POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Benchmarking Models for Very Short-Term Forecasting of Imbalance Prices in the Italian Electricity Market

Tesi di Laurea Magistrale in
Energy Engineering - Ingegneria Energetica

Valerio Rivadossi, 222033

Abstract: The increasing integration of non-programmable renewable energy sources has made the management of power grid balancing increasingly complex. In the Italian context, a major shift has occurred regarding the management and economic valuation of grid balancing resources, creating new opportunities for market operators to exploit their flexibility. In this new scenario, driven by the introduction of the TIDE reform (Testo Integrato del Dispacciamento Elettrico), having an accurate, real-time forecast of the imbalance price becomes therefore relevant for optimizing market strategies. This work aims to develop and compare different models for the near real-time forecasting of imbalance prices in the Italian electricity market. To this end, five algorithms have been implemented and compared: Lasso Estimated AutoRegressive (LEAR), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Support Vector Regression (SVR), and Long Short-Term Memory (LSTM). The experimental results demonstrate that XGBoost achieves the highest accuracy with a MAE of 43.77 €/MWh and an sMAPE of 43.57%. Meanwhile, Random Forest proves to be the most effective in managing price volatility, achieving the best RMSE of 65.91 €/MWh and an R^2 of 0.39. Finally, all implemented models would allow market operators to significantly increase profitability by making intentional imbalances based on price forecasts, a practice that remains strictly prohibited by Italian market regulations.

Advisor:
Prof. Filippo Bovera

Co-advisors:
Andrea Scrocca

Academic year:
2024-2025

Key-words: imbalance price forecasting, Italian electricity market, Short-Term Forecasting, machine learning

1. Introduction

The last two decades have been characterized by the increasing decarbonization of the energy sector, leading to a massive integration of Variable Renewable Energy (VRE) sources. In 2024, Italy's solar production reached a record 36 TWh, accounting for approximately 13.6% of the total national generation, while wind energy contributed about 8.4% [1]. These numbers are expected to increase as Italy is required to comply with the European 'Fit for 55' targets, which aim to reduce net greenhouse gas emissions by at least 55% by 2030 compared to 1990 levels. These targets are implemented through the Piano Nazionale Integrato Energia e Clima (PNIEC) [2]. By 2030, these targets require reaching approximately 131 GW of installed renewable capacity to cover over 65% of national electricity generation, a necessary step towards the carbon neutrality set by the European Green Deal for 2050. However, the stochastic nature of VRE increases generation forecast uncertainty, making the management of real-time grid balancing more complex.
In fact, in order to maintain the proper and stable operation of the grid, the quantity of energy injected into the grid must be equal to the quantity withdrawn at every instant, a balance guaranteed by the Transmission System Operator (TSO) which is Terna.
The TSO compensates for deviations between scheduled and actual energy production/consumption, known as imbalance volumes, through the balancing platforms and the Balancing Market (BM), where the main components of the imbalance price are determined, which represents the economic cost to maintain grid equilibrium.
In this context, the TIDE reform (Testo Integrato del dispacciamento Elettrico) was introduced in January 2025 [3]. Among its various innovations, it introduces the principle of technological neutrality, potentially allowing all grid resources to participate in balancing services. Furthermore, the imbalance settlement mechanism has been revolutionized, shifting from an hourly-based dual pricing scheme to a single pricing system with quarter-hourly granularity. While dual pricing penalizes any deviation from the commercial schedule, single pricing remunerates imbalances that help restore grid equilibrium.
This transformation makes imbalance price forecasting crucial, as it represents a concrete financial risk and opportunity for market operators. Therefore, the development of predictive models capable of supporting real-time decisions is essential to effectively manage imbalances within this new framework.

1.1. Literature review and research gap

The scientific literature on Electricity Price Forecasting has historically been dominated by studies focused on the Day-Ahead Market (DAM). In this context, the works of Weron [4] and Lago et al. [5, 6] established the fundamental benchmarks for the field. Weron highlighted the necessary transition from classical statistical models to computational intelligence to properly handle the non-linearities and high volatility of electricity prices. Subsequently, through an extensive comparative analysis of a wide range of forecasting models, Lago et al. demonstrated the superiority of Deep Learning (DL) architectures over traditional algorithms. Furthermore, they formalized the Lasso Estimated AutoRegressive (LEAR) model as the linear benchmark, proving that regularization is essential to automatically select the most influential exogenous variables in complex markets. In light of its proven effectiveness, the LEAR model is widely recognized as a robust linear benchmark in the existing literature.

Despite the maturity reached in DAM forecasting, the literature becomes significantly sparser when shifting towards near real-time environments, such as the Balancing Market. As highlighted by O'Connor et al. [7] in their comprehensive review of the sector, price formation in these markets is highly stochastic, leading to inconsistent forecasting performance that heavily depend on the geographical area and the specific settlement rules adopted by the Transmission System Operators.
Focusing on a specific geographical context, a related study led by O'Connor et al. [8] evaluated different forecasting models on the Irish Integrated Single Electricity Market. In this environment, the authors showed that Machine Learning (ML) algorithms such as Random Forest and eXtreme Gradient Boosting, alongside strongly regularized regressions like LEAR, actually outperform more complex DL architectures in managing extreme volatility. This finding highlights a crucial divergence from the Day-Ahead Market, where deep neural networks typically represent the state-of-the-art. The complexity of these near real-time markets is further underscored by Dinler [9], who points out that in the context of renewable-rich power systems, the literature on balancing price forecasting remains largely underdeveloped. Dinler specifically emphasizes the critical need for "spike-resilient" approaches capable of handling sudden price fluctuations. Furthermore, the review identifies an emerging trend towards "profit-aware" methodologies, acknowledging that models should be evaluated not only on standard statistical metrics but also on their practical economic impact.

Expanding on the capabilities of Machine Learning in near real-time markets, the work of Lucas et al. [10] extensively explored the application of tree-based ensemble algorithms, such as Random Forest and eXtreme

Gradient Boosting, in the United Kingdom's balancing market. Their study identified XGBoost as the most robust and suitable model for real-time forecasting, due to its well-distributed feature dependency. Crucially, the authors demonstrated that introducing the Loss of Load Probability (LOLP) as an exogenous variable significantly improves predictive accuracy compared to simpler baseline models. Furthermore, by combining the XGBoost algorithm with a quantile regression approach to manage the uncertainty of imbalance volumes, the proposed framework proved highly effective in capturing market dynamics and anticipating sudden price spikes. Operating on the same British balancing market data, Deng et al. [11] further extended this analysis by testing several deep learning architectures for the explicit point forecasting of near real-time imbalance prices. After comparing multiple machine learning baselines against different neural networks, they proposed a novel Seasonal Attention-based Bidirectional Long Short-Term Memory (SA-BiLSTM) framework. Rather than a standard LSTM network, this architecture incorporates bidirectional memory and seasonal attention mechanisms, allowing it to significantly outperform traditional tree-based models and standard neural networks in capturing complex cyclical patterns and extreme price variations.
In the German context, Narajewski [12] investigated the probabilistic forecasting of electricity imbalance prices. His study highlighted the significant difficulty of outperforming naive benchmarks, such as the last observed intraday price, in terms of standard point metrics like MAE and RMSE. However, by comparing advanced statistical models with probabilistic neural networks and GAMLSS, Narajewski demonstrated that these complex architectures provide superior empirical coverage compared to naive approaches, confirming their value in assessing market uncertainty.
Similarly, in Belgium, Dumas et al. [13] focused on forecasting imbalance prices by proposing a two-step approach. They demonstrated that utilizing Multi-Layer Perceptrons (MLPs) to first estimate TSO net regulation volumes, and subsequently mapping them to the final imbalance prices, yields superior performance compared to direct models that attempt to estimate the price directly from the input variables.

Across the analyzed studies, regardless of the specific methodology employed, the selection of exogenous variables remains a critical determinant of forecasting accuracy. The reviewed literature consistently highlights that the most widely utilized features include historical and real-time electricity market prices (such as Day-Ahead and Intraday prices), system load forecasts, and power generation predictions, with a specific focus on non-programmable renewable sources like wind and solar power.

As evident from the discussed studies, the global literature on imbalance price forecasting remains remarkably sparse. In the Italian context, this scarcity is even more pronounced. Existing academic works on Italian imbalances [14–16] are strictly limited to forecasting their sign or volume. To date, the point forecasting of Italian imbalance prices has not been systematically addressed, leaving a significant research gap that this study aims to fill.

1.2. Contribution and novelty

This research project seeks to develop and compare predictive models, establishing a comprehensive benchmark for the near real-time point forecasting of imbalance prices in the Italian electricity market, specifically focusing on the North macrozone. To mirror practical trading operations, the forecasting framework is designed to simulate the perspective of a market operator. Specifically, it executes point forecasts for a given quarter-hour delivery period (t) under strict real-time constraints, approximately 10 minutes before delivery ($t - 10$ min), relying exclusively on public data available at that exact moment.
In detail, the specific contributions and novelties provided by this work include:

- **Assessment of forecasting accuracy:** quantifying the empirical predictive performance that can be obtained under near real-time constraints to establish a benchmark for the imbalance price in the Italian electricity market.
- **Quantification of economic potential:** evaluating the practical financial value of the predictions by simulating a scenario of voluntary imbalance based on the forecasted price to quantify the resulting economic profitability.
- **Implementation of predictive models and feature configurations:** answering the previous questions by implementing and comparing five forecasting algorithms: a regularized linear model (Lasso Estimated AutoRegressive), three Machine Learning algorithms (Random Forest, eXtreme Gradient Boosting, Support Vector Regression), and one Deep Learning network (Long Short-Term Memory). Each model is analyzed using different sets of exogenous features and historical lags to evaluate their impact on the final prediction.

The thesis is structured as follows. First, an overview of the Italian electricity market framework is provided, with particular attention to the structure of the spot markets and the imbalance price determination mechanism. The dataset construction and feature selection process are then described, including the exogenous variables considered and the data transformation procedures adopted. Subsequently, the forecasting models implemented in this study are introduced, presenting their main concepts and mathematical formulations, moving from a simple Naive baseline to more advanced machine learning and deep learning architectures. The experimental setup and empirical results are then discussed, covering the statistical forecasting performance, the feature importance analysis, and the economic evaluation based on a simulated trading scenario. Finally, the thesis concludes by summarizing the main findings and outlining potential directions for future research.

2. Italian electricity markets

The scope of this section is to provide an overview of the Italian electricity market framework, highlighting the imbalance price determination mechanism.

2.1. Structure

Italian spot electricity market (Mercato Elettrico a Pronti MPE) is structured into 5 parts [17]: Mercato del Giorno Prima (MGP) - Day-Ahead Market (DAM), Mercato Infragiornaliero (MI) – Intraday Market (IDM), Mercato dei prodotti giornalieri (MPEG) - Daily Products Market, Mercato per il Servizio di Dispacciamento (MSD) – Ancillary Services Market (ASM).
The time granularity in the Italian electricity markets has shifted from an hourly basis to a quarter-hourly basis under the TIDE reform, with the IM and ASM adopting 15-minute intervals from 1 January 2025, and DAM together with the MPEG transitioning to 15-minute intervals from 1 October 2025. Regarding spatial granularity, Italian electricity market is divided into seven zones: Nord - North, Centro Nord - Centre-North, Centro Sud - Centre-South, Sud - South, Calabria - Calabria, Sicilia - Sicily, Sardegna - Sardinia.

2.1.1. Day-Ahead Market

In this market, the majority of wholesale electricity trading transactions take place. Prices and traded quantities are determined, as well as commercial injection and withdrawal positions for the following day. The market is organized according to an implicit auction model, which takes into account transmission constraints between grid zones.
DAM opens at 08:00 on the ninth day prior to the delivery day (D-9) and closes at 12:00 on the day prior to the delivery (D-1). Gestore dei Mercati Elettrici (GME), which acts as the central counterparty for all transactions, publishes the final market results at 12:58 on the same day (D-1). Market participants submit sell and buy bids, specifying for each the quantity and the minimum or maximum price at which they are respectively want to sell or purchase electricity. For each time interval, bids are accepted on the basis of economic merit, through the intersection of supply and demand curves, and in compliance with transmission constraints between market zones. The intersection of the two curves determines the quantity of electricity traded, the accepted bids and the clearing price. All accepted sell and buy bids are settled at the zonal clearing price. However, for accepted buy bids, the GME also calculates a compensatory component, equal to the product of the accepted quantity and the difference between the zonal price and the PUN Index GME.

2.1.2. Intra-Day Market

Participants adjust their commercial injection and withdrawal positions defined in the DAM by trading electricity demand bids and supply offers. The MI is organized into sequential auction sessions (MI-A) and continuous trading phases (MI-XBID). The sessions follow this order: MI-A1 (12:55–15:00 D-1), MI-XBID Phase I (15:30–21:40 D-1); MI-A2 (12:55–22:00 D-1), MI-XBID Phase II (22:30 D-1–H-1 for delivery hours 1–12 and until 09:40 D for delivery hours 13–24); MI-A3 (12:55 D-1–10:00 D), MI-XBID Phase III (10:30 D–H-1). In the MI-A auctions, interconnection capacities between Italian zones and cross-border lines are allocated using the Euphemia algorithm, and accepted bids and offers are valued at the zonal clearing price. In MI-XBID, bids and offers are continuously matched across zones based on available interconnection capacity, with transaction prices determined by time priority, and all accepted bids and offers are valued at the zonal clearing price of the relevant zone.

2.1.3. Daily Products Market

Market participants trade daily electricity products with delivery obligations through continuous trading. GME acts as counterparty and registers net delivery positions. Products include baseload (all hours of the day) and peak-load (08:00–20:00, weekdays). Orders are organized in a price-ranked order book, and transactions are automatically matched. At the end of each session, GME determines each participant's net delivery position and publishes minimum, maximum, reference prices, and volumes.

