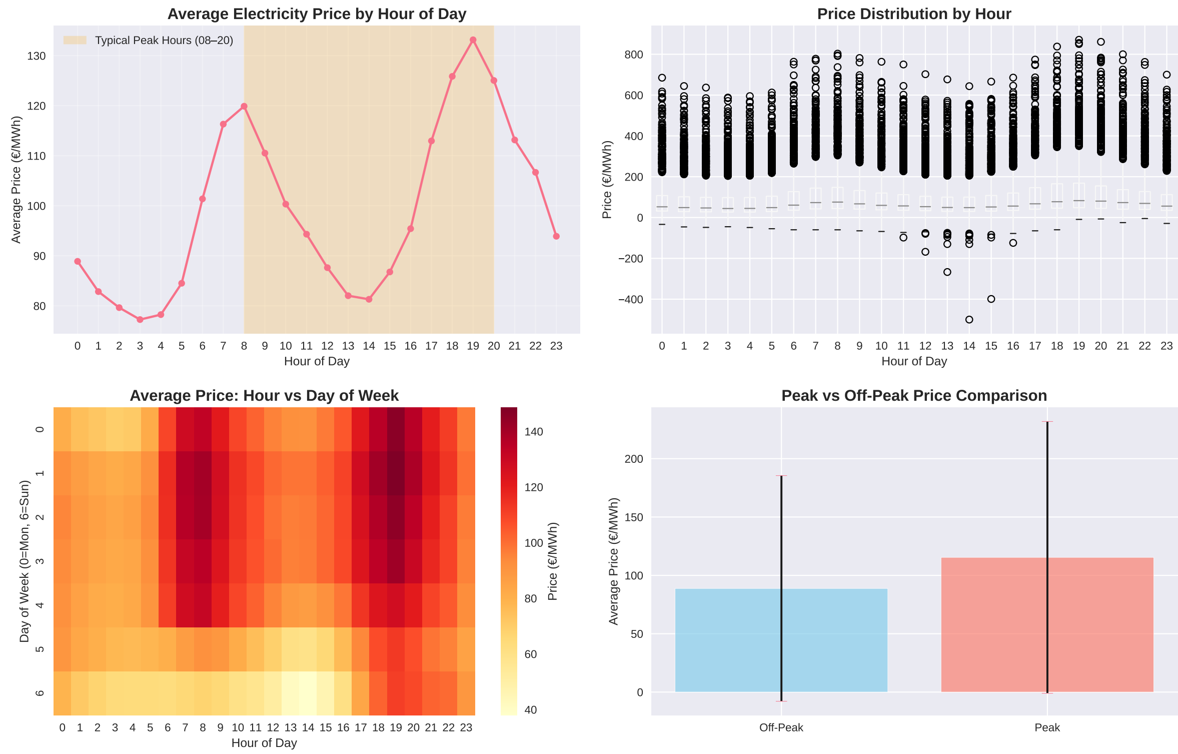



Figure 2: Pattern for the daily chart. Peak hour: 19:00 (133.17 €/MWh), Off-peak hour: 3:00 (77.25 €/MWh), Weekday: More Consumption, Weekend: Less Consumption

the weekends. The two global maxima (from the top left curve) are also visible as two dark red spots.

- **Bottom Right Plot**: This bar graph shows the average price during peak hours vs off-peak hours. Both the mean price and volatility are substantially higher during peak hours. Note that here, off-peak hours also include full weekend periods.

## 2.2 Weekly Patterns

If one zooms out the time frame from an hourly window to a daily one, one can get insights about weekly activities and patterns, as given in fig. 3. Again, there are four subplots:

- **Top Left Plot**: This bar chart shows the average price for each day of the week. The data clearly reflects typical weekly routines, with prices lowest on weekends due to reduced commercial and industrial activity. Prices peak on Tuesday and gradually decline over the remainder of the week.
- **Top Right Plot**: This plot compares average prices and their volatility (shown by the black vertical lines) between weekdays and weekends. It confirms that both the average price and price volatility are significantly higher on weekdays.
- **Bottom Left Plot**: This heatmap shows price variation across all weeks of the year. Two periods of consistently high prices are visible: a primary one in August (Weeks 33–37) and a secondary spike during November and December (Weeks 47–51). The drivers behind these seasonal trends will be explored in the following section.
- **Bottom Right Plot**: Shows the probability distribution (PDF) of the prices for weekdays vs weekends. Interestingly, the PDFs are highly non-Gaussian with a heavy tail on the

