Technical Note: Imbalance Price Forecasting Pipeline

Candidate Submission

December 1, 2025

Abstract

This document outlines the methodology employed to develop a forecasting pipeline for imbalance prices (P_{long}, P_{short}) and system state probabilities (P_{state}). We detail the data preprocessing steps, the evaluation of classical time series benchmarks (SARIMA), and the rationale behind the transition to a Machine Learning approach. Particular emphasis is placed on feature engineering strategies designed to capture market inertia and regime-conditional dynamics without reliance on external data.

1 Introduction

The objective of this assignment is to predict the dual nature of imbalance prices—specifically the conditional prices for "Long" and "Short" system states, along with the probability of the system state itself. This report documents the end-to-end process, from initial data inspection to the engineering of a feature set suitable for Gradient Boosting models.

2 Data Inspection and Preprocessing

The provided dataset consists of imbalance prices and quantities at a 15-minute resolution (MTU). Initial inspection revealed a high-quality dataset with no 'NaN' values in the raw files, preserving the strict temporal ordering required for rigorous time series analysis.

However, a structural mismatch was detected: the `imb_quantity` dataset contained 96 more records than `imb_price`. Furthermore, a gap analysis on the MTU sequence identified missing price data for exactly one operational day. To maintain the integrity of the frequency (96 steps/day) required for lag creation, a custom imputation strategy was implemented.

2.1 Conditional Imputation Strategy

Simply filling gaps with a global mean would distort the signal. Instead, missing price values were reconstructed using a **conditional mean approach**. We calculated the average price from other days in the same month, conditional on:

1. The specific Market Time Unit (MTU), to respect daily seasonality.
2. The system state (Surplus or Deficit), to respect the regime-dependent nature of prices.

This ensured that the imputed values preserved both the seasonal shape and the statistical properties of the imbalance regime.

3 Feature Engineering

Given the project constraint prohibiting external data (e.g., weather forecasts, day-ahead prices), the feature engineering strategy focused on extracting latent signals from the historical series. The function `build_all_features` generates a rich feature space designed to capture four specific market dynamics.

3.1 Temporal Context & Cyclicality

Tree-based models do not inherently understand the continuity of time. To address this, we engineered:

- **Cyclical Encoding:** Sine and Cosine transformations of the MTU index were created to preserve the temporal proximity between MTU 96 (23:45) and MTU 1 (00:00).
- **Calendar Flags:** Boolean indicators for `is_weekend` and `is_peak_hour` (08:00–20:00) serve as explicit split points for the decision trees, isolating distinct demand behaviors.

3.2 Pseudo-Exogenous Variables (Weather Proxies)

Lacking actual weather data, we constructed heuristic proxies to capture renewable generation effects, validated by empirical observation:

- **Solar Proxy (`is_sun`):** By combining month and hour-of-day, this feature identifies windows of likely high solar generation. This helps the model distinguish between a "surplus caused by night wind" and a "surplus caused by midday sun," which impact prices differently due to cannibalization effects.
- **Ramp Indicators:** Features flagging morning (07:00–09:00) and evening ramps capture transition periods where grid stress is statistically highest.

3.3 Market Inertia & Volatility

- **Lags:** Lag steps (1, 2, 4, 8, 12, 24) were selected based on PACF analysis. The dominance of Lag-1 (ACF ≈ 0.80) confirms high system inertia.
- **Rolling Statistics:** Rolling means and standard deviations (windows of 1h to 4h) capture the local volatility regime. This allows the model to adapt its uncertainty: a high rolling standard deviation signals an unstable grid, prompting more conservative predictions.

3.4 Regime Dynamics & Interactions

The core challenge is predicting conditional prices based on the system state.

- **Regime Memory:** Features such as `long_streak` (consecutive MTUs in surplus) and `long_share` quantify the persistence of the current state. A system that has been "Long" for 4 hours is structurally different from one oscillating rapidly.
- **Price-Quantity Interaction:** We created interaction terms decomposed by regime (e.g., `price` × `quantity_pos`). This explicitly models the non-linear supply curve, allowing the Gradient Boosting algorithm to learn the slope of the "hockey stick" curve—i.e., how price sensitivity increases as the imbalance quantity becomes extreme.