2.1.4. Ancillary Services Market

ASM is the market where Terna procures the majority of resources necessary for the management and control of the Italian power system. These resources are essential for tasks such as resolving congestion, establishing energy reserves, and real-time balancing.
In the ASM, Terna is the buyer and the BSPs (Balance Service Providers) are the sellers, and accepted offers are remunerated at the price submitted (pay-as-bid).
The ASM is organized into the programming phase MSD ex-ante and the Balancing Market (BM), both conducted in multiple sessions.
The MSD ex-ante is divided into six programming steps within a single offer submission session opening at 12:55 on D-1 and closing at 17:00 on D-1. In this phase, Terna accepts bids and offers of energy aimed at resolving residual congestion and voltage constraints, as well as establishing reserve margins for redispatching.
The BM involves the continuous submission of offers with a quarter-hourly detail for the 96 quarter-hours of the delivery day D. The session for submitting offers opens at 22:30 on D-1, and operators can submit offers up to one hour before the start of the relevant quarter-hour (H-1). In the BM, Terna accepts bids and offers to maintain real-time balance between injections and withdrawals on the network.
The ancillary services traded in the MSD ex-ante and BM include primary, secondary, and tertiary reserves. Primary reserve (FCR – Frequency Containment Reserve) provides almost instantaneous adjustments to stabilize the grid frequency. Secondary reserve (aFRR – automatic Frequency Restoration Reserve) involves automatic adjustments of generation or consumption over short periods to restore frequency. Tertiary reserves are used over longer periods to maintain system balance and are subdivided into manual Frequency Restoration Reserve (mFRR), which is manually activated to restore balance in the short to medium term, and Replacement Reserve (RR), a long-term reserve that replaces other activated reserves to ensure energy balance at all nodes.
In addition to the ASM, there are European market platforms dedicated to balancing: the Trans European Replacement Reserves Exchange (TERRE), for trading RR; Manually Activated Reserves Initiative (MARI) for mFRR; and the Platform for the International Coordination of Automated Frequency Restoration (PICASSO), for aFRR, coordinating balancing across multiple countries. If accepted offers on the MSD or European platforms are not executed, non-compliance fees for dispatching orders are applied.
The prices formed in the ASM and on the European balancing platforms directly contribute to the determination of the imbalance price.

2.2. Imbalance Price

Due to forecasting errors in energy demand and generation, particularly from renewable sources, discrepancies arise between the scheduled injection and withdrawal of electricity and the actual realized programs. These are known as imbalances, and their management is handled by the TSO. The imbalance price represents the economic valuation of this activity.
Terna utilizes resources from the ASM and European platforms based on the system state: if the imbalance is positive, which means that there is an excess of energy injected into the grid compared to withdrawals, due to higher actual generation or lower consumption than forecasted, the so-called 'downward services' are activated, aimed at reducing generation and increasing electricity withdrawal, conversely if the imbalance is negative, 'upward services' are activated. Typically, prices for downward services are lower than the day-ahead price, whereas prices for upward services are higher.
As a result of TIDE reform [3], the imbalance price formation mechanism shifted from a predominantly dual pricing system to a single pricing one.
As shown in equations (2), this algorithm considers only the zonal imbalance: for each time interval, its sign is assessed to determine the price applied to all operators, regardless of the sign of their individual imbalance.
Unlike the spot market's 7-zone structure, imbalance determination is based on two macrozones: the North Macrozone, consisting of the North market zone, and the South Macrozone, which includes the remaining six zones. This research will take the North Macrozone as the market area where the proposed forecasting models and features are analyzed.

In the TIDE, the macrozonal imbalance S_{mz} is defined as [3]:

$$S_{mz} = -\sum_{u \in mz} Prg_u^{fin} - \sum_{\substack{\text{UAS} \\ \text{UVA} \in mz}} E_u^{freq} - \sum_{u \in mz} E_u^{mod} - \sum_{j \neq mz} F_{mz,j}^{exc} + \sum_{\substack{\text{UCS} \\ \text{UCP} \in mz}} S_u \tag{1}$$

Where:

- Prg_u^{fin} is the final program for unit u;
- E_u^{mod} represents the overall modulation associated with unit u;
- $F_{mz,j}^{exc}$ is the energy exchanged in real-time between the imbalance macrozone mz and the imbalance macrozone or foreign bidding zone j;
- The last term represents the sum of the imbalances S_u of all Commercial Storage Units (UCS) and Commercial Collection Units (UCP) within the macrozone.

The imbalance price can be divided into 3 different cases: positive imbalance price P_{mz}^{sb+}, negative imbalance price P_{mz}^{sb-} and avoided activation price P_{mz}^{AE}:

$$P_{mz}^{sb} = \begin{cases} P_{mz}^{sb+} & \forall t \mid \left(S_{mz} > 0 \wedge \overline{Q \downarrow_{mz}^{bil}} \neq 0\right) \\ P_{mz}^{sb-} & \forall t \mid \left(S_{mz} < 0 \wedge \overline{Q \uparrow_{mz}^{bil}} \neq 0\right) \\ P_{mz}^{AE} & \forall t \mid S_{mz} = 0 \vee \left(S_{mz} \neq 0 \wedge \overline{Q \downarrow_{mz}^{bil}} + \overline{Q \uparrow_{mz}^{bil}} = 0\right) \end{cases} \tag{2}$$

where $Q \downarrow_{mz}^{bil}$ and $Q \uparrow_{mz}^{bil}$ are respectively the downward and upward balancing energy activated for the macrozone mz:

$$\overline{Q \downarrow_{mz}^{bil}} = \sum_{z \in mz} \left(\overline{Q \downarrow_z^{mFRR}} + \overline{Q \downarrow_z^{aFRR}} + \overline{Q \downarrow_z^{MB}}\right) \tag{3}$$

$$\overline{Q \uparrow_{mz}^{bil}} = \sum_{z \in mz} \left(\overline{Q \uparrow_z^{mFRR}} + \overline{Q \uparrow_z^{aFRR}} + \overline{Q \uparrow_z^{MB}}\right) \tag{4}$$

$\overline{Q \downarrow_z}$ and $\overline{Q \uparrow_z}$ represent the total downward and upward quantities for:

- aFRR / mFRR: energy procured on the respective balancing platforms for bidding zone z;
- MB: energy accepted on the Balancing Market for units within the zone.

Therefore, the single pricing mechanism leads to two distinct economic outcomes based on the alignment between the participant's deviation from the scheduled program and the macrozonal imbalance:

- **System-Aggravating Imbalances:** when an operator's deviation matches the sign of the macrozonal imbalance, the market player worsens the grid state, and so a financial disadvantage occurs. In a deficit state, the missing energy is purchased by the operator from Terna at the P_{mz}^{sb-} price, which is higher than the P_z^{MGP}. In a surplus state, the excess energy is sold by the operator to Terna at the P_{mz}^{sb+} price, which is lower than the P_z^{MGP}.
- **System-Helping Imbalances:** when the operator's deviation has the opposite sign to the macrozonal imbalance, the market player supports the grid. In a deficit state, the operator's surplus is sold to Terna and remunerated at the P_{mz}^{sb-} price, which results in a gain relative to the P_z^{MGP}. In a surplus state, the operator's additional consumption is purchased from Terna and settled at the P_{mz}^{sb+} price, which is lower than the P_z^{MGP}, offering an economic advantage.

2.2.1. Positive imbalance price

Positive imbalance price P_{mz}^{sb+} is determined by two components: the base price P_{mz}^{base+} and the incentive P_{mz}^{inc+}:

$$P_{mz}^{sb+} = P_{mz}^{base+} + P_{mz}^{inc+} \tag{5}$$

where:

$$P_{mz}^{base+} = \frac{\sum_{z \in mz} \left(P_z^{mFRR} \times \overline{Q \downarrow_z^{mFRR}} + P_a^{aFRR} \times \overline{Q \downarrow_z^{aFRR}} + P \downarrow_z^{MB} \times \overline{Q \downarrow_z^{MB}}\right)}{\sum_{z \in mz} \left(\overline{Q \downarrow_z^{mFRR}} + \overline{Q \downarrow_z^{aFRR}} + \overline{Q \downarrow_z^{MB}}\right)} \tag{6}$$

$$P_{mz}^{inc+} = \min \left\{0, \left[\min_{z \in m} \left(P_z^{MGP}\right) - P_{mz}^{base+}\right]\right\} \tag{7}$$

Where:

- P_a^{aFRR}: the price relative to the Load-Frequency Control (LFC) area a for resources activated on the aFRR platform;
- P_z^{mFRR}: the price relative to the bidding zone z for resources activated on the mFRR platform;
- $P\uparrow_z^{MB}$ and $P\downarrow_z^{MB}$: the average upward and downward activation prices, respectively, on the Balancing Market for resources located within the bidding zone z.
- P_z^{MGP}: the zonal price of bidding zone z resulting from the Day-Ahead Market;

Therefore, the base imbalance price is equal to the weighted average of the prices of downward resources procured on the balancing platforms and on the BM.
The incentive component P_{mz}^{inc+} ensures that the positive imbalance price remains at or below the minimum zonal price cleared in the DAM.

2.2.2. Negative imbalance price

Analogously, the negative imbalance price P_{mz}^{sb-} is defined by the sum of a base price P_{mz}^{base-} and an incentive component P_{mz}^{inc-}:

$$P_{mz}^{base-} = \frac{\sum_{z \in mz} \left(P_z^{mFRR} \times \overline{Q \uparrow_z^{mFRR}} + P_a^{aFRR} \times \overline{Q \uparrow_z^{aFRR}} + P_z^{\uparrow MB} \times \overline{Q \uparrow_z^{MB}} \right)}{\sum_{z \in mz} \left(\overline{Q \uparrow_z^{mFRR}} + \overline{Q \uparrow_z^{aFRR}} + \overline{Q \uparrow_z^{MB}} \right)} \tag{8}$$

$$P_{mz}^{inc-} = \max \left\{ 0, \left[\max_{z \in mz} \left(P_z^{MGP} \right) - P_{mz}^{base-} \right] \right\} \tag{9}$$

Therefore, the base imbalance price is equal to the weighted average of the prices of upward resources procured on the balancing platforms and on the BM.
The incentive component P_{mz}^{inc-} ensures that the negative imbalance price remains at or above the maximum zonal price cleared in the DAM.

2.2.3. Avoided activation price

In the event of a null macrozonal imbalance ($S_{mz} = 0$), Terna is not required to activate any resources to balance the system. In this scenario, the imbalance price is determined based on the value of avoided activations, that is also relevant when the aggregate macrozonal imbalance is non-zero but is fully compensated through the Imbalance Netting platform. Imbalance Netting is a cross-border cooperation mechanism that allows TSOs to net opposite imbalances between adjacent control areas, thereby avoiding the unnecessary activation of secondary reserves aFRR. This process is executed via the European IGCC (International Grid Control Cooperation) platform. Functionally, IGCC operates in coordination with the PICASSO platform, as the Imbalance Netting process precedes the activation phase, effectively reducing the volume of reserves that need to be mobilized on PICASSO. Indeed, even in this case, Terna avoids activating specific balancing resources, meaning it does not issue any activation program for the units located in the macrozone mz.
The avoided activation price is determined by two components: the base avoided activation price P_{mz}^{AEbase} and the incentive component P_{mz}^{AEinc}:

$$P_{mz}^{AE} = P_{mz}^{AEbase} + P_{mz}^{AEinc} \tag{10}$$

The calculation of the base avoided activation price P_{mz}^{AEbase} is governed by specific criteria designed to ensure it is representative of the marginal imbalance value. Specifically, this price must reflect the merit order of bids on the Balancing Market for the relevant macrozone. However, an exception applies when the aggregate macrozonal imbalance S_{mz} is fully compensated through the Imbalance Netting platform. In this specific scenario, P_{mz}^{AEbase} is determined to reflect the opportunity cost defined by Terna for the valuation of exchanges on the Imbalance Netting platform. For further details regarding the components of these equations and their evaluation, please refer to the Italian Grid Code [18].

3. Dataset

The purpose of this chapter is to construct the dataset used by the forecasting models, defining the features selected and the criteria for their inclusion.
This choice was based on the exogenous variables involved in the physical and economic process of imbalance price and quantity formation, as well as those most frequently identified in literature. Furthermore, since one of the main objectives of this work is to develop models capable of near real-time forecasting, the selection was restricted to variables available at the moment of execution. Specifically, to estimate the imbalance price for the time interval t, only data published up to approximately $t - 10$ minutes are taken into consideration. This ensures that the forecasting process is feasible even for the most computationally intensive models analyzed in this work. The dataset spans from 1 April 2022 to 30 September 2025, corresponding to the period of availability for the target variable. Only data relating to the North macrozone are considered.

The selected features are:

- **Imbalance Price** P^{sb}_{mz} [€/MWh]: the target variable of the model, published by Terna at approximately 17:00 on the day following delivery (D+1) [19].
- **Preliminary Imbalance Price** $P^{sb}_{mz,prel}$ [€/MWh]: the imbalance price calculated from unconsolidated data; while the TIDE framework expects publication within 30 minutes for each time interval t. Terna typically observes a 20-minute delay, meaning that values from $t - 30$ minutes and prior are taken into consideration. This feature is the most significant exogenous variable as it directly represents the same quantity that the model aims to predict [19].
- **Preliminary Imbalance Volume** $S_{mz,prel}$ [MWh]: the imbalance volume S_{mz} derived from unconsolidated data; since the publication timing is identical to that of the preliminary price, values from $t - 30$ minutes and prior are considered [19].
- **DAM Price** P^{MGP}_{z} [€/MWh]: the zonal price resulting from the Day-Ahead Market (DAM), published at D-1 by the GME [20].
- **MI-A mean price** P^{mean}_{MI-A} [€/MWh]: the volume-weighted average price of the three auction sessions of the Intraday Market (MI-A1, MI-A2, and MI-A3), published by the GME [20] and available one hour in advance (t-60 minutes).
- P^{last}_{XBID} [€/MWh]: the last coupling price resulting from the continuous trading sessions (MI-XBID), published on the GME website one hour before the delivery time [20]. Among the candidate variables evaluated for this market were the first coupling price, the volume-weighted average of all trades, and the volume-weighted average of trades within the final hour of trading. P^{last}_{XBID} was selected as it yielded better correlation coefficients and error metrics. Note that while MI-XBID includes products with both 60-minute and 15-minute granularity, the latter has been available only since January 2025 and was consequently excluded due to the lack of sufficient historical data.
- **Solar and Wind Generation Error** Err^{sw}_{gen} [MW]: a synthetic variable defined as the difference between the forecasted generation and the actual production for the sum of solar and wind sources. The two sources were aggregated because the wind contribution in the North macrozone is very small. The data are sourced from European Network of Transmission System Operators for Electricity platform (ENTSO-E) [21], which is required to publish actual generation values within 60 minutes; since these values are typically available with a 30 to 45-minute delay, values from $t - 60$ minutes and prior are taken into consideration.
- G^{f}_{tot} [MW]: total generation forecast, provided by ENTSO-E [21] by 18:00 on D-1 (Central European Time).
- L^{f}_{tot} [MW]: total load forecast, published by Terna [22] by 12:00 on D-1.