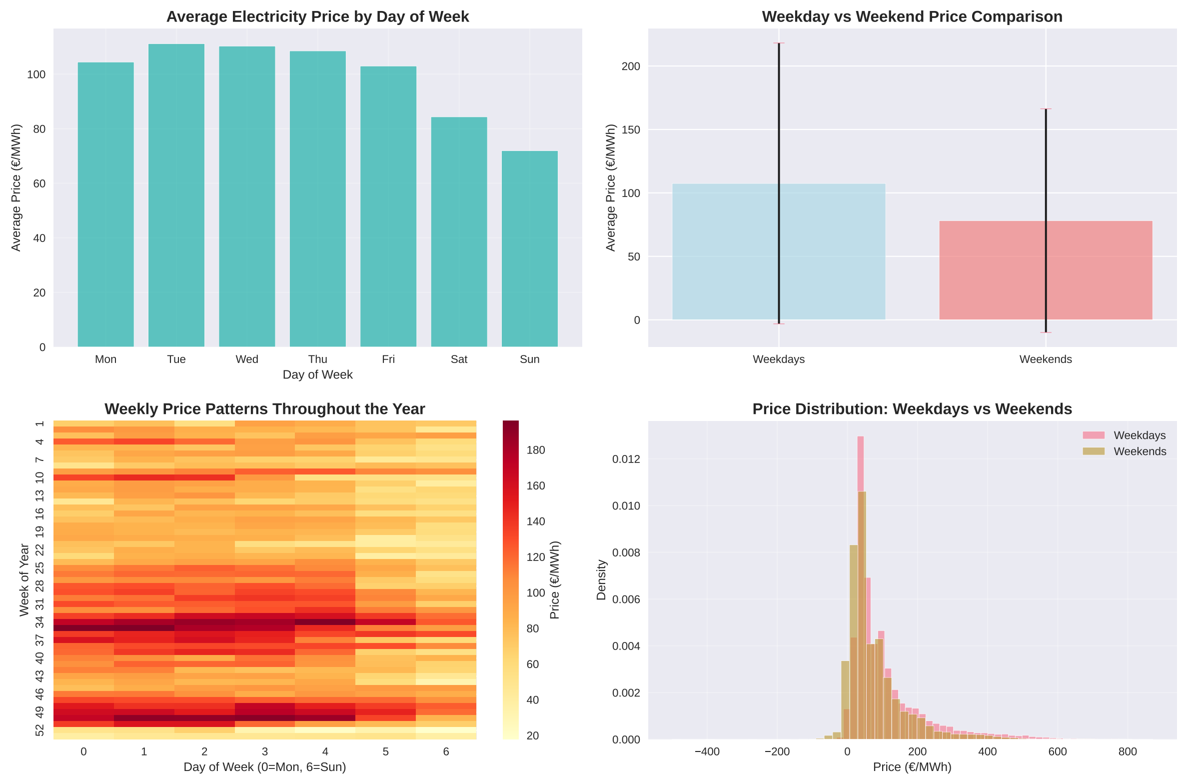



Figure 3: The pattern for the weekly chart.

positive side (positively skewed). The mean is substantially higher for the weekday PDF. This statistical analysis deserves another report, but we shall reserve this for future studies. Overall, the heavy tails imply that the probability of higher prices is enhanced owing to several uncertainties associated with the demand and supply of electricity.

## 2.3 Seasonal Patterns

While the daily and weekly patterns highlight the human activities, the impact of climate can be studied by further zooming out the time frame to create monthly windows and study the seasonal patterns. The four subplots shown in fig. 4 express these patterns, such that:

- **Top Left Plot**: Shows the average monthly price as a function of the months. The prices are highest during the August-September period, owing to the extreme heat and the need for cooling appliances. The prices stay low during the January-May period due to high wind power and lower demand.
- **Top Right Plot**: The seasonal average prices are shown here. As expected, the prices are highest during summer due to increased demand for cooling. This is followed closely by autumn, then winter, and finally spring. The water levels remain low during autumn, which could result in lower hydropower and high demand. During winter, there is a period of “Dunkelflaute” when the wind and solar power run low, and the days are cloudy. Overall, the power generation is low; hence, the prices tend to be slightly high.
- **Bottom Left Plot**: Shows the distribution of prices across the seasons. The highest variation is observed during summer, thanks to the combination of large solar power generation and high demand. On extremely hot days, the prices also become negative (excess supply). This behaviour is also noticed in the spring season, albeit on a smaller scale, due to high renewable generation. The distribution is primarily concentrated during


Figure 4: The pattern for the seasonal chart. Most expensive month: August (142.78 €/MWh), Least expensive month: May (73.60 €/MWh), Most volatile month: August ($\sigma = 178.48$ €/MWh), Least volatile month: February ($\sigma = 57.09$ €/MWh).

the winter and autumn seasons.

- **Bottom Right Plot**: The monthy standard deviation of prices is shown. The high volatility in the period from July to September is attributed to fluctuations in renewable energy and heatwaves. There is also a high volatility in December, mainly driven by large demand due to the festive season and Dunkelflaute.

## 2.4 Crisis Pattern

This section examines the impact of the Russia-Ukraine conflict, which began in February 2022, on European power markets. This geopolitical event had severe consequences, primarily affecting the Title Transfer Facility (TTF), the main hub for natural gas trading in the European Union. Damage to the Nord Stream pipelines, critical for supplying gas to Germany, further exacerbated the situation.

In electricity markets, gas-fired power plants typically have a high marginal cost, placing them last in the merit order for dispatch. They are only utilized when cheaper renewable sources are fully exhausted. Consequently, the TTF gas price often sets the marginal cost of electricity. The crisis caused TTF gas prices to skyrocket, which in turn drove electricity prices to unprecedented levels, as illustrated in Figure 5:

- **Top Left Plot**: The full timeline of hourly prices shows a rapid and sustained price increase beginning in early 2022, coinciding with the start of the crisis. Prices remained elevated until early 2023.
- **Top Right Plot**: This bar chart of average yearly prices, including their volatility (black

lines), clearly shows the crisis's impact. Both the mean price and its volatility increased dramatically in 2022.

- **Bottom Left Plot**: A direct comparison of monthly prices highlights the distinct signature of the crisis in 2022. The price surge was characterized by a very steep and narrow peak, lasting from May to October. By August 2022, prices had reached nearly four times their level from the previous year[1].
- **Bottom Right Plot**: This plot shows the year-over-year price growth percentage. Consistently large peaks throughout 2022 demonstrate the profound and sustained impact of the crisis. This is followed by a period of negative growth in early 2023, indicating a gradual easing of prices starting from that point.


Figure 5: The pattern for the Russia-Ukraine Crisis.

## 3 Model architecture

ARIMA (AutoRegressive Integrated Moving Average) is a widely used model for forecasting time series data. It combines three key components: AR (AutoRegression), I (Differencing), and MA (Moving Average). The AR part models the non-Markovian nature of the current observation. In other words, the value of the current observation is assumed to be proportional to its values in the past, $p$ steps back in time, where $p$ is the AR parameter. The differencing (I) takes care of converting a non-stationary time series to a stationary one. The differencing parameter $d$ represents the number of time differencing is carried out to obtain a stationary TSD. Finally, the moving average (MA) part is responsible for storing the impact of errors in the past observations on the current observation. The MA parameter $q$ tells how much in the past one needs to go to accurately model the impact of the collective errors.

[1]The year-on-year increase in 2021 was largely due to a rebound in energy demand following the economic shutdowns of the Covid-19 pandemic.

SARIMA (Seasonal ARIMA) extends the ARIMA framework to capture seasonal patterns, which are common in many real-world datasets like electricity prices, temperature, and sales figures. In addition to the standard ARIMA parameters {p, d, q}, SARIMA brings its own set of seasonal order parameters, viz. {P, D, Q, s}.

SARIMAX (Seasonal ARIMA with eXogenous regressors) is a further upgrade of SARIMA, which includes additional variables (exogenous variables) on top of SARIMA's architecture. These eXo variables are crucial for modelling any ambiguous circumstances that may influence the seasonal patterns in the data in a specific way. For example, the electricity demand is low during peak hours or holidays, resulting in a short-term deviation from the seasonal behaviour. Such pecularity can be modelled by regressing an eXo variable to the SARIMA model [1]. A detailed comparison between ARIMA, SARIMA, and SARIMAX is provided in table 1. The ultimate goal in ARIMA-based models is to find the coefficients of the involved hyperparameters and eXo variables with the help of a training dataset, and once these coefficients are known, use them to make predictions.