4 Models and Results

4.1 Forecasting strategy

All models share the same two–stage, multi–horizon design. At each forecast origin (D, t) and for each horizon $h = 1, \ldots, 24$:

1. **Stage 1 – Quantity:** a regression model predicts the future imbalance quantity $Q_{D,t}(h)$ from the feature vector at time (D, t).
2. **Stage 2 – Prices and state:** the predicted quantity $\widehat{Q}_{D,t}(h)$ is appended as an extra feature and used to:
   - regress the conditional imbalance price in each regime (Long / Short);
   - estimate the probability that the system will be Long, $\hat{\pi}_{D,t}(h)$.

For the stronger models, the price regressions are performed on residuals

$$r_{D,t}(h) = P_{D,t+h} - P_{D,t},$$

so that forecasts are anchored to the latest observed price: $\widehat{P}_{D,t}(h) = P_{D,t} + \widehat{r}_{D,t}(h)$. This reduces non–stationarity and stabilises the multi–step behaviour.

The forecasting strategy is **direct multi–horizon**: we train a separate pair of models for each horizon h. This avoids error accumulation typical of recursive schemes and lets the model adapt to the different noise level at short vs. long horizons. Train/validation splits are strictly time–based. Hyperparameters are selected with a rolling `TimeSeriesSplit` and a gap of 24 MTUs between train and validation blocks to prevent any leakage via overlapping lags.

4.2 Benchmark: linear two–stage model

As a simple benchmark I use a two–stage linear model (OLS / ridge) with the engineered features. Figure 1 summarizes its out–of–sample performance on the last 30 days.

The top–left panel shows that the conditional price RMSE starts around 30 EUR/MWh for $h = 1$ and increases smoothly towards ≈ 39 EUR/MWh by $h = 24$. The MAE follows the same pattern, rising from about 17 to 27 EUR/MWh. This is expected: the further we look into the future, the less information is contained in the current state.

The top–right panel reports the AUC of the Long/Short classifier. Discrimination is excellent at very short horizons (AUC ≈ 0.95 at $h = 1$) and degrades gradually as h increases, but remains around 0.70 at $h = 24$, which is still useful for trading decisions.

The lower–left panel plots the mean conditional forecasts for P_{long}, P_{short} and their expectation $\mathbb{E}[\text{price}] = \hat{\pi}\,\widehat{P}_{\text{long}} + (1 - \hat{\pi})\,\widehat{P}_{\text{short}}$. The model learns a realistic wedge between Long and Short prices (Short prices consistently much higher) and a mild downward slope across horizons.

Finally, the lower–right panel shows the Brier score as a function of h. Values are low at short horizons and increase towards ≈ 0.18 at one day ahead, indicating that probability forecasts become less sharp but remain reasonably well calibrated.

4.3 Stronger model: XGBoost two–stage

The stronger model replaces linear regressions with gradient–boosted trees (XGBoost) in both the quantity and price residual stages, and an XGBoost classifier for the Long/Short state. Figure 2 reports the corresponding test performance.

The price–error curves (top–left) lie slightly below those of the linear benchmark for almost all horizons. The improvement is most visible in the first few hours ahead, where both RMSE and MAE are reduced by a few EUR/MWh. Beyond 12 hours the two curves are very close, suggesting that non–linear interactions and regime effects are most informative at short horizons.

The AUC and Brier profiles (top–right and bottom–right) are very similar to the linear model. This indicates that most of the discrimination power comes from the engineered features themselves, while the choice of classifier (logistic vs. XGBoost) has only a second–order effect. Given the already high AUC and moderate Brier scores, I did not apply an additional post–hoc calibration method such as isotonic regression.

The bottom–left panel highlights the main difference between the models. XGBoost produces more dispersed conditional prices, especially in the Short regime, and is able to generate higher spikes than the linear benchmark. This reflects the non–linear price–quantity relationship: extreme imbalances map to disproportionately large prices, a pattern that trees capture more naturally than linear regressions.