Table 1 presents the final selection of exogenous variables, reporting the statistical metrics corresponding to the optimal time lag identified for each feature. Correlation coefficients (Pearson, Spearman and R^2) were calculated to assess the dependency with the target price. Additionally, for variables sharing the same unit of measurement as the target (€/MWh), error and fit metrics (MAE, RMSE) were also computed.
The results show a clear distinction between price-based features and other physical variables. The former exhibit a significant linear dependency with the target, with P^{last}_{XBID} achieving the highest Pearson correlation (0.67) and an R^2 of 0.45. Conversely, the last four features show very weak correlations and negligible R^2 values. Furthermore, the analysis of MAE and RMSE reveals that although the preliminary data $P^{sb}_{mz,prel}$ is the most accurate on average with the lowest MAE (56.49 €/MWh), its high RMSE (132.66 €/MWh) indicates a greater susceptibility to price spikes compared to market-based variables.

Exogenous features	Best Lag	MAE [€/MWh]	RMSE [€/MWh]	R^2	Pearson	Spearman
$P^{sb}_{mz,prel}$	60 min	56.49	132.66	0.30	0.63	0.67
P^{MGP}_{z}	0 min	72.23	119.36	0.43	0.66	0.59
P^{mean}_{MI-A}	0 min	71.10	119.26	0.44	0.66	0.60
P^{last}_{XBID}	0 min	65.80	117.28	0.45	0.67	0.63
$S_{mz,prel}$	60 min	—	—	0.05	-0.21	-0.42
Err^{sw}_{gen}	60 min	—	—	0.05	-0.22	-0.24
G^{f}_{tot}	0 min	—	—	0.04	0.21	0.27
L^{f}_{tot}	0 min	—	—	0.02	0.16	0.19

Table 1: Statistical metrics and correlation coefficients for the selected exogenous features.

Regarding data granularity, the imbalance prices (P^{sb}_{mz} and $P^{sb}_{mz,prel}$), preliminary volumes ($S_{mz,prel}$), and total load forecast (L^{f}_{tot}) feature a quarter-hourly resolution for the entire study period. Conversely, the remaining exogenous variables were initially published with hourly resolution, transitioning to a 15-minute granularity only as of 1 January 2025. To ensure a uniform 15-minute time series, hourly data from the preceding period were upsampled by applying the single hourly observation to each of its four constituent quarter-hourly intervals. Furthermore, to address the presence of gaps in the raw dataset, a forward-fill strategy was employed to handle missing values. This approach involves propagating the last valid observation forward to fill subsequent missing timestamps until a new value is recorded. In addition, the dataset is expanded with specific temporal features and rolling statistics, which are described in detail in Section 3.1 and Section 3.2, respectively.

3.1. Time features

To incorporate temporal information, specific features were added to the dataset using three different configurations, which were tested across all the models implemented in this study. These configurations represent common solutions found in the literature for handling periodicity in electricity markets:

- **Seven Binary Dummy Variables**: this mechanism, also known as one-hot encoding, transforms the categorical "day of the week" variable into seven separate binary columns. For each observation, only the column corresponding to the specific day is assigned a value of 1, while the remaining 6 columns are set to 0.
- **Seven Dummies and Integer Variable**: this setup combines the seven binary indicators described above with an additional integer variable representing the quarter-hours of the day, with values ranging from 0 to 95. Random Forest (RF) and XGBoost models achieved slightly better performance using this specific configuration.
- **Four Cyclic Variables**: this method uses sine and cosine transformations to create two variables for the day of the week and two for the quarter-hour. After testing, SVR and LSTM models showed slightly better results with this configuration. The variables are defined by equations (11a)–(11d)

$$Day_{sin} = \sin\left(\frac{2\pi \cdot d}{7}\right) \tag{11a}$$

$$Day_{cos} = \cos\left(\frac{2\pi \cdot d}{7}\right) \tag{11b}$$

$$QH_{sin} = \sin\left(\frac{2\pi \cdot q}{96}\right) \tag{11c}$$

$$QH_{cos} = \cos\left(\frac{2\pi \cdot q}{96}\right) \tag{11d}$$

Regarding the LEAR model, a configuration of six dummy variables representing the days of the week was adopted, instead of seven. This is a standard procedure in literature to avoid perfect multicollinearity among features, ensuring the stability of the linear estimation.

3.2. Rolling Statistics

A 24-hour moving average and a 24-hour moving volatility of preliminary imbalance prices were included as common solutions found in the literature [4, 6]. These features were calculated using preliminary data because the target variable is published at 17:00 on D+1 and is therefore unavailable during the forecasting process. To assess their impact, every model was tested both with and without these variables; while their inclusion improved the predictive accuracy for most architectures, the LEAR model achieved better performance in the configuration without these statistics.

3.3. Data Transformation

The data may undergo a transformation process divided into two categories: variance stabilization and normalization [6].

3.3.1. Variance Stabilizing Transformation (VST)

Electricity price time series are characterized by high volatility, extreme spikes, and the possible occurrence of zero or negative values. To address these behaviors, a variance stabilizing transformation based on the inverse hyperbolic sine function is applied. The transformation compresses large observations with a logarithmic-like behavior while remaining approximately linear around zero, thereby preserving information contained in small price variations. As a result, extreme price movements are moderated without removing them, leading to a more symmetric and statistically stable distribution that is more suitable for forecasting models. Unlike logarithmic transformations, arcsinh is defined for both positive and negative values, making it particularly appropriate for electricity market data.
The transformation is defined as:

$$y_{trans} = \text{arcsinh}(x) = \ln\left(x + \sqrt{x^2 + 1}\right) \tag{12}$$

3.3.2. Normalization

Following variance stabilization, normalization can be applied to rescale all variables onto a comparable numerical scale. This step prevents differences in magnitude from influencing the estimation process and improves numerical stability during model training. Scaling also facilitates the learning process, often leading to faster and more reliable convergence. In addition to standard normalization, robust scaling may be considered to limit the influence of extreme observations while preserving the overall structure of the data.
Two different scaling methods are used. The first is standard scaling, which is defined as:

$$z_{standard} = \frac{y_{trans} - \mu}{\sigma} \tag{13}$$

where μ represents the mean and σ the standard deviation of the distribution. The second approach is robust scaling, defined as:

$$z_{robust} = \frac{y_{trans} - \text{Median}(Y)}{\text{MAD}} \tag{14}$$

The Median Absolute Deviation (MAD) is calculated as:

$$\text{MAD} = \text{Median}\left(|y_{trans} - \text{Median}(Y)|\right) \tag{15}$$

Each model was tested with all possible combinations of these transformations, applied to both the input features and the target variable. The empirical results highlighted that the optimal preprocessing strategy strictly depends on the specific architecture. Specifically, LEAR achieved its best performance using both the VST and robust scaling. In contrast, RF yields the best results without applied transformations. For XGBoost, the optimal configuration involved applying VST exclusively to the target variable. Finally, both SVR and LSTM reached their peak performance when the data were processed using a combination of VST and standard scaling.

4. Forecasting Models

This section provides an overview of the models implemented in this study and describes their main architectural characteristics and underlying logic. The methodology progresses from a simple baseline to more complex machine learning and deep learning frameworks.

4.1. Baseline: the Naive model

The Naive model serves as the baseline for evaluating the performance of more complex architectures. This model utilizes the preliminary imbalance price lagged by 60 minutes ($t-60$ min) as its prediction for the current time step t. This specific configuration was selected because, as analyzed in Section 3, the 60-minute lag offers the best predictive performance among the preliminary features. The model is defined as follows:

$$\hat{P}_t = P_{mz,prel}(t - 60min) \tag{16}$$

where $\hat{P}_t$ represents the forecasted imbalance price for the interval t and $P_{mz,prel}(t-60min)$ is the preliminary price recorded one hour prior.

4.2. Lasso Estimated AutoRegressive

This approach is grounded in the foundational work of Uniejewski et al. [23], which underscores the importance of automated variable selection and shrinkage in time series forecasting to handle high-dimensional datasets while preventing overfitting. Building on these principles, the Lasso Estimated AutoRegressive model has been formalized and recognized as an established linear benchmark in the electricity price forecasting literature, as extensively demonstrated by Lago et al. [6]. The LEAR model effectively combines an autoregressive (AR) structure with the Lasso (Least Absolute Shrinkage and Selection Operator) estimator, maintaining model interpretability even in complex market scenarios.

The core mechanism of the Lasso estimator is based on adding an L_1 penalty term to the cost function, which is proportional to the sum of the absolute values of the coefficients. Unlike standard linear regression models, this method forces the coefficients of irrelevant or redundant variables to become exactly zero by applying a shrinkage effect on the parameters. This characteristic transforms the model into an automated variable selection tool. This is crucial because it allows for the inclusion of a large number of potential input features without selecting them manually beforehand. Consequently, the LEAR model keeps only the predictors that really contribute to the forecasting accuracy, reducing statistical noise and the risk of overfitting.
A distinct aspect of the implementation developed in this work concerns the autoregressive component. Due to real-time data constraints, actual imbalance prices are not immediately available when the forecast is made. Therefore, the model uses preliminary prices $P_{mz,prel}$ as a proxy for the target variable to build the autoregressive features.
Formally, the price forecast $\hat{P}_t$ at time step t is defined as a linear combination of the features:

$$\hat{P}_t = \hat{\beta}_0 + \sum_{j=1}^{N_f} x_{t,j}\hat{\beta}_j \tag{17}$$

where:
- N_f denotes the total number of features considered by the model;
- $x_{t,j}$ represents the value of the j-th feature at time t;
- $\hat{\beta}_0$ is the estimated intercept of the model, representing the baseline value when all features are zero;
- $\hat{\beta}_j$ are the estimated coefficients associated with each feature.

Although the forecasting equation uses proxies for the input features, the training phase aims to minimize the error with respect to the real imbalance price P_t^{sb}. The optimal parameters $\hat{\beta}$ are estimated by solving the following minimization problem:

$$\hat{\beta} = \arg\min_{\beta} \left(\underbrace{\sum_{t=1}^{N} (P_t - \hat{P}_t)^2}_{\text{Error on Real Prices}} + \underbrace{\lambda \sum_{j=1}^{N_p} |\beta_j|}_{\text{Lasso Penalty}} \right) \tag{18}$$

In this equation:

- N represents the number of observations in the training dataset;
- P_t is the actual imbalance price;
- λ is the regularization hyperparameter. It controls the penalty intensity applied to the model: higher values of λ force a larger number of coefficients to become exactly zero: λ acts as an automatic filter that keeps only the most relevant input features and discards the uninformative ones, effectively preventing the model from overfitting the data.

The procedure used to optimally tune the hyperparameter λ is described in Section 5.1.2.

4.3. Random Forest

The Random Forest (RF) is a widely adopted machine learning model due to its versatility, ease of implementation and strong predictive performance across a broad range of applications. It is an ensemble learning method based on the combination of multiple decision trees for both regression and classification tasks [24]. In this work, the model was implemented using the scikit-learn library.
Figure 1 schematizes the algorithm, which relies on two core stochastic principles: bootstrap aggregating (bagging) and feature randomness.

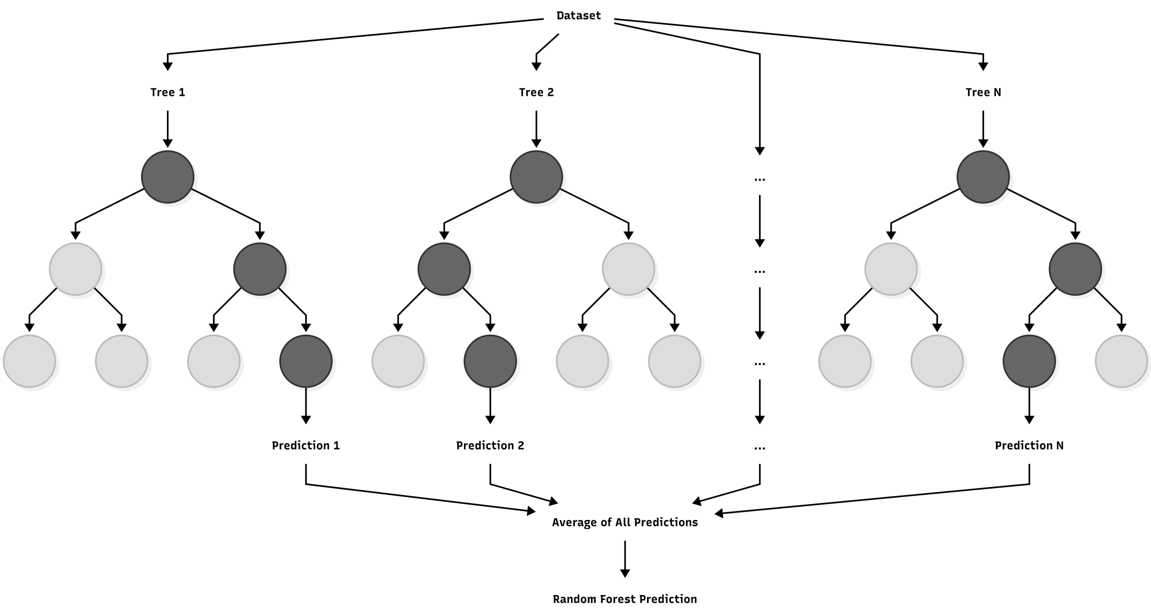

Figure 1: Schematic representation of the Random Forest algorithm architecture.