We choose the SARIMAX model for two primary reasons:

- Electricity prices exhibit strong seasonality driven by climatic conditions and human activity cycles.
- Specific factors, such as weekends, crisis periods, and night hours, act as significant external drivers that are best incorporated as additional variables.

| Model | Core Features | Upgrade Over Previous | When to Use |
|---|---|---|---|
| ARIMA | Captures trend, autocorrelation; handles non-stationarity; no seasonality or external variables | Baseline for time series with trend and autocorrelation | Data with trend, but no seasonality or exogenous factors |
| SARIMA | All ARIMA features; adds explicit seasonality (seasonal AR, I, MA terms); still no exogenous variables | Adds ability to model seasonal patterns | Data with regular cycles (e.g., hourly, daily, yearly), but no external regressors |
| SARIMAX | All SARIMA features; adds exogenous regressors (external variables); models both seasonality and outside influences | Adds ability to include external factors (e.g., weather, holidays) | Data with seasonality and external influences affecting the target variable |

Table 1: A comparison of salient features and hierarchy of ARIMA, SARIMA, and SARIMAX models for time series data.

### 3.1 Stationarity Check

ARIMA-based forecasting models require the underlying time series data to be stationary. A stationary series is one whose statistical properties, such as its mean and variance, remain constant over time. If these properties change with time, the series is considered non-stationary.

The necessity for stationarity arises from the model's underlying assumptions. In a non-stationary series, fundamental relationships like autocorrelation and partial autocorrelation become time-dependent. This instability means that the model's parameters (such as the AR and MA coefficients) would not be consistent throughout the series, leading to unreliable and inaccurate forecasts.

### 3.2 Best-fit parameters

The SARIMAX model is defined by seven hyperparameters: $p, d, q, P, D, Q, s$. To achieve the highest forecast accuracy, it is essential to find the optimal combination of these parameters.

Manually testing all possible combinations to identify the best-fit model is impractical. To automate this process, we employ a `GridSearch` algorithm. This method systematically constructs a grid of all possible hyperparameter combinations within predefined ranges. Each resulting model is then evaluated based on the Akaike Information Criterion (AIC) score [2]. The model with the lowest AIC score is selected as the optimal configuration for our forecasts.

After performing the `GridSearch`, we find the following best-fit parameters:

$$\texttt{order} = \{p, d, q\} = \{1, 0, 3\},\ \texttt{seasonal\_order} = \{P, D, Q, s\} = \{2, 1, 5, 24\}\ ,$$

The seasonal period parameter, $s$, defines the length of one seasonal cycle in the data. Common values include $s = 4$, $12$, $24$, or $168$, corresponding to quarterly, monthly, daily, and weekly cycles in hourly data. For this analysis, we set $s = 24$ to explicitly model the daily seasonality present in the hourly electricity prices.

Selecting the optimal set of hyperparameters for the backtesting procedure proved computationally prohibitive due to time constraints and limited computational resources. Higher values for the hyperparameters increase the model's complexity by requiring it to identify patterns from a more distant past. This significantly increases the computational cost through high-dimensional matrix operations, slowing the entire backtesting pipeline.

Therefore, we selected a parameter set that provides a practical balance between computational speed and forecast accuracy, which is:

$$\texttt{order} = \{p, d, q\} = \{1, 0, 2\},\ \texttt{seasonal\_order} = \{P, D, Q, s\} = \{1, 1, 1, 24\}\ .$$

On a final note, one could find more optimal parameters by expanding the domain of `GridSearch` or by using Bayesian optimization.

### 3.3 Exogenous Variables

As mentioned earlier, the Exo variables work by informing the model that there are more indicators in the data than just its seasonality. In principle, there could be infinite external factors influencing the electricity prices, but we have to choose the ones whose effects are most significant. After doing a literature review about renewable energy power plants, climatic factors and statistics of human activity, we proceeded to include the following Exo variables in our analysis:

- Peak hours & Off-peak hours
- Weekdays and weekends
- Historical natural gas prices traded at TTF
- Sinusoidal variable corresponding to the hour of the day (see top right figure in fig. 2)
- Flag indicating the Crisis phase from February 2022 to December 2022 (see section 2.4)

### 3.4 Model Workflow

The workflow is straightforward but requires significant hyperparameter tuning, parallelization, and memory usage: Clean & preprocess the TSD $\rightarrow$ Perform the stationarity check $\rightarrow$ Perform differencing if non-stationary TSD, otherwise, skip this step $\rightarrow$ Generate eXo variables $\rightarrow$ Perform GridSearch & find optimal hyperparameters $\rightarrow$ Create the SARIMAX model $\rightarrow$ Select a backtesting model $\rightarrow$ Train and fit the model based on optimal parameters and eXo variables $\rightarrow$ Forecast results $\rightarrow$ Evaluate performance metrics (MSE, MAE).

## 4 Backtesting The Model

Backtesting forms a crucial part of time series data (TSD) analysis. Once trained on the training dataset, the model needs to be evaluated on the basis of how well it predicts the data outside its training. This "outside" data is usually seen (known) to the user, but unseen to the model. The predicted values are then compared to the actual values, and the difference is quantified using the evaluation metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE). Backtesting thus provides us with the assessment of the model's accuracy and its reliability to predict actual future data (which is unknown, even to the user).