4.4 Why calibration matters

In this setup a trading strategy would size positions as a function of $\hat{\pi}_{D,t}(h)$: for instance, increasing exposure when the model assigns a high probability to the system being Long. Good discrimination alone is not sufficient; if probabilities are miscalibrated (e.g. events predicted at 80% actually happen only 60% of the time), the strategy will systematically over– or under–size positions and can have negative expected P&L despite a high AUC.

Well–calibrated probabilities ensure that predicted confidence levels correspond to empirical frequencies. This makes it possible to map $\hat{\pi}_{D,t}(h)$ into position sizes and risk limits in a consistent way across horizons, turning the model outputs into actionable trading signals rather than just ranking scores.

5 Possible Improvements and Extensions

This work deliberately focuses on a self–contained setup, using only imbalance prices and quantities. Several extensions could further improve the quality and robustness of the forecasts:

- **Including day–ahead prices as exogenous drivers.** Having access to day–ahead prices would allow us to work with stationary price *spreads* or returns (e.g. imbalance minus day–ahead), which are typically much more stable than raw levels. This would make time–series models (ARIMA, VAR, state–space models) more effective and would also reduce the burden on the feature engineering needed to enforce stationarity.
- **Longer history and richer training set.** The current models are trained on a relatively short sample. Additional years of data would improve the estimation of rare but economically relevant events (extreme imbalance and price spikes), stabilise the regime–specific regressions and reduce variance in long–horizon forecasts.
- **Leveraging GPUs and more expressive models.** With access to GPUs, the CatBoost and XGBoost pipelines could be tuned more aggressively (deeper trees, larger ensembles, extensive hyperparameter search) without violating time constraints. Moreover, sequence models such as LSTMs or temporal convolutional networks could be explored on top of the engineered features and/or raw sequences, potentially capturing higher–order temporal patterns that are difficult to model with purely tabular methods.

These extensions were considered out of scope for the present submission, but they indicate a clear path to turn the current prototype into a production–grade forecasting system.