The process starts with bagging: instead of training every tree on the exact same data, the algorithm creates a custom training set for each tree by randomly sampling observations from the original dataset with replacement. As a result, a specific observation may appear multiple times in one tree while being absent in another, allowing each tree to learn from a slightly different version of the dataset.
Simultaneously, at each node split, the model considers only a random subset of the features rather than the full set. This mechanism reduces the correlation between individual trees, preventing the ensemble from relying excessively on a few dominant variables and promoting greater diversity among the learned trees.
During the training phase, for which a detailed description can be found in [24], the individual trees are grown by minimizing the Mean Squared Error (MSE) [6]. After training, each regression tree T_b produces an independent prediction for the input vector x_t. The final price forecast $\hat{P}_t$ is obtained by calculating the average of these outputs:

$$\hat{P}_t = \frac{1}{N_{tree}} \sum_{b=1}^{N_{tree}} T_b(x_t) \tag{19}$$

where N_{tree} represents the total number of trees in the forest. This aggregation reduces the overall variance of the model, thereby mitigating generalization error.
The model configuration involves several hyperparameters that control the structure of the individual trees and the ensemble aggregation. Table 2 provides the list and description of the hyperparameters optimized in this

study along with their respective search ranges. The optimization procedure is discussed in Section 5.1.2.

Hyperparameter	Description	Range
Number of Estimators	Total number of trees in the forest (N_{tree})	50 - 1000
Maximum Depth	Maximum levels in each tree to control complexity	10 - 100
Minimum Samples for Split	Minimum number of samples required to split an internal node	5 - 100
Minimum Samples per Leaf	Minimum number of samples required to be at a leaf node	2 - 50
Maximum Features	Number or fraction of predictors considered at each split	0.3 - 1.0
Maximum Samples	Fraction of training samples drawn with replacement for each tree (bagging)	0.5 - 1.0
Minimum Impurity Decrease	Threshold for splitting based on error reduction	0.0 - 0.2

Table 2: List of optimized Random Forest hyperparameters with their descriptions and search ranges.

4.4. eXtreme Gradient Boosting

eXtreme Gradient Boosting is an ensemble learning algorithm based on gradient boosting decision trees, widely adopted in machine learning applications due to its high predictive accuracy and computational efficiency [25]. Unlike Random Forest, which utilizes an ensemble of independent trees trained in parallel, XGBoost arranges its trees sequentially. In this framework, each subsequent tree is specifically built to correct the residual errors of the prediction made by the previous one.
Mathematically, the final electricity price forecast $\hat{P}_t$ is defined as:

$$\hat{P}_t = \sum_{k=1}^{N_{tree}} f_k(x_t) \tag{20}$$

where N_{tree} is the total number of trees in the ensemble, f_k represents the independent prediction of the k-th regression tree, and x_t is the input feature vector at time step t.
To train the ensemble, XGBoost sequentially adds the tree f_k that best minimizes the objective function $\mathcal{L}$:

$$\mathcal{L} = \sum_t l(P_t, \hat{P}_t) + \sum_{k=1}^{N_{tree}} \Omega(f_k) \tag{21}$$

where l is a differentiable loss function measuring the error between the actual and forecast price. The term Ω is a penalty used to limit the complexity of each tree, defined as follows:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 + \alpha \sum_{j=1}^{T} |w_j| \tag{22}$$

In this expression, T represents the number of leaves in the tree, while w_j are the scores (weights) assigned to each leaf. The parameters γ, α, and λ act as penalties that prevent the tree from becoming too large or detailed. This explicit regularization is the key feature that distinguishes XGBoost from standard gradient boosting, helping to avoid overfitting when dealing with highly volatile data like electricity prices.
In this study, the model has been implemented using the xgboost library through the XGBRegressor interface. The model configuration relies on several hyperparameters that control both the tree-building process and the sequential learning dynamics. Table 3 provides the list and description of the hyperparameters optimized in this study along with their respective search ranges. The optimization procedure is discussed in Section 5.1.2.

Hyperparameter	Description	Range
Number of Estimators	Total number of sequential trees (N_{tree})	200 - 2500
Maximum Depth	Maximum depth of each tree to control complexity	3 - 12
Learning Rate	Step size shrinkage used to prevent overfitting (η)	0.01 - 0.3
Minimum Child Weight	Minimum weight required to create a new node split	1 - 150
Subsample	Fraction of training samples used for each tree	0.5 - 0.95
Colsample by Tree	Fraction of features considered for each tree	0.5 - 0.95
L1 Regularization (α)	Penalty term on leaf weights (Lasso)	10^{-8} - 20.0
L2 Regularization (λ)	Penalty term on leaf weights (Ridge)	10^{-8} - 20.0

Table 3: List of optimized XGBoost hyperparameters with their descriptions and search ranges.

4.5. Support Vector Regression

Support Vector Regression is a regression technique based on the principles of Support Vector Machines (SVM). Unlike traditional regression methods that aim to minimize the distance between all data points and the regression line, SVR seeks to find a function that stays within a specific error range, defined by a threshold ϵ [26].
As shown in Figure 2, this approach creates an "ϵ-insensitive tube" around the prediction. Errors smaller than ϵ are ignored, while points falling outside the tube, known as support vector, are penalized. To handle these points, the model introduces slack variables, ξ_i and ξ_i^*, which measure the deviation of samples outside the tube [26].

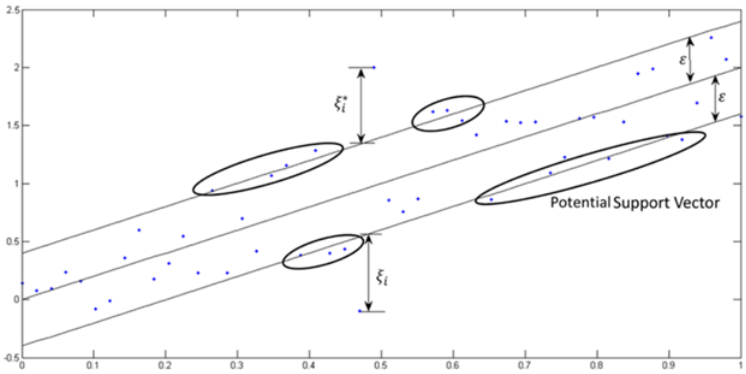

Figure 2: Schematic representation of the SVR ϵ-insensitive tube [26].

Mathematically, the electricity price forecast $\hat{P}_t$ at time step t is expressed as a linear combination of the input features:

$$\hat{P}_t = \langle w, x_t \rangle + b = \sum_{j=1}^{M} w_j x_{t,j} + b \tag{23}$$

where w represents the weight vector, x_t is the input feature vector of size M at time t, $\langle w, x_t \rangle$ denotes their inner product, and b is the bias term. To train the SVR model, the goal is to find the optimal weights w by solving the following constrained optimization problem:

$$\min \left(\underbrace{\frac{1}{2}||w||^2}_{\text{Regularization}} + \underbrace{C \sum_{i=1}^{N} (\xi_i + \xi_i^*)}_{\text{Error Penalty}} \right)$$
$$\text{subject to} \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i^* \\ \langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{24}$$

In this formulation, y_i represents the actual observed electricity price P_t at time t. This equation is composed of two main elements: a regularization penalty on the model weights w, and the sum of the slack variables ξ_i and ξ_i^*. Minimizing the first term promotes a flatter and less complex function to prevent overfitting, while the second term quantifies the magnitude of the errors for points lying outside the ϵ-tube. The constraints ensure that most data points remain within the distance ϵ from the regression line, while the slack variables allow for deviations in the case of outliers or high volatility.

In this study, the model has been implemented using the `scikit-learn` library through the `SVR` interface. The model configuration relies on essential hyperparameters that control the geometry of the tube and the regularization trade-off. Table 4 provides the list and description of the hyperparameters optimized in this study along with their respective search ranges. The optimization procedure is discussed in Section 5.1.2.

Hyperparameter	Description	Range
Regularization Parameter (C)	Determines the trade-off between function flatness and prediction error	1.0 - 1000.0
Epsilon (ϵ)	Defines the width of the insensitive tube where errors are ignored	0.001 - 0.2

Table 4: List of optimized SVR hyperparameters with descriptions and search ranges.

4.6. Long Short-Term Memory

Long Short-Term Memory is an advanced architecture of Recurrent Neural Networks (RNNs) specifically designed to model temporal sequences and capture long-range dependencies [27]. Unlike standard RNNs, which struggle with the vanishing gradient problem when dealing with long sequences, LSTMs incorporate a memory cell and a sophisticated gating mechanism to regulate the flow of information.

By examining the core building block of this architecture, a single LSTM unit (shown in Figure 3), the exact data flow through the system can be clearly observed.

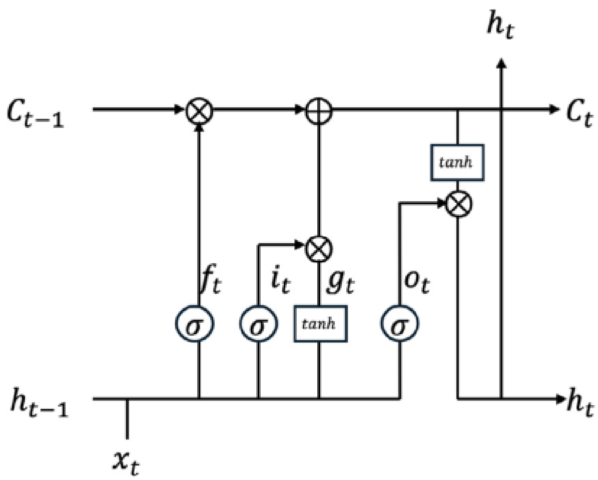

Figure 3: Internal architecture of an LSTM cell [11].

As seen on the top horizontal line of the diagram, the network maintains a cell state C_t, which serves as the primary memory track of the model. Rather than completely overwriting the state at each time step, this linear path allows information to flow across the temporal sequence with minimal modifications, ensuring that long-term dependencies are preserved.

At the bottom of the figure, the current input x_t and the previous hidden state h_{t-1} are combined. The interaction with the main memory track is then controlled by three distinct branches, known as gates:

- **The forget gate (f_t):** uses a sigmoid function (σ) to decide what information from the past should be discarded.
- **The input gate (i_t):** works alongside the candidate cell state (g_t) to determine what new information should be added to the memory track.
- **The output gate (o_t):** determines what parts of the updated cell state should be output as the new hidden state (h_t).

To capture particularly complex patterns in time-series data, such as electricity prices, multiple LSTM layers can be stacked one after another, allowing the network to progressively learn deeper trends. Mathematically, given an input sequence, the operation of the LSTM cell at time step t is governed by the following set of equations:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) && \text{Forget Gate} \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) && \text{Input Gate} \\ g_t &= \tanh(W_g \cdot [h_{t-1}, x_t] + b_g) && \text{Candidate Cell State} \\ C_t &= f_t \odot C_{t-1} + i_t \odot g_t && \text{Cell State Update} \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) && \text{Output Gate} \\ h_t &= o_t \odot \tanh(C_t) && \text{Hidden State} \end{aligned} \tag{25}$$

where x_t is the input vector at time t, h_{t-1} is the hidden state from the previous time step, and $[h_{t-1}, x_t]$ denotes their concatenation. C_t is the updated cell state. The terms W represent the weight matrices, b the bias vectors, σ the sigmoid activation function, and $\odot$ denotes element-wise multiplication.

To obtain the final electricity price forecast $\hat{P}_t$, the hidden state h_t of the last LSTM layer is passed through a fully connected dense layer:

$$\hat{P}_t = W_y h_t + b_y \tag{26}$$

where W_y and b_y are the weights and bias of the dense layer. During training, the model learns the optimal weights by minimizing a loss function that computes the difference between the predicted price $\hat{P}_t$ and the actual observed imbalance price P_t.

In this study, the LSTM network has been implemented in Python using the `TensorFlow/Keras` framework. To process the data efficiently, past observationspast observations (also known as lags, e.g., $t - 15$ min, $t - 30$ min) were pre-calculated and added directly as new features in the dataset prior to model training. The dataset was then structured into the standard 3D format required by LSTM networks.

By preparing and locking the data structure beforehand, the optimization algorithm did not have to continuously rebuild the time sequences for every single trial. This approach significantly reduced the overall computational time, which is an essential requirement for achieving the near real-time forecasting goals of this work.

The model configuration relies on hyperparameters that control the network's architecture and learning process. Table 5 provides the list of these hyperparameters along with their respective search spaces, while the detailed optimization procedure is presented in Section 5.1.2.

Hyperparameter	Description	Search Space
Number of LSTM Layers	Determines the depth of the network	1, 2
Number of Units	Dimensionality of the hidden and cell state space	32, 64, 128
Dropout Rate	Fraction of units dropped to prevent overfitting	0.2 - 0.5
Learning Rate	Step size during the gradient descent optimization	0.0001 - 0.01
Batch Size	Number of samples processed before weight update	16, 32, 64

Table 5: List of optimized LSTM hyperparameters with descriptions and search spaces.

5. Results and Discussion

This chapter presents the empirical findings and a systematic evaluation of the proposed predictive models. First, the experimental setup is described, detailing the operational constraints, the feature configuration, and the hyperparameter tuning procedures required to train the algorithms. Subsequently, the statistical metrics employed to assess predictive accuracy are defined. The core of the chapter focuses on the empirical results, beginning with a dedicated analysis of the LEAR model to investigate how the selection of specific features and time lags impacts its predictive performance. Following this, a comparison of all implemented models is provided in terms of forecasting errors, feature importance, and computational efficiency. Finally, an economic assessment is conducted through a simulated trading scenario, evaluating the potential economic viability of the forecasting strategies based on their directional accuracy.

5.1. Experimental Setup and Forecasting Pipeline

Before analyzing the numerical results, it is essential to define the operational constraints and the methodological framework used for training and evaluation. The primary objective of the implemented models is to predict the imbalance price for the immediate next time step, corresponding to the upcoming quarter-hour interval t. To ensure that the forecasting tool is practically viable for market operators, the models operate under near real-time constraints, exclusively utilizing information available up to $t - 10$ minutes before delivery.