It is utmost important to preserve the temporal order of the TSD while backtesting the model. This means that one can only backtest the model that was trained over the past dataset. Furthermore, an ideal backtesting strategy should interpret the trends and seasonality of the data. The most common backtesting strategies are:

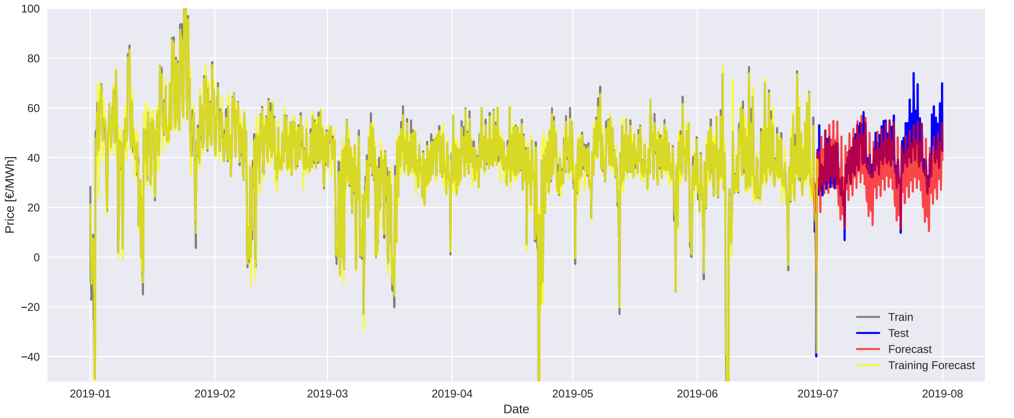
- **Static train-test split**: This is the simplest and quickest strategy. It works by (single) splitting the full data into training data and test data. Usually, the ratio of training data and test data is 80:20. However, it is sensitive to volatility and cannot capture the patterns for the full data.
- **Expanding Window**: In this strategy, the training window keeps growing gradually. The next period outside the training set is forecasted in every step. For example, start with a training set from January to March and forecast the April results; next, train from January to April and forecast the May results, and so on. Since this strategy uses all the data in training, it is useful for seasonally patterned data with non-stationarity.
- **Seasonal Window**: This strategy uses the blocks of data containing one full period of the season (for example, one-year blocks). This leads to the full seasonal cycles being fed to the model at a time, hence accurately accounting for the pattern.
- **Sliding Window**: In this strategy, the training set is a fixed time window (for example, 6 months), and so is the test set (for example, 1 month). Backtesting occurs on a sliding basis; the training and test sets shift forward in time but remain fixed in size. This strategy is faster and less memory-intensive compared to the expanding strategy, and the reasons we shall see below.

We have backtested the model using a train-test split, an expanding window, and a sliding window. The key features are highlighted below.

### 4.1 Static Train-Test Split

### 4.2 Expanding Window

In the expanding window backtesting, we start with an initial training set of a two-month window (January 01, 2019, to February 28, 2019) and the corresponding initial test set with a 15-day window (March 01, 2019, to March 07, 2019). In the next step, the training window expands by



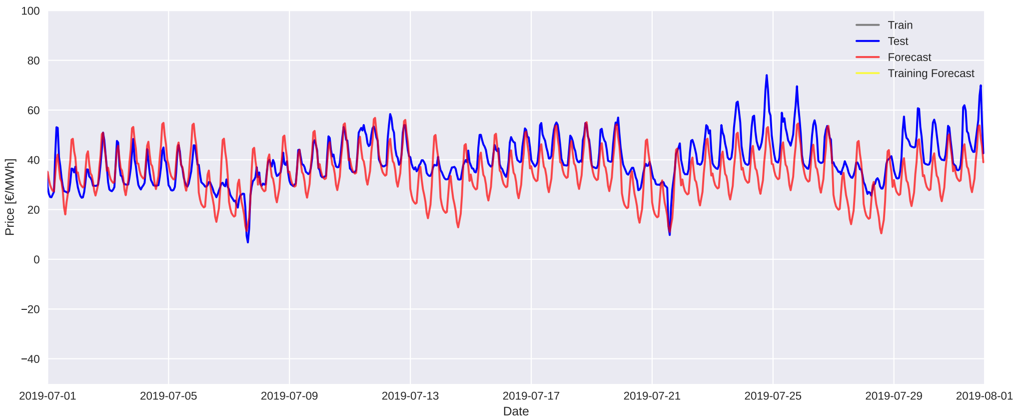




Figure 6: **Top Panel**: The figure shows the backtesting performed using a fixed train-test split strategy with `training_period`: 6 months and `test_period`: 1 month. **Bottom Panel**: Same as the figure in the top panel, but focused on the test period to demonstrate the capacity of the model. The evaluation metrics are `MAE`: 7.51, `MSE`: 83.21, and `RMSE`: 9.12.

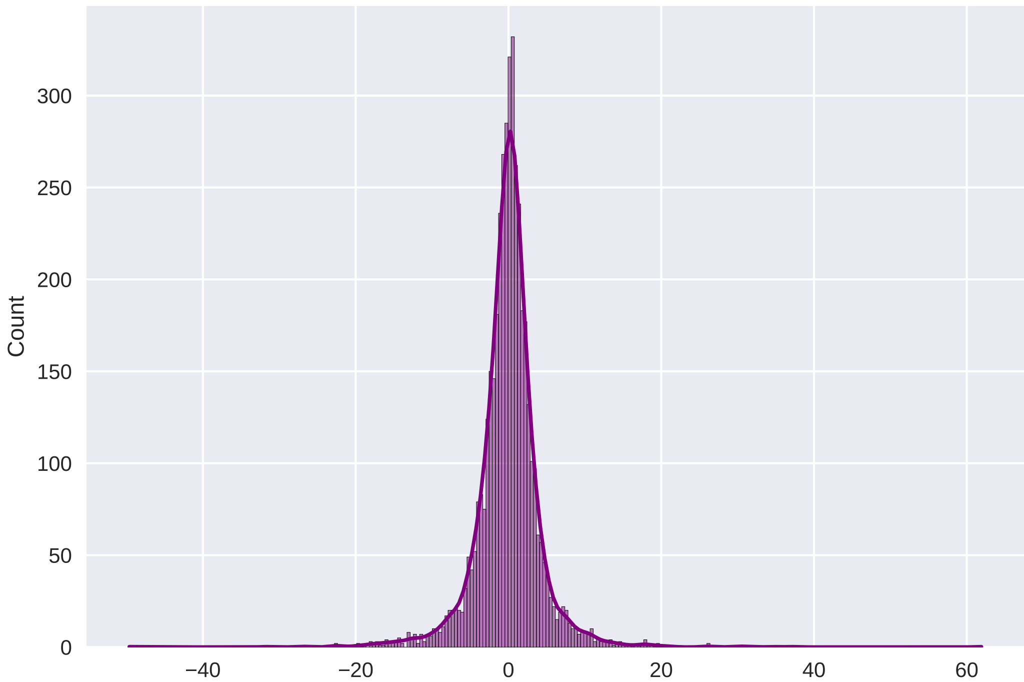



Figure 7: The probability distribution of residuals for the train-test split strategy. The PDF is nearly symmetric on both sides of zero, with the mean being close to zero. This implies similar probabilities for positive and negative values, suggesting a white noise behaviour of the residuals.