A Model Diagnostics

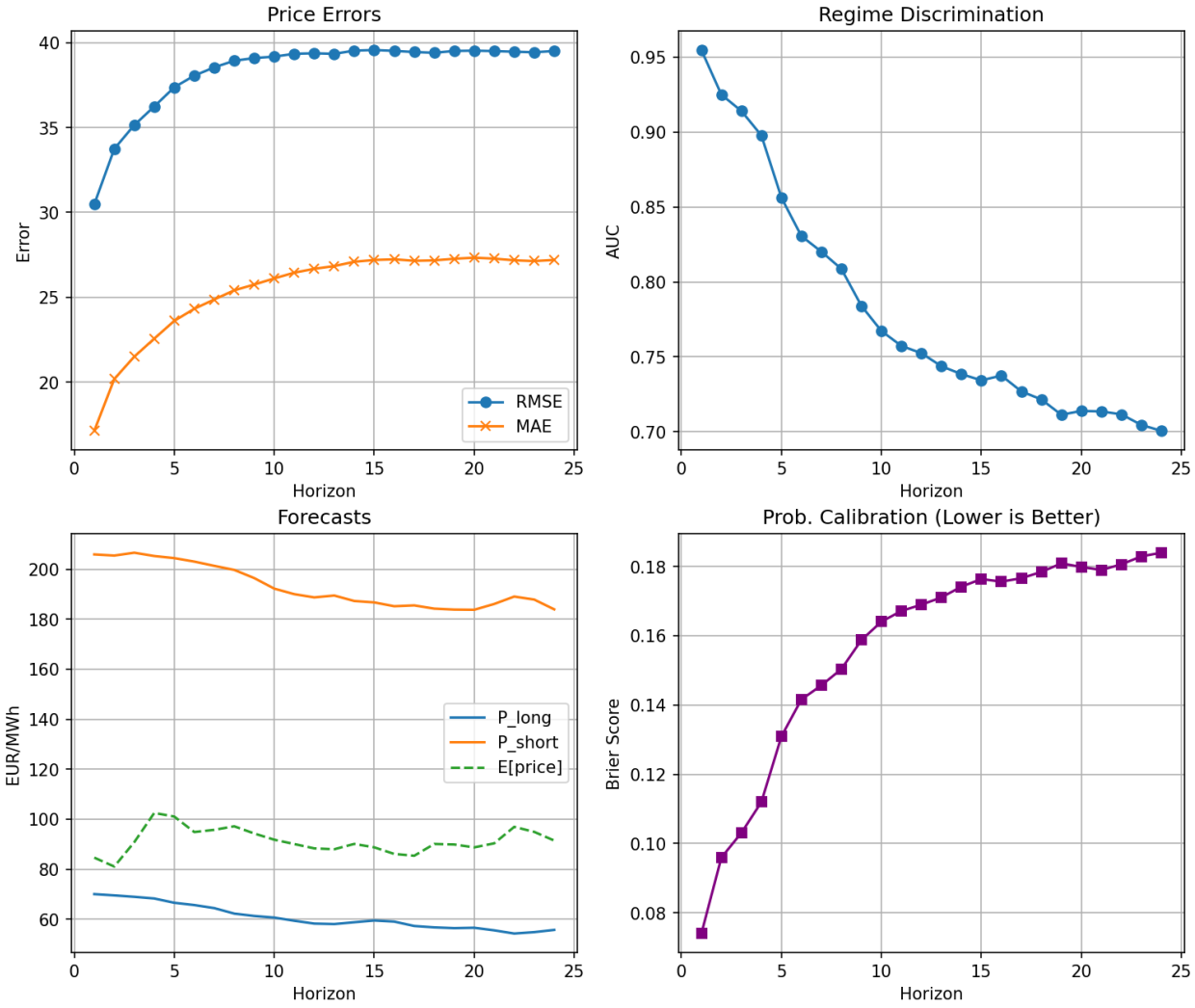

Figure 1: Linear / Ridge two–stage benchmark: price errors, regime discrimination, conditional forecasts and probability calibration across horizons.