To achieve this goal and guarantee a rigorous evaluation, the experimental pipeline applied to each implemented model (with the exception of the Naive baseline) is structured into the following sequential phases:

- **Feature configuration and dataset splitting:** dividing the dataset to simulate a realistic operational environment, explicitly selecting both the exogenous features to include and the specific number of historical time lags for each of them.
- **Hyperparameter optimization:** tuning the specific hyperparameters of each model by applying explicit optimization methodologies, namely the Least Angle Regression algorithm combined with the Akaike Information Criterion for LEAR, and the Tree-structured Parzen Estimator for the other architectures.
- **Training and testing strategies:** establishing how the models are trained and evaluated over the dataset. This involves choosing whether to continuously retrain the algorithm every day or to train it only once depending on its computational cost.
- **Performance evaluation:** quantifying the empirical accuracy using standard statistical metrics supported by statistical significance testing, and conducting a feature importance analysis using the SHapley Additive exPlanations framework to measure the contribution of each input variable.
- **Economic assessment:** evaluating the financial value of the predictions by simulating a trading scenario where a market operator intentionally generates strategic imbalances based on the price forecast.

5.1.1. Feature Configuration and Dataset Splitting

A key characteristic of the data preparation phase is the flexible configuration of the input features. For all exogenous variables, the framework allows selecting multiple time lags as predictors. In other words, for each external variable, the system can use past observations from different previous time steps as separate input features.

To rigorously evaluate the models and simulate a realistic operational environment, the dataset was chronologically divided into two distinct periods:

- **Training and Validation Set (1 April 2022 – 30 September 2024):** this period is dedicated to model training and hyperparameter optimization.
- **Out-of-Sample Test Set (1 October 2024 – 30 September 2025):** this period represents the unseen data used to evaluate the final predictive performance.

5.1.2. Hyperparameter Tuning

In this study, a rigorous hyperparameter tuning procedure was implemented to ensure maximum forecasting accuracy.

For the Lasso Estimated LEAR model, the optimization process focuses exclusively on selecting the optimal regularization penalty λ. To avoid the high computational costs of traditional methods like Grid Search, this study adopts the *Least Angle Regression* (LARS) algorithm combined with the *Akaike Information Criterion*

(AIC) [6]. Instead of testing discrete values through slow iterative trials, the LARS algorithm explores the full range of model configurations in a single pass. The optimal λ is then identified by minimizing the AIC score.

$$AIC = 2k + n \ln(RSS/n) \tag{27}$$

where k represents the number of non-zero coefficients (model complexity), n is the number of observations, and RSS is the Residual Sum of Squares, defined as the squared difference between actual and predicted values. By minimizing this objective function, this method identifies the specific λ that achieves the best compromise between predictive accuracy and simplicity, effectively preventing overfitting. This approach allows for near-instantaneous parameter selection while maintaining full statistical rigor.

For the remaining four models (SVR, Random Forest, XGBoost, and LSTM), hyperparameter tuning was performed using the Tree-structured Parzen Estimator (TPE) algorithm [28], implemented via the Optuna framework [29]. This method was selected specifically to reduce computational time compared to a standard Grid Search. TPE is a Bayesian optimization approach that learns from previous evaluations to focus the search on the most promising hyperparameter combinations. Specifically, it works by modeling two probability distributions to separate good from bad performing configurations, selecting new parameters that are most likely to have better results.
The optimization process was carried out for 1000 trials per model. The hyperparameters and their corresponding search spaces are detailed in the previous sections dedicated to each model. To evaluate the performance of each configuration while respecting the chronological order of the data, a Time Series Cross-Validation with five splits was applied. During this validation process, the Mean Absolute Error (MAE) was chosen as the objective metric to be minimized.

5.1.3. Training and Testing Strategy

Two different training and evaluation strategies were adopted:

1. **Rolling Window with Daily Refit:** For LEAR, Random Forest, XGBoost and SVR models, a rolling window approach with a fixed width of 365 days was implemented. This consists of a training set that shifts forward by one day every 24 hours, including the most recent observations while discarding the oldest ones. The models are retrained daily to learn from this updated historical period. Separately, the prediction for each specific quarter-hour interval t is generated using the most recent data available up to $t - 10$ minutes.
2. **Static Hold-out:** Due to the high computational cost of training deep learning architectures, the LSTM model was evaluated using a static hold-out strategy. The network was trained only once on the initial training set and its parameters remain fixed throughout the entire test period.

5.1.4. Performance Evaluation Metrics

To assess the predictive accuracy and reliability of the developed models, four standard statistical metrics were employed.
The Mean Absolute Error (MAE) measures the average magnitude of the errors in a set of predictions, without considering their direction:

$$MAE = \frac{1}{n} \sum_{t=1}^{n} |P_t - \hat{P}_t| \tag{28}$$

where n is the number of observations, P_t is the actual price at time t, and $\hat{P}_t$ is the predicted price. Expressed in €/MWh, a low MAE indicates good average accuracy, while a high value suggests that the model's predictions are, on average, far from the real market prices. The Root Mean Square Error (RMSE) is calculated as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (P_t - \hat{P}_t)^2} \tag{29}$$

Also measured in €/MWh, the RMSE attributes a significantly higher weight to large errors compared to small ones due to the squaring of the residuals. Consequently, a high RMSE value relative to the MAE indicates that the model is prone to making occasional but very large forecasting mistakes. The Symmetric Mean Absolute Percentage Error (sMAPE) expresses the forecast error as a percentage of the price level, measuring the relative accuracy of the model:

$$sMAPE = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|P_t - \hat{P}_t|}{(|P_t| + |\hat{P}_t|)/2} \tag{30}$$

This metric provides an intuitive understanding of the error magnitude: for example, an sMAPE of 10% means that, on average, the predicted price deviates by 10% from the actual value. It ranges from 0% (perfect prediction) to 200%. The Coefficient of Determination (R^2) provides a measure of how well the model follows the actual price movements by quantifying the proportion of the total variance in the target variable explained by the model:

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(P_t - \hat{P}_t)^2}{\sum_{t=1}^{n}(P_t - \bar{P})^2} \tag{31}$$

where $\bar{P}$ is the mean of the actual prices. This coefficient can range from 1 to negative infinity. A value of 1 indicates a perfect prediction, where the model follows the price curve exactly. A value of 0 means the model is only as good as predicting the average price failing to catch any specific price change. If R^2 is negative, the model is performing worse than the average price, meaning its predictions are misleading and do not represent the market trend.

In addition to the standard performance metrics, a statistical test was conducted to evaluate whether the differences in predictive accuracy are statistically significant. The Diebold-Mariano (DM) test is used to assess this by testing the null hypothesis (H_0) that the expected value of the loss differentials is zero. This procedure is applied to both MAE and RMSE metrics, ensuring that the observed improvements are not merely due to random fluctuations in the data.

Beyond numerical accuracy, the SHAP (SHapley Additive exPlanations) framework was employed to measure feature importance [30]. SHAP is an interpretability tool based on cooperative game theory that computes the contribution of each input variable to the final prediction. This method was selected to ensure a consistent analytical framework for quantifying feature influence across all models, regardless of their architecture. Using SHAP makes it possible to identify the specific drivers for each model, highlighting how different algorithms interpret market dynamics and ensuring transparency in the evaluation of their internal logic. In this work, to allow for a clearer comparison, these values are expressed as relative percentages, calculated by normalizing the mean absolute SHAP value of each feature against the total impact of all predictors.

5.2. Empirical Results

5.2.1. LEAR Preliminary Analysis

For the LEAR model, as illustrated in Tables 6 to 26, the forecasting results were compared by combining the price proxy variable (preliminary imbalance price) with one additional exogenous variable at a time. Simultaneously, different configurations regarding the number of selected historical lags were evaluated. Specifically, the following lag setups were considered:

- **Best Lag**: the single best-performing lag for each variable, as detailed in Table 1.
- **Up to $t - 60$ min**: all available lags up to one hour before real-time.
- **Up to $t - 120$ min**: all available lags up to two hours before real-time.
- **Up to $t - 1$ day**: all available lags up to one day before real-time.

It can be observed that the performance are generally inferior to the Naive Baseline in terms of MAE and sMAPE, whereas it systematically improves the RMSE. This suggests that while a single isolated lag may not refine the average prediction error, it helps the model mitigate larger price deviations. Conversely, providing the algorithm with a deeper sequence of recent historical data (Table 7 and Tables 25, 26 in Appendix A) consistently improves the accuracy, allowing the LEAR model to outperform the Naive benchmark across all evaluated metrics.
However, the results indicate a saturation effect regarding the informational value of historical depth. There are no appreciable improvements when extending the lag window from 60 minutes (Table 7) to 120 minutes (Table 25) before real-time. Furthermore, including lags up to the previous day (Table 26) negatively impacts the performance.
Finally, as expected, it can be noted that market-based price variables (P_{XBID}^{last}, followed by P_{MI-A}^{mean} and P_z^{MGP}) achieve better performance compared to the physical variables. Among them, the preliminary imbalance volume $S_{mz,prel}$ emerges as the best feature.
Additionally, to determine the optimal number and specific set of exogenous variables, a forward feature selection approach was employed. Starting from the preliminary imbalance price, the most promising variables identified in the previous step were incrementally added to the model. For all configurations, the setup including all

available historical lags up to one hour before real-time ($t - 60$ min) was maintained. The results of this progressive integration are presented in Table 8.
The optimal result is achieved by combining the baseline regressive variable ($P^{sb}_{mz,prel}$) with the three market price features (P^{last}_{XBID}, P^{mean}_{MI-A}, and P^{MGP}_{z}) and the preliminary imbalance volume ($S_{mz,prel}$), which represents the physical driver of the target variable ($P^{sb}_{mz,prel}$). Compared to the Naive Baseline, this specific combination provides an improvement of 9.66% in MAE and 16.71% in RMSE. All other feature combinations present very similar results, with the exception of the setup including only two variables and the one incorporating all available features. Finally, the last row highlights the significant impact of the autoregressive variable $P^{sb}_{mz,prel}$ on the optimal configuration: removing it brings the MAE close to the level of the Naive model.

Feature	**MAE [€/MWh]**	**RMSE [€/MWh]**	**sMAPE [%]**	**R^2**
Naive Baseline ($P^{sb}_{mz,prel}(t-60)$)	54.16	86.50	52.12	-0.05
$P^{sb}_{mz,prel}$, P^{MGP}_{z}	56.18	77.14	54.58	0.16
$P^{sb}_{mz,prel}$, P^{mean}_{MI-A}	55.36	76.54	53.91	0.17
$P^{sb}_{mz,prel}$, P^{last}_{XBID}	54.55	75.98	53.41	0.18
$P^{sb}_{mz,prel}$, $S_{mz,prel}$	57.89	80.27	57.22	0.09
$P^{sb}_{mz,prel}$, Err^{sw}_{gen}	58.83	81.16	57.15	0.07
$P^{sb}_{mz,prel}$, L^{f}_{tot}	58.44	80.69	57.22	0.08
$P^{sb}_{mz,prel}$, G^{f}_{tot}	58.18	80.43	57.53	0.09
$P^{sb}_{mz,prel}$	58.42	81.08	56.84	0.07

Table 6: Forecasting performance of the LEAR model with imbalance preliminary price and single exogenous variables using the single best-performing lag.

Feature	**MAE [€/MWh]**	**RMSE [€/MWh]**	**sMAPE [%]**	**R^2**
Naive Baseline ($P^{sb}_{mz,prel}(t-60)$)	54.16	86.50	52.12	-0.05
$P^{sb}_{mz,prel}$, P^{MGP}_{z}	51.38	72.74	50.52	0.25
$P^{sb}_{mz,prel}$, P^{mean}_{MI-A}	51.32	73.08	50.40	0.25
$P^{sb}_{mz,prel}$, P^{last}_{XBID}	50.83	72.94	50.11	0.25
$P^{sb}_{mz,prel}$, $S_{mz,prel}$	51.98	75.26	51.80	0.20
$P^{sb}_{mz,prel}$, Err^{sw}_{gen}	53.28	75.80	52.26	0.19
$P^{sb}_{mz,prel}$, L^{f}_{tot}	52.73	75.39	52.15	0.20
$P^{sb}_{mz,prel}$, G^{f}_{tot}	53.41	75.86	52.53	0.19
$P^{sb}_{mz,prel}$	52.93	75.81	52.01	0.19

Table 7: Forecasting performance of the LEAR model with the imbalance preliminary price and single exogenous variables using all available lags up to $t - 60$ min.

Features	MAE [€/MWh]	RMSE [€/MWh]	sMAPE [%]	R^2
Naive Baseline ($P^{sb}_{mz,prel}(t-60)$)	54.16	86.50	52.12	-0.05
$P^{sb}_{mz,prel}$, P^{last}_{XBID}	54.55	75.98	53.41	0.18
$P^{sb}_{mz,prel}$, P^{last}_{XBID}, P^{mean}_{MI-A}	50.51	72.40	49.76	0.26
$P^{sb}_{mz,prel}$, P^{last}_{XBID}, P^{mean}_{MI-A}, P^{MGP}_{z}	50.24	72.17	49.58	0.27
$P^{sb}_{mz,prel}$, P^{last}_{XBID}, P^{mean}_{MI-A}, P^{MGP}_{z}, $S_{mz,prel}$	**48.93**	**72.05**	**48.80**	**0.27**
$P^{sb}_{mz,prel}$, P^{last}_{XBID}, P^{mean}_{MI-A}, P^{MGP}_{z}, $S_{mz,prel}$, L^{f}_{tot}	49.35	72.47	49.14	0.26
$P^{sb}_{mz,prel}$, P^{last}_{XBID}, P^{mean}_{MI-A}, P^{MGP}_{z}, $S_{mz,prel}$, L^{f}_{tot}, Err^{sw}_{gen}	49.74	72.69	49.29	0.25
$P^{sb}_{mz,prel}$, P^{last}_{XBID}, P^{mean}_{MI-A}, P^{MGP}_{z}, $S_{mz,prel}$, L^{f}_{tot}, Err^{sw}_{gen}, G^{f}_{tot}	52.15	73.01	50.02	0.23
P^{last}_{XBID}, P^{mean}_{MI-A}, P^{MGP}_{z}, $S_{mz,prel}$	53.60	75.63	54.25	0.19

Table 8: Forecasting performance of the LEAR model with progressive feature integration using lags up to $t-60$ min. The best-performing configuration is highlighted in bold.