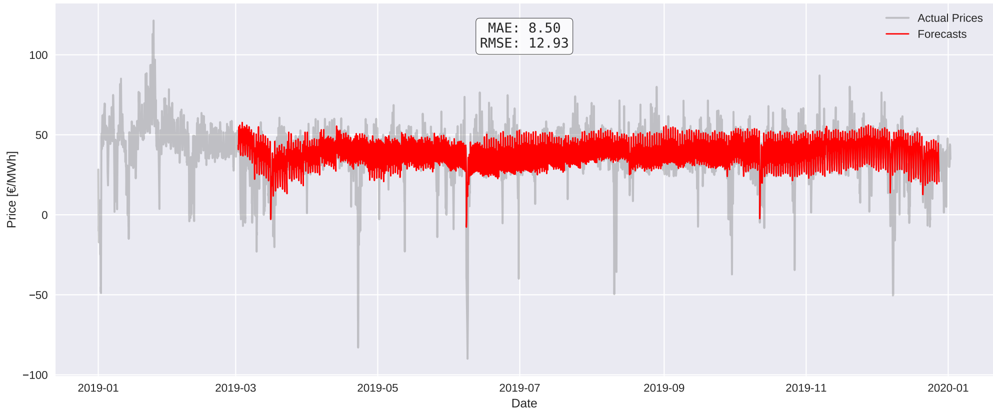



Figure 8: The figure shows backtested results using the expanding window method. The initial training set is 60 days, and the test set is 7 days.

also taking the test set. Moreover, the text set now shifts to the next 7-day period. This process is repeated for several iterations over one year (January 01, 2019, to January 01, 2020). The results are shown in fig. 8.

An expanding window is a powerful backtesting method that can capture extended period correlations and long-term seasonality in the TSD. However, it is perhaps the most computationally expensive method among the three backtesting methods we work with. The reason is that the model has to train on the data that expands after every iteration. Thus, the training period keeps increasing after every iteration. This is precisely why we were not able to use the expanding window backtesting method for a period longer than one year. On top of the runtime, the memory usage also keeps increasing gradually with the number of iterations.

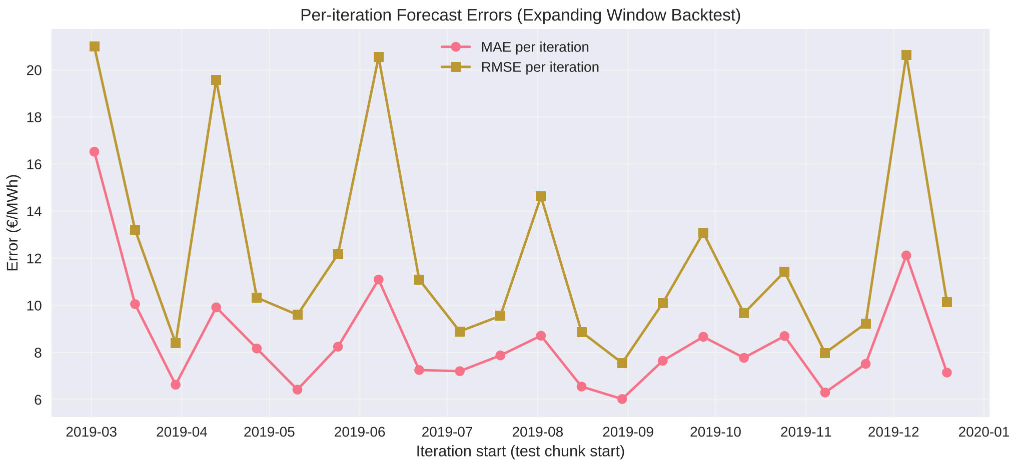



Figure 9: The figure shows improvement in the evaluation metrics as a function of the number of iterations.

We also plot the evolution of RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) with respect to the iterations in the expansion phase in fig. 9. Overall, the error reduces significantly at later stages, as the training set becomes larger and accounts for longer seasonality.

Furthermore, we also show rolling MAE in fig. 10. A rolling evaluation is a more reliable way to

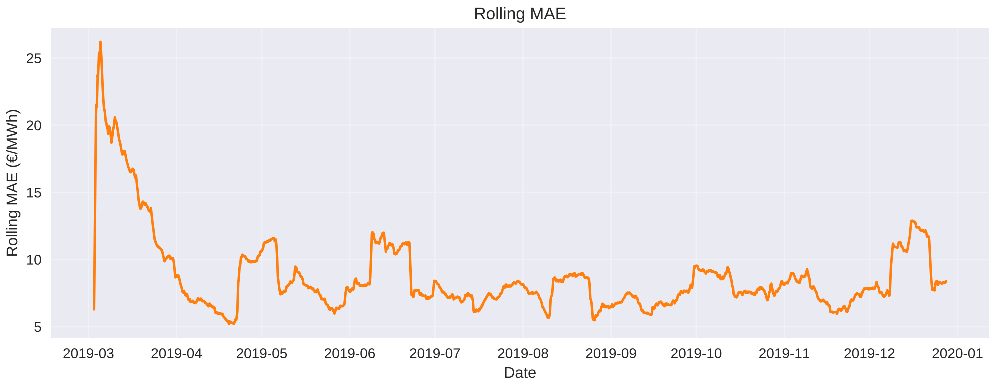



Figure 10: Plot of the rolling MAE over 14 days.

quantify the progressive nature of the errors with expansion. It can be noticed that the rolling MAE keeps decreasing as the training size expands, as expected.

### 4.3 Sliding Window

In the sliding window backtest, we keep the training size fixed and slide it further after every iteration. This makes it much faster than the expanding window method, where the training size is growing continuously. As shown in fig. 11, we fix the training set to a 3-month window and the test set to a 14-day window. In the next iteration, the training set will include the test set from the previous iteration and simultaneously remove the equally sized earliest data from the previous training set. An important consequence of the sliding method is the lack of long-term correlations. Hence, this method is robust in testing a TSD with short-term patterns; however, in a TSD with longer seasonality, it begins to lose accuracy. The latter feature is visible in the plots for the evaluation metrics figs. 12 and 13.