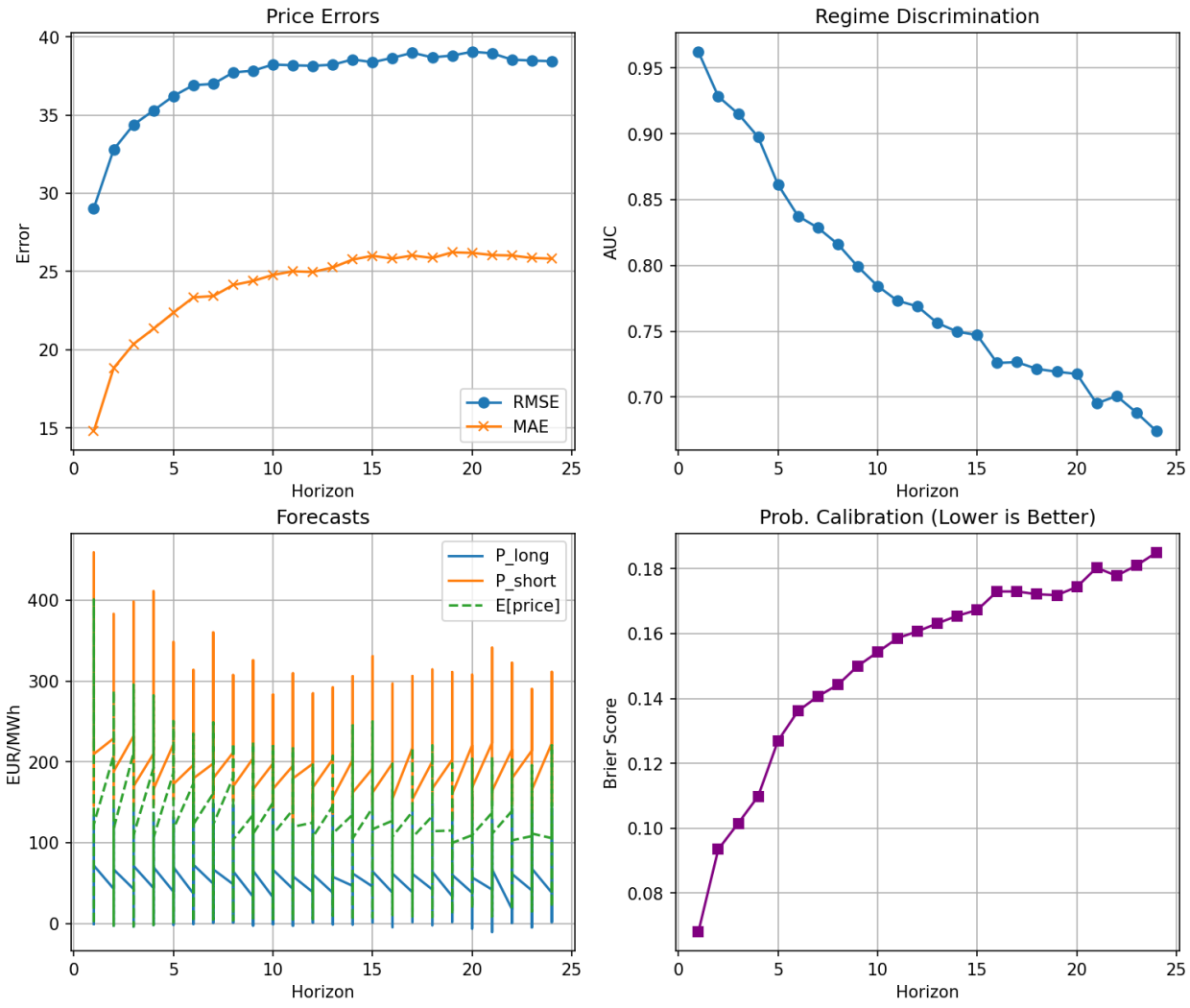

Figure 2: XGBoost two–stage model: price errors, regime discrimination, conditional forecasts and probability calibration across horizons.

B Time Series Benchmark: SARIMA

As a baseline experiment, we attempted to model the price series using a Seasonal AutoRegressive Integrated Moving Average (SARIMA) framework.

B.1 Methodology and Stationarity

Electricity prices exhibit complex dual seasonality: a daily cycle ($S_1 = 96$) and a weekly cycle ($S_2 = 672$). To address the inherent non-stationarity and volatility clustering (heteroskedasticity), we applied a Variance Stabilizing Transformation (VST) as recommended in recent literature. Specifically, we utilized the Inverse Hyperbolic Sine transformation:

$$Y_t^* = \operatorname{arcsinh}\left(\frac{Y_t}{\sigma}\right) \tag{1}$$

where σ is the standard deviation of the series. This approach is supported by Uniejewski et al. (2019) [1] as robust for periods of increased volatility.

B.2 Limitations and Pivot to Machine Learning

Despite these transformations, the SARIMA experimentation was halted due to three critical limitations:

1. **Computational Intensity:** Fitting a double-seasonal SARIMA model on 15-minute resolution data proved computationally prohibitive for a recursive forecasting pipeline.
2. **Regime-Blindness:** The SARIMA model is univariate and outputs a single expected value $\hat{y}_t$. It cannot inherently produce the two distinct conditional forecasts (P_{long} and P_{short}) required by the problem statement without significant, likely inaccurate, assumptions.
3. **Lack of Exogenous Drivers:** The model failed to capture the non-linear "hockey-stick" relationship between Price and Quantity.

Consequently, the strategy shifted towards a Feature-Engineering-heavy approach to feed non-linear Gradient Boosting models (XGBoost), which can handle conditional targets and exogenous interactions naturally.

References

[1] Uniejewski, B., Weron, R. (2019). *Variance Stabilizing Transformations for Electricity Price Forecasting in Periods of Increased Volatility.* IFRO Working Paper, No. 2019/10. Available at: `https://www.econstor.eu/bitstream/10419/211087/1/IFRO_WP_2019_10.pdf`