Figure 4 illustrates the feature importance analysis based on the SHAP method, expressed as relative percentage contributions. The results are derived from the model trained with the optimal configuration and lags up to $t-60$ min, as identified in Table 8.
Contrary to what might have been initially expected, the most important variable is P^{mean}_{MI-A}, which accounts for 32.1% of the model's predictive decisions, followed by the autoregressive proxy variable of the preliminary imbalance price ($P^{sb}_{mz,prel}$) at 27.0%. Interestingly, although P^{last}_{XBID} achieved the best forecasting performance when evaluated as a single exogenous feature, it emerges as the least impactful market variable in the combined model setup, just above the day-of-the-week dummy variable.

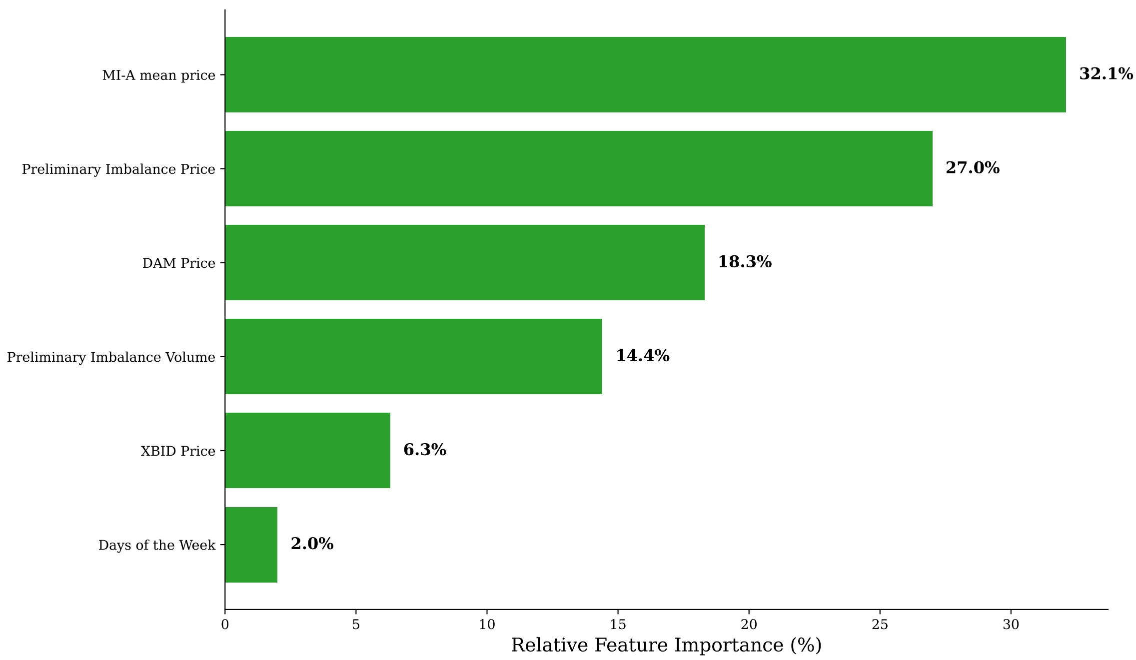

Figure 4: Feature importance analysis of the LEAR model.

5.2.2. Model Performance

For each model considered in this work the performance were evaluated using different temporal input lags for all exogenous variables as shown in Tables 9 to 13:

- The *best lags* for each variable, as reported in Table 1 in Section 3;
- The lags available within one hour from real-time ($t - 60$ min);
- The lags available within two hours from real-time ($t - 120$ min).

For the LEAR model, the case including lags up to the previous day (Up to D) was also analyzed. For the other predictive models, in fact, handling such a high number of features would be computationally too expensive while still maintaining the ability to compute forecasts close to real-time.
Furthermore, it should be noted that, exclusively for the LEAR model, the term "Optimal config." refers to the optimal configuration composed of the 5 variables ($P^{sb}_{mz,prel}$, P^{last}_{XBID}, P^{mean}_{MI-A}, P^{MGP}_{z}, $S_{mz,prel}$), as described and justified in Section 5.2.1.

It can be observed that for all models, performance improves when transitioning from a single lag to a multi-lag configuration. Extending the input horizon up to two hours, that increases the computational cost, results in performance variations that are not statistically significant, showing either marginal improvements or slight worsening depending on the specific model.
The last row of each table illustrates the performance impact of excluding the preliminary imbalance price variable $P^{sb}_{mz,prel}$. The RF and SVR models demonstrate greater robustness to this omission, maintaining performance levels that are even superior to their respective single best lag scenarios. In contrast, for the other models, most notably the LSTM, performance degrades significantly, approaching the values of the naive baseline.

Exogenous features	**Lag**	**MAE [€/MWh]**	**RMSE [€/MWh]**	**sMAPE [%]**	$\mathbf{R^2}$
Naive Baseline	t-60 min	54.16	86.50	52.12	-0.05
Optimal config.	Best lag	52.65	75.41	52.11	0.20
Optimal config.	Up to t-60 min	48.93	72.05	48.80	0.27
Optimal config.	Up to t-120 min	48.84	72.05	48.69	0.27
Optimal config.	Up to D	49.79	72.73	49.51	0.25
Optimal config. w/o $P^{sb}_{mz,prel}$	Up to t-60 min	53.60	75.63	54.25	0.19

Table 9: Forecasting performance of the LEAR model with optimal feature configuration and different lags.

Exogenous features	**Lag**	**MAE [€/MWh]**	**RMSE [€/MWh]**	**sMAPE [%]**	$\mathbf{R^2}$
Naive Baseline	t-60 min	54.16	86.50	52.12	-0.05
All features	Best lag	50.79	68.12	50.04	0.34
All features	Up to t-60 min	47.99	65.91	47.60	0.39
All features	Up to t-120 min	47.85	65.85	47.41	0.39
All features w/o $P^{sb}_{mz,prel}$	Up to t-60 min	49.37	66.64	49.19	0.37

Table 10: Forecasting performance of the RF model with all features and different lags.

Exogenous features	Lag	MAE [€/MWh]	RMSE [€/MWh]	sMAPE [%]	R^2
Naive Baseline	t-60 min	54.16	86.50	52.12	-0.05
All features	Best lag	47.18	70.58	46.57	0.29
All features	Up to t-60 min	43.77	67.63	43.57	0.35
All features	Up to t-120 min	44.06	67.73	43.72	0.35
All features w/o $P^{sb}_{mz,prel}$	Up to t-60 min	51.77	73.56	51.80	0.24

Table 11: Forecasting performance of the XGBoost model with all features and different lags.

Exogenous features	Lag	MAE [€/MWh]	RMSE [€/MWh]	sMAPE [%]	R^2
Naive Baseline	t-60 min	54.16	86.50	52.12	-0.05
All features	Best lag	48.54	72.84	47.67	0.25
All features	Up to t-60 min	45.92	69.23	45.46	0.32
All features	Up to t-120 min	46.23	69.32	45.87	0.32
All features w/o $P^{sb}_{mz,prel}$	Up to t-60 min	47.93	69.52	48.27	0.32

Table 12: Forecasting performance of the SVR model with all features and different lags.

Exogenous features	Lag	MAE [€/MWh]	RMSE [€/MWh]	sMAPE [%]	R^2
Naive Baseline	t-60 min	54.16	86.50	52.12	-0.05
All features	Best lag	50.87	72.80	50.98	0.25
All features	Up to t-60 min	48.66	70.32	48.12	0.30
All features	Up to t-120 min	48.74	70.00	48.61	0.30
All features w/o $P^{sb}_{mz,prel}$	Up to t-60 min	54.83	76.78	53.92	0.17

Table 13: Forecasting performance of the LSTM model with all features and different lags.

Table 14 compares the performance of the different models using exogenous features with lags up to $t - 60$ min. This specific configuration was chosen to allow for a direct comparison under consistent settings and temporal intervals. Simultaneously, this approach minimizes the computational cost while maintaining the same predictive performance.
The results show that XGBoost is the best-performing model in terms of MAE, recording an improvement of approximately 19.18% compared to the Naive Baseline. This means that, in absolute terms, the predictions made by XGBoost deviate less on average from the actual values, providing the most accurate typical estimate for a single forecast. Additionally, the Random Forest model performs best in terms of RMSE, showing a notable improvement of 23.80% compared to the benchmark. Since the RMSE metric penalizes larger errors, this result indicates that the RF model is the most effective at limiting the frequency and magnitude of large deviations, thus proving more robust against price spikes.
It should be noted, however, that for all evaluated models, the values of the coefficient of determination (R^2) remain moderate, despite showing a clear improvement over the baseline. The maximum value achieved is 0.39 with the RF model, indicating that the model can explain only 39% of the variance in the data. This highlights the lack of a strong linear relationship between the predictions and the actual imbalance prices, confirming the stochastic and highly unpredictable nature of this variable.
Finally, the analysis of the sMAPE further reflects the inherent difficulty of the forecasting task. The best value obtained, 43.57% with XGBoost, shows that the average percentage error remains considerable. In practical terms, this means that even the best-performing model produces forecasts that, on average, deviate by more than 43% from the actual value, with this error being even higher for the other evaluated models.
Following the LEAR model, the LSTM network exhibits the poorest performance among the evaluated models, although it still achieves an improvement of approximately 10.16% in MAE and 18.71% in RMSE compared to

the baseline. In light of these overall results, and given the optimal balance achieved between computational efficiency and predictive accuracy, this $t - 60$ min configuration is adopted as the reference configuration for all subsequent comparisons and further analyzes presented in the remainder of this work.

Model	MAE [€/MWh]	RMSE [€/MWh]	sMAPE [%]	R^2
Naive Baseline	54.16	86.50	52.12	-0.05
LEAR	48.93	72.05	48.80	0.27
RF	47.99	**65.91**	47.60	**0.39**
XGBoost	**43.77**	67.63	**43.57**	0.35
SVR	45.92	69.23	45.46	0.32
LSTM	48.66	70.32	48.12	0.30

Table 14: Comparison of all models using lags up to $t - 60$ min. The best-performing metrics are highlighted in bold.

Figures 5–9 illustrate the trend of the forecasted imbalance prices compared to the actual values over the entire duration of the test set.

A more detailed view is presented in Figure 10, which compares the trends of the predictions generated by the LEAR and RF models from March 29 to March 31. It can be observed that these models, reflecting the general behavior of all the evaluated algorithms, successfully capture the overall trend of the target variable, accurately predicting the time intervals when imbalance prices are stable and low. However, they all struggle to forecast sudden changes and extreme price spikes. Furthermore, it is visually evident from the comparison that the RF model manages to estimate the instances of high prices slightly better, which consequently explains its lower RMSE.

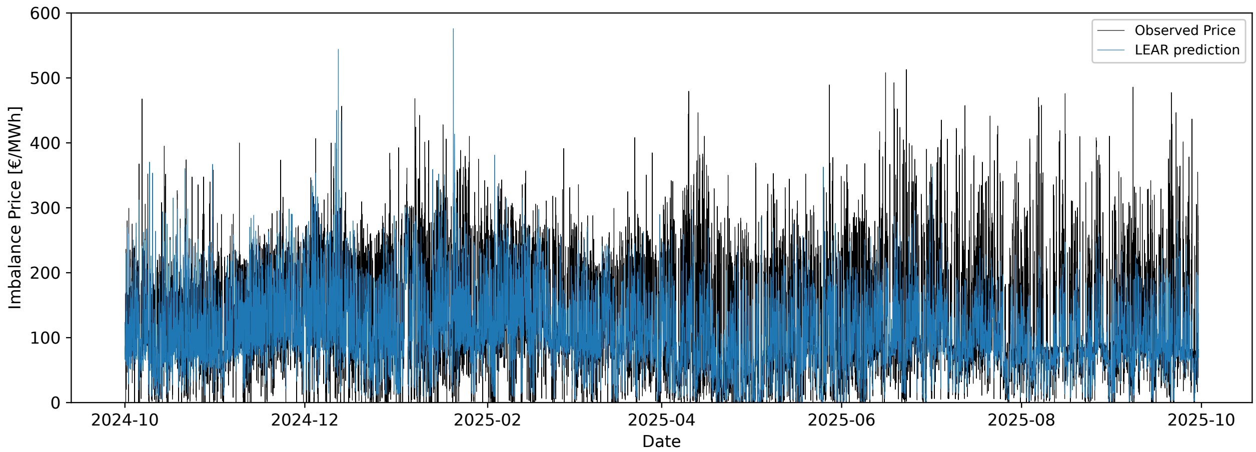

Figure 5: Overall forecasting trend on the test set for the LEAR model compared to the actual imbalance prices.

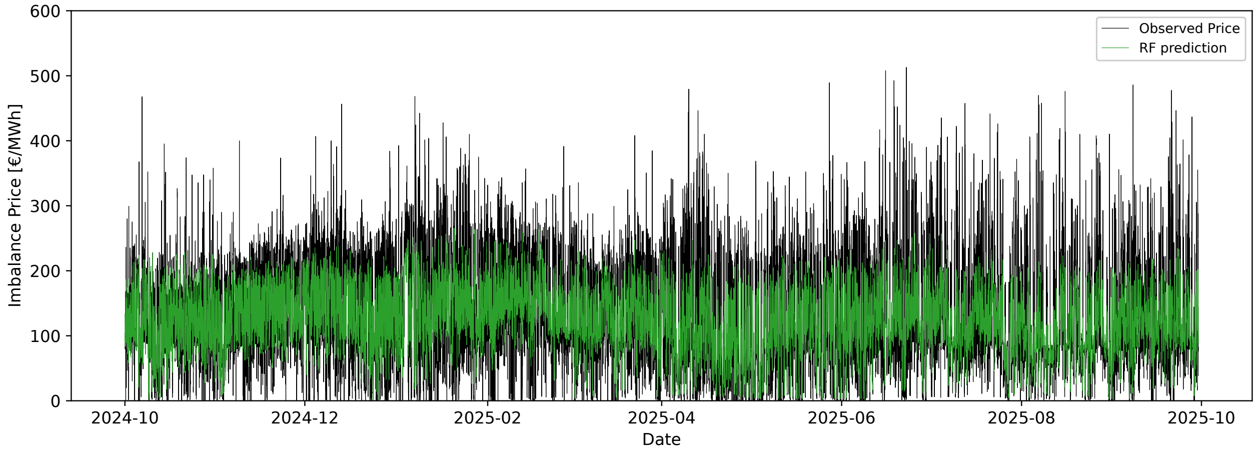

Figure 6: Overall forecasting trend on the test set for the Random Forest (RF) model compared to the actual imbalance prices.