## 5 Forecast Results

Finally, we perform actual forecasts to predict the hourly prices for the whole 2024 calendar year. Here, we test four setups, viz., a model trained on the full five-year history and the ones trained on the most recent one, two, and three years. It is evident from fig. 14 that the hourly forecasts are nearly identical (ratio $\approx 1$, very similar hour-by-hour deviations), indicating that the shape of the HPFC curve is stable and exogenous variables and the most recent market conditions. In other words, the training data from the most recent three years is sufficient to make the hourly price forecasts.

In the analysis shown in fig. 15, we work with the models trained on the last year and the last two years of training data. The result shows a marginal difference exhibited by the forecasts of the two models. Note, however, that for this setup comparison, there is a factor of two difference in the forecasts, as seen from fig. 17. Model B is expected to have a higher amplitude of oscillations, implying a higher volatility. This is intuitive, since the training data of model B includes the 2022 crisis period, which makes its forecasts more sensitive to the market volatility.

## 6 Improvements & Future Work

The forecasts can be made more accurate by extending the above analysis in several directions, some of which are listed below:

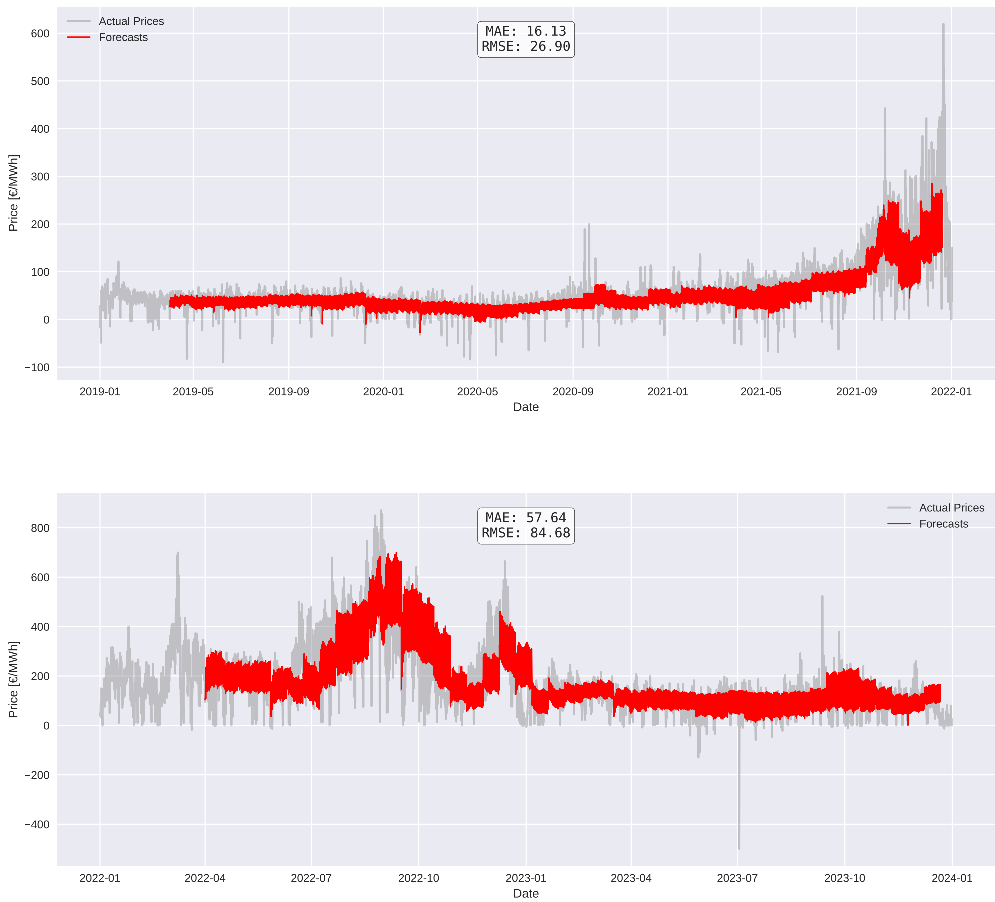



Figure 11: **Top Panel**: Shows the sliding window applied with an initial training set of 90 days and a corresponding initial test set of 14 days. The analysis is performed on the 3-year dataset, from January 01, 2019, to December 31, 2021. **Bottom Panel**: Same as the figure in the top panel, but analysis performed on the last two years of data, i.e., from January 01, 2022, to December 31, 2023.


Figure 12: The figures show the corresponding analysis ofig. 11. **Top Panel**: Shows improvement in the evaluation metrics as a function of the number of iterations. The error stays nearly constant and increases rapidly towards the final few iterations. The analysis is performed on the 3-year dataset, from January 01, 2019, to December 31, 2021. **Bottom Panel**: Same as the figure in the top panel, but analysis performed on the last two years of data, i.e., from January 01, 2022, to December 31, 2023. Note that, the large spikes in the metrics during the entire 2022 are related to the crisis period discussed in section 2.4.