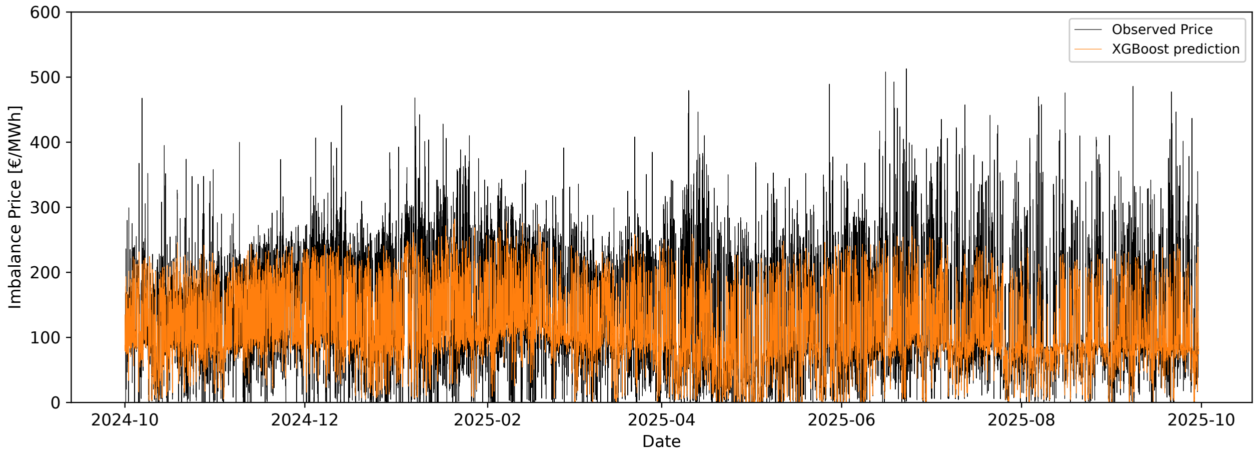

Figure 7: Overall forecasting trend on the test set for the XGBoost model compared to the actual imbalance prices.

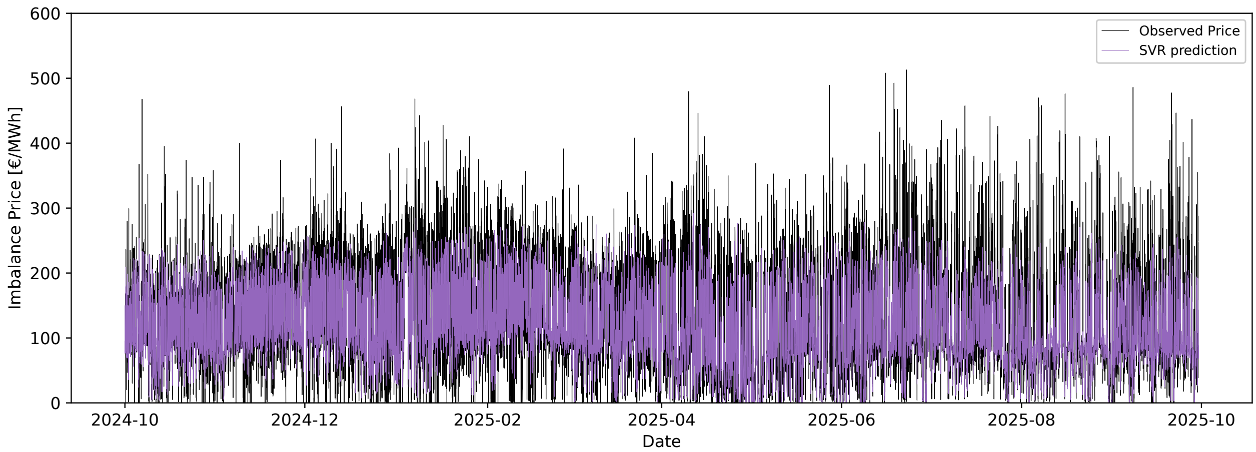

Figure 8: Overall forecasting trend on the test set for the SVR model compared to the actual imbalance prices.

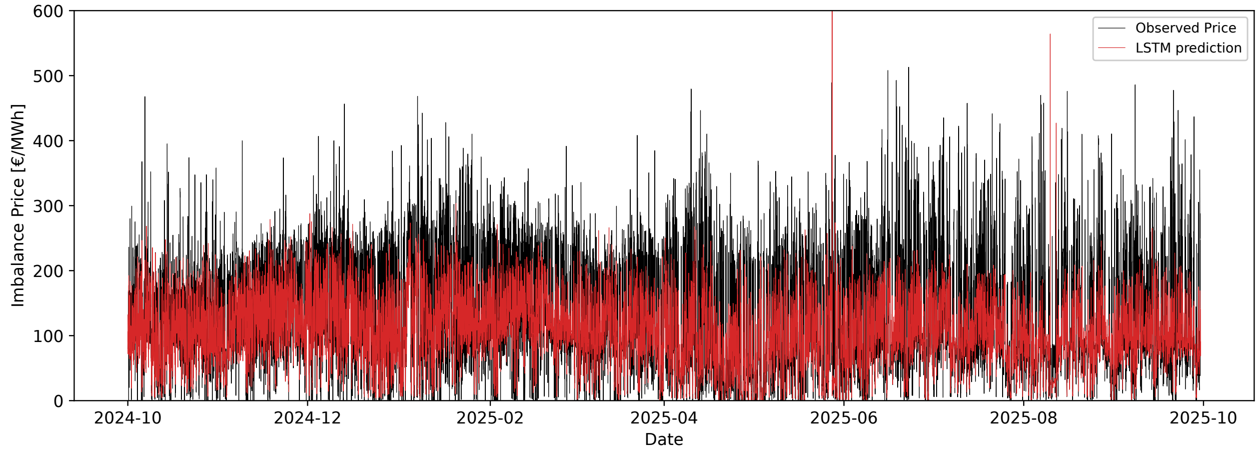

Figure 9: Overall forecasting trend on the test set for the LSTM model compared to the actual imbalance prices.

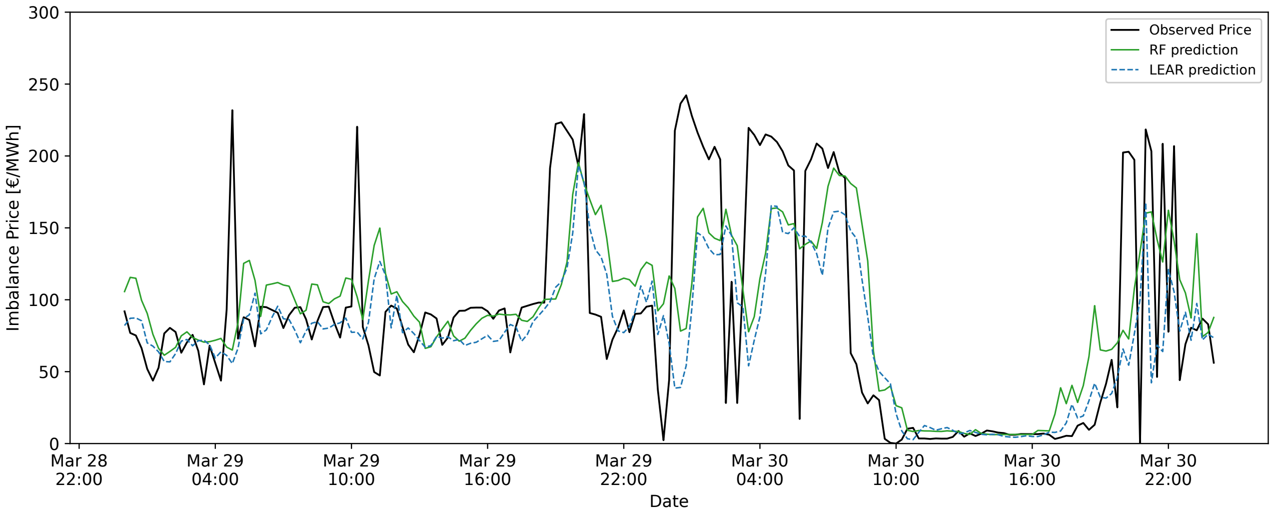

Figure 10: Detailed comparison of the forecasted imbalance prices by the LEAR and RF models against the actual values from March 29 to March 31.

Table 15 reports the feature importance results obtained for each evaluated model, expressed in percentage values. The top three most influential variables for each algorithm are highlighted in bold.
Random Forest and XGBoost exhibit very similar and highly intuitive decision patterns: their most important features are the preliminary imbalance price $P^{sb}_{mz,prel}$ and its corresponding physical driver, the preliminary imbalance volume $S_{mz,prel}$, closely followed by the other market price variables. Conversely, the aggregated physical features, namely the total forecasted load L^{f}_{tot} and the total forecasted generation G^{f}_{tot}, prove to be highly relevant for the SVR and LSTM models, respectively. In these architectures, the impact of the preliminary market variables is significantly reduced, becoming almost negligible in the case of the SVR model.
Furthermore, the forecasting error of renewable generation Err^{sw}_{gen} shows a generally limited impact across the models. It is hypothesized that this behavior is primarily due to the chosen lag configuration (up to $t-60$ min), in which only a single historical value of this specific feature is available, thereby limiting its overall informative power. Finally, as might be expected, both the rolling statistics and the time-related features demonstrate a very marginal impact on the overall predictive processes across all evaluated models.

	Features									
Model	$P^{sb}_{mz,prel}$	P^{MGP}_{z}	P^{mean}_{MI-A}	P^{last}_{XBID}	$S_{mz,prel}$	Err^{sw}_{gen}	L^{f}_{tot}	G^{f}_{tot}	**Rolling**	**Time**
RF	**33.40**	6.50	**12.10**	5.40	**33.00**	0.40	1.30	5.20	1.60	1.10
XGBoost	**38.00**	9.09	**11.67**	4.78	**22.14**	0.96	2.27	6.53	2.32	2.24
SVR	1.80	**10.17**	4.48	0.70	7.09	9.58	**41.93**	**23.55**	0.01	0.69
LSTM	**14.23**	1.87	2.99	1.50	8.47	1.18	**11.61**	**57.17**	0.42	0.56

Table 15: Feature importance values (expressed in percentages) for the evaluated models

Table 16 presents the Diebold-Mariano test results. For the RMSE, the differences between all pairs of models are statistically significant. Regarding the MAE, the results are also significant except between LEAR and LSTM.

Model	LEAR	RF	XGB	SVR	LSTM	Naive
LEAR	-	✓	✓	✓	X	✓
RF		-	✓	✓	✓	✓
XGBoost			-	✓	✓	✓
SVR				-	✓	✓
LSTM					-	✓
Naive						-

Model	LEAR	RF	XGB	SVR	LSTM	Naive
LEAR	-	✓	✓	✓	✓	✓
RF		-	✓	✓	✓	✓
XGBoost			-	✓	✓	✓
SVR				-	✓	✓
LSTM					-	✓
Naive						-

Table 16: Diebold-Mariano test results for MAE (left) and RMSE (right). A checkmark (✓) indicates a statistically significant difference in predictive accuracy (p-value < 0.05), while X denotes no significant difference.

Table 17 details the computational cost required by each evaluated model, reporting the execution times for the daily retraining process (in seconds) and the generation of a single quarter-hour prediction (in milliseconds). All computational tests were performed on a machine equipped with an AMD EPYC 7502P processor (32 physical cores, 64 threads, up to 2.5 GHz) and 125 GB of RAM. All models were trained and evaluated using CPU resources only, as no GPU acceleration was available on the system.
As might be expected, the LEAR model emerges as the most computationally efficient algorithm across both metrics, while the execution times progressively increase as the underlying machine learning and deep learning architectures become more complex. Finally, the daily retraining time for the LSTM model is omitted, as a static hold-out approach was adopted for its training phase instead of a daily refit.

Model	Daily Retraining Time [s]	Prediction Time [ms]
LEAR	6.82	1.53
RF	10.01	10.25
XGBoost	11.40	10.32
SVR	23.12	18.02
LSTM	—	59.01

Table 17: Computational times for the evaluated models, detailing the daily retraining duration and the single-step prediction time.

5.3. Economic Results

Beyond evaluating the statistical point-forecasting performance of the implemented models, an economic assessment was conducted to quantify the practical and financial impact of the predictions. To this end, a simulated case study was designed based on a hypothetical scenario.

Consider a market operator managing a fully flexible asset, such as a Battery Energy Storage System (BESS), with a predefined commercial baseline on the Day-Ahead Market. In a specific quarter-hour q, the operator has sold a baseline power volume P_{sold} at the zonal clearing price P_z^{MGP}. It is assumed that the operator has 1 MW of fully available capacity to intentionally deviate from the scheduled baseline in real-time, either upwards or downwards.