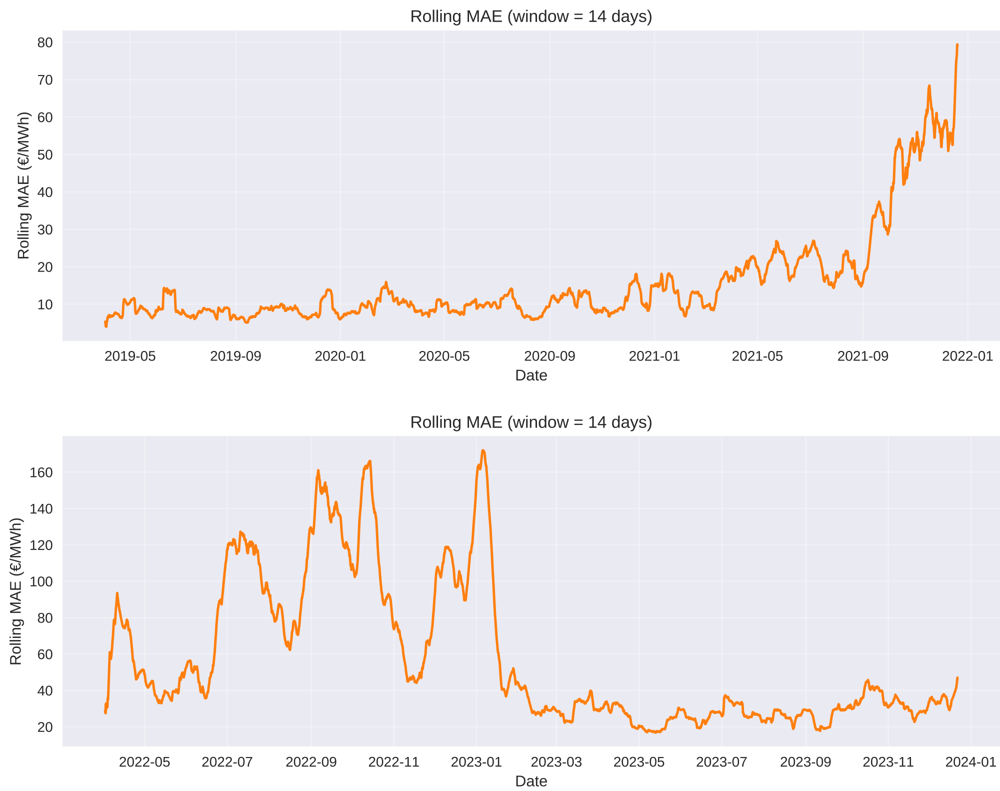



Figure 13: The figures show the corresponding analysis ofig. 11. **Top Panel**: Shows the rolling MAE as a function of sliding period. As expected, the error gradually rises over the longer time periods, but it stays sufficiently low for most of the time, except towards the very end. The analysis is performed on the 3-year dataset, from January 01, 2019, to December 31, 2021. **Bottom Panel**: Same as the figure in the top panel, but analysis performed on the last two years of data, i.e., from January 01, 2022, to December 31, 2023. Note that the large spikes in the metrics during the entire 2022 are related to the crisis period discussed in section 2.4.

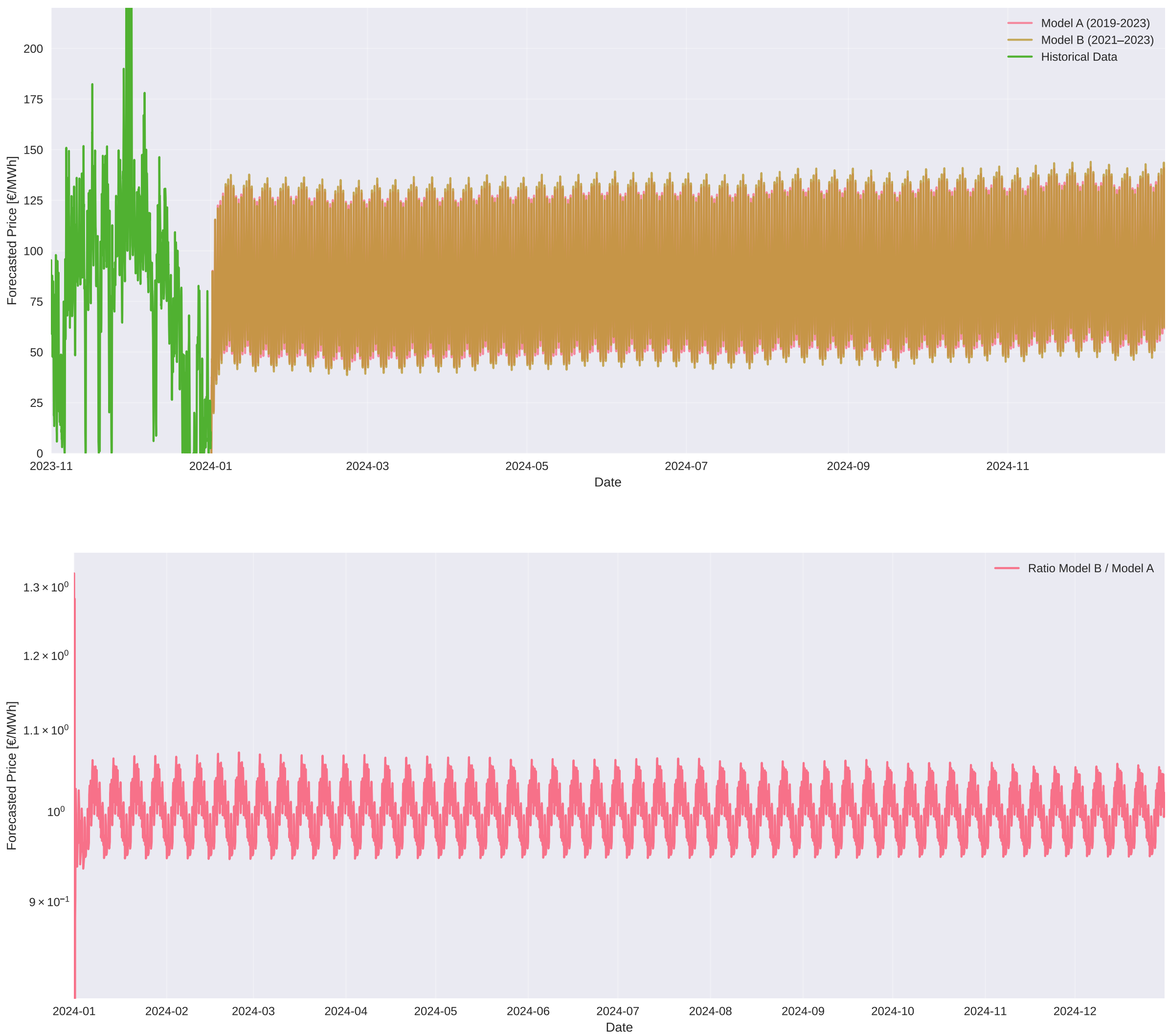



Figure 14: **Top Panel**: The figure shows a comparison between the hourly price forecasts for 2024 based on Model A (trained on five years of data from January 01, 2019, to December 31, 2023) and Model B (trained on the last two years of data from January 01, 2019, to December 31, 2023). The forecasts are scaled so that the yearly average is equal to 85 €/MWh (An arbitrary amount). **Bottom Panel**: Shows the ratio of the models in the top panel.


Figure 15: **Top Panel**: The figure shows a comparison between the hourly price forecasts for 2024 based on Model A (trained on the last year of data from January 01, 2023, to December 31, 2023) and Model B (trained on the last two years of data from January 01, 2019, to December 31, 2023). The forecasts are scaled so that the yearly average is equal to 85 €/MWh (An arbitrary amount). **Bottom Panel**: Shows the ratio of the models in the top panel.

- In 2024, renewable energy accounted for more than 60 percent of the share of Germany's total generated electricity, with wind power (32%), solar power (15%), and hydropower (4%) being the leaders. Renewables are thus major contributors to the volatility in prices. If we include the historical data of local climate (where power plants are situated), such as average wind speed, UV Index, or Average rainfall, as exogenous variables in the SARIMAX model, we can predict the period of high power generation with more accuracy.
- More advanced models promise better forecasts, particularly, LSTM (Deep Learning & Neural Networks) or XGBoost (Machine Learning) [3]. A primary reason is that these models catch the non-linear patterns and correlations present in the TSD. On the other hand, ARIMA-based models are linear models (both autoregression and moving averages are linear combinations of different terms). Furthermore, it is relatively more intuitive and easier to add new features to ML models, such as rolling periods or crisis regimes. The disadvantage is that one requires large computational resources and data to train these advanced models.
- We chose the seasonality parameter $s = 24$ in the analysis (see section 3.2); however, by doing so, we only captured the daily pattern and overlooked the weekly pattern of the hourly data. If we keep $s = 168, 8760$, we can also capture the weekly seasonality or yearly seasonality in the data, but this increases the model complexity, and the Python kernel crashes owing to memory overflow. Such an analysis could be carried out with a computing cluster environment that boasts large memory.

# A Unscaled Forecasts

This section shows the unscaled forecasts by the four different setups. By unscaled, we mean that we keep the original scales of the forecasted results and refrain from scaling them with the factor of 85 €/MWh.

# References

[1] C. McHugh, S. Coleman, D. Kerr, and D. McGlynn, "Forecasting day-ahead electricity prices with a sarimax model," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1523–1529. 2019.

[2] H. Bozdogan, "Model selection and akaike's information criterion (aic): The general theory and its analytical extensions," *Psychometrika* **52** no. 3, (1987) 345–370.

[3] H. Mubarak, A. Abdellatif, S. Ahmad, M. Zohurul Islam, S. Muyeen, M. Abdul Mannan, and I. Kamwa, "Day-ahead electricity price forecasting using a cnn-bilstm model in conjunction with autoregressive modeling and hyperparameter optimization," *International Journal of Electrical Power & Energy Systems* **161** (2024) 110206.

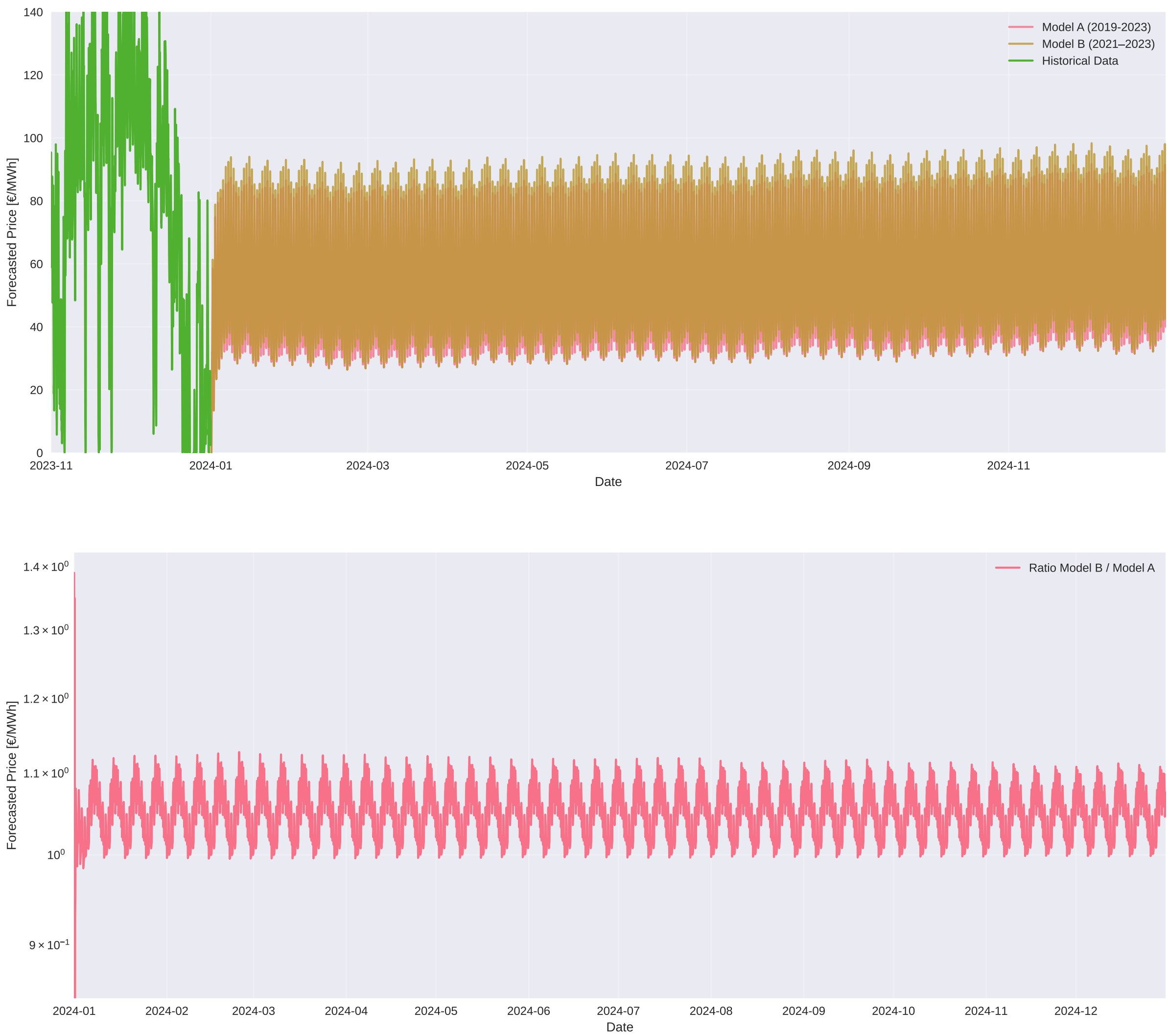



Figure 16: The unscaled counterpart of fig. 14.


Figure 17: The unscaled counterpart of fig. 15.