The operator relies on the model's forecasted imbalance price $\hat{P}_{mz}^{sb}$ to decide whether to strategically deviate from the DAM schedule. The decision-making logic is structured as follows:

- **Forecasting a Negative Macrozonal Imbalance ($\hat{P}_{mz}^{sb} > P_z^{MGP}$):** the operator anticipates a negative macrozonal imbalance. Consequently, the operator decides to inject 1 MW of additional power into the grid compared to the baseline P_{sold}. The economic rationale is to sell this marginal megawatt at the expected higher imbalance price rather than the lower day-ahead price. The realized actual profit for the quarter-hour is mathematically formulated as:

$$Actual\ Profit = 1\ \text{MW} \times 0.25\ \text{h} \times (P_{mz}^{sb} - P_z^{MGP})$$

- **Forecasting a Positive Macrozonal Imbalance ($\hat{P}_{mz}^{sb} < P_z^{MGP}$):** the operator anticipates a positive macrozonal imbalance. In this scenario, the operator opts to inject 1 MW less than the contracted baseline P_{sold}. The strategy relies on having already sold the power at a higher price on the MGP, and then settling the resulting energy deficit by "buying it back" as a real-time imbalance at the expected lower price. The realized actual profit is formulated as:

$$Actual\ Profit = 1\ \text{MW} \times 0.25\ \text{h} \times (P_z^{MGP} - P_{mz}^{sb})$$

Naturally, while the decision is triggered by the forecasted price $\hat{P}_{mz}^{sb}$, the actual economic outcome of this strategy depends entirely on the realized real-time imbalance price P_{mz}^{sb}. If the model's directional prediction is incorrect, the actual profit turn into a financial penalty.

To evaluate this risk, the confusion matrices presented in Tables 18 to 23 were computed. These matrices illustrate the models' capability to correctly anticipate whether the real-time imbalance price will be higher or lower than the MGP clearing price, which represents the fundamental driver of the proposed strategy. By aggregating the percentages of the true positive and true negative quadrants, it is possible to determine the overall directional accuracy of the forecasting algorithms.

The best performance is achieved by the XGBoost model with a total accuracy of 76.81%, followed by the LSTM (75.21%), LEAR (75.13%), SVR (74.97%), and RF (73.94%) models. Finally, the Naive benchmark model records an overall accuracy of 73.34%, a result very close to the others.

	$\hat{P}^{sb}_{mz} \geq P^{MGP}_z$	$\hat{P}^{sb}_{mz} < P^{MGP}_z$
$P^{sb}_{mz} \geq P^{MGP}_z$	27.97%	13.35%
$P^{sb}_{mz} < P^{MGP}_z$	13.31%	45.37%

Table 18: Confusion matrix for the Naive model

	$\hat{P}^{sb}_{mz} \geq P^{MGP}_z$	$\hat{P}^{sb}_{mz} < P^{MGP}_z$
$P^{sb}_{mz} \geq P^{MGP}_z$	25.65%	15.67%
$P^{sb}_{mz} < P^{MGP}_z$	9.20%	49.48%

Table 19: Confusion matrix for the LEAR model

	$\hat{P}^{sb}_{mz} \geq P^{MGP}_z$	$\hat{P}^{sb}_{mz} < P^{MGP}_z$
$P^{sb}_{mz} \geq P^{MGP}_z$	32.74%	8.58%
$P^{sb}_{mz} < P^{MGP}_z$	17.48%	41.20%

Table 20: Confusion matrix for the RF model

	$\hat{P}^{sb}_{mz} \geq P^{MGP}_z$	$\hat{P}^{sb}_{mz} < P^{MGP}_z$
$P^{sb}_{mz} \geq P^{MGP}_z$	27.82%	13.50%
$P^{sb}_{mz} < P^{MGP}_z$	9.69%	48.99%

Table 21: Confusion matrix for the XGB model

	$\hat{P}^{sb}_{mz} \geq P^{MGP}_z$	$\hat{P}^{sb}_{mz} < P^{MGP}_z$
$P^{sb}_{mz} \geq P^{MGP}_z$	28.13%	13.19%
$P^{sb}_{mz} < P^{MGP}_z$	11.84%	46.84%

Table 22: Confusion matrix for the SVR model

	$\hat{P}^{sb}_{mz} \geq P^{MGP}_z$	$\hat{P}^{sb}_{mz} < P^{MGP}_z$
$P^{sb}_{mz} \geq P^{MGP}_z$	26.24%	15.08%
$P^{sb}_{mz} < P^{MGP}_z$	9.71%	48.97%

Table 23: Confusion matrix for the LSTM model

Table 24 provides a concise overview of the financial performance achieved over a one-year period, which coincides with the test set, using the simulated trading strategy. The table details gross revenues, financial losses, and the total profit. Random Forest emerges as the most profitable model, generating a total actual profit of € 307,449.02 (+24.28% compared to the Naive baseline), followed by the XGBoost model. The SVR and LSTM architectures achieve intermediate results, while the LEAR model exhibits an economic performance highly similar to the simple baseline, which nonetheless maintains a comparable and solid profit.

Model	**Gross Revenues [€]**	**Losses [€]**	**Profit [€]**
Naive	412,981.20	-165,611.92	247,369.28
LEAR	417,419.06	-161,745.51	255,673.55
RF	**443,306.79**	**-135,857.77**	**307,449.02**
XGBoost	432,103.47	-147,061.09	285,042.38
SVR	427,115.45	-152,049.12	275,066.33
LSTM	424,469.06	-154,668.51	269,827.55

Table 24: Annual economic results of the simulated trading strategy for each evaluated model.

These results reveal a counterintuitive finding: although the RF model exhibited the second lowest overall directional accuracy among all evaluated models (73.94%), it significantly outperforms them in terms of economic profitability.
This discrepancy is primarily driven by the pronounced asymmetry of the imbalance price. To better understand this dynamic, we define the price spread as the arithmetic difference between the realized real-time imbalance price and the Day-Ahead Market zonal price ($P^{sb}_{mz} - P^{MGP}_z$). An analysis of the dataset reveals that positive price spreads ($P^{sb}_{mz} > P^{MGP}_z$) occur in only 39.10% of the evaluated quarter-hours, exhibiting an average value of 91.98 €/MWh. Conversely, negative spreads ($P^{sb}_{mz} < P^{MGP}_z$) are much more frequent, occurring 58.68% of the time, but are financially less impactful, with an average value of -51.68 €/MWh.
To visually analyze this behavior, Figures 11, 12, and 13 report the average accuracy, total gross revenues, and total financial losses, respectively. The data is segmented into ten deciles based on the price spread, ordered from the most negative to the most positive.

As evident in Figure 11, while the RF model generally underperforms compared to the others in the deciles associated with negative or near-zero spreads, its predictive accuracy sharply increases when the spread turns positive. Consequently, the RF model successfully captures the most lucrative market opportunities, maximizing revenues and minimizing severe financial penalties precisely when the price spread is at its highest. This demonstrates that, in automated trading strategies, predicting high-value events with great precision is far more profitable than maintaining a uniformly high accuracy across low-value market conditions.

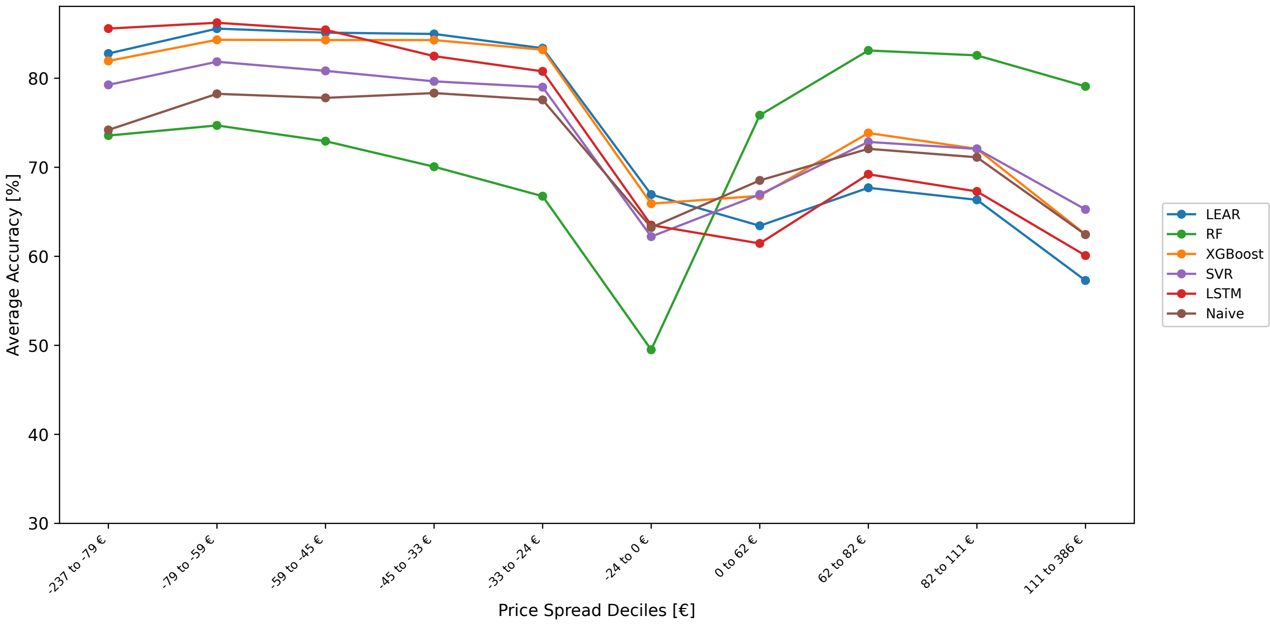

Figure 11: Average directional accuracy of the forecasting models across ten price spread deciles.

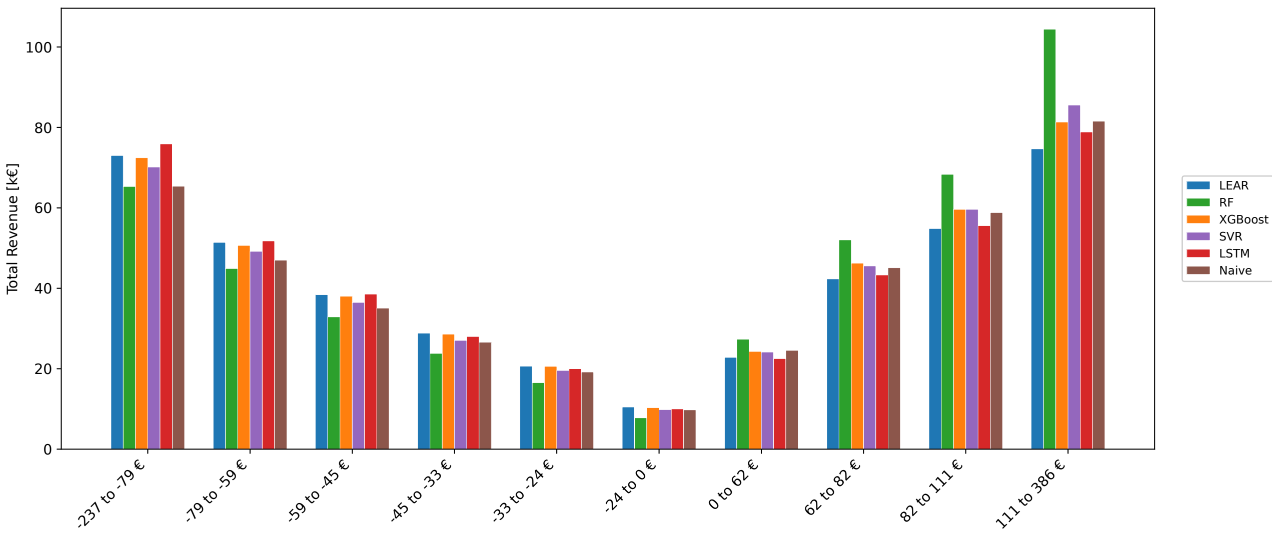

Figure 12: Total gross revenues generated by each model across the ten price spread deciles.

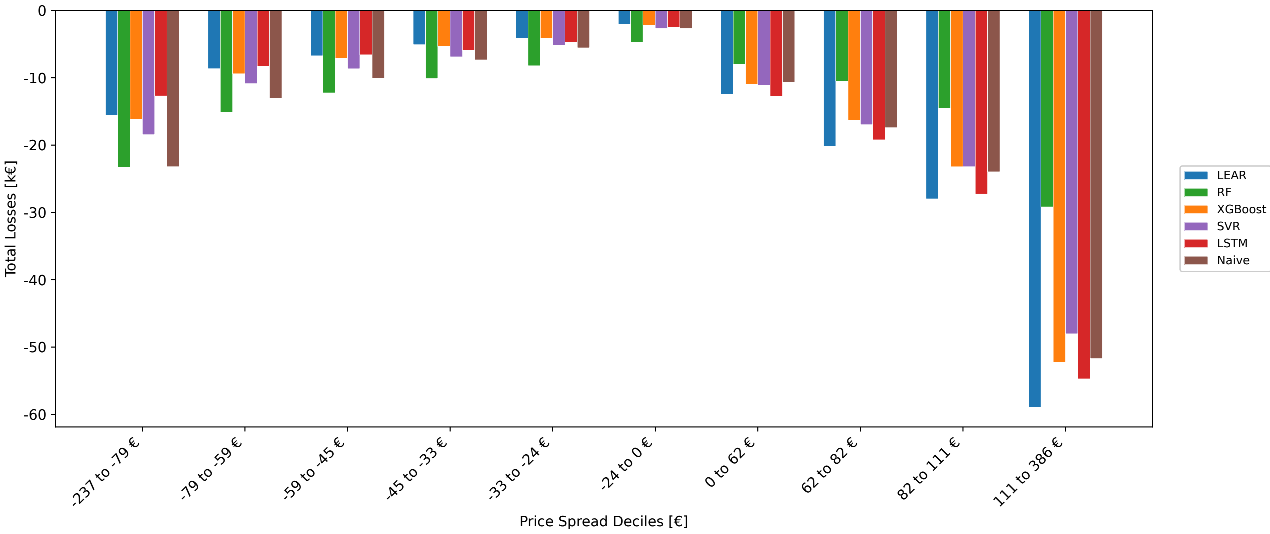

Figure 13: Total financial losses incurred by each model across the ten price spread deciles.

To further investigate the temporal distribution of these profits, Figure 14 illustrates the total annual revenue generated by the Random Forest model across the 96 quarter-hourly intervals of the day. The chart clearly shows that positive economic returns are achieved in every single interval. This indicates that the proposed strategic imbalance approach is profitable regardless of the specific time of day. All other implemented models exhibit this exact same behaviour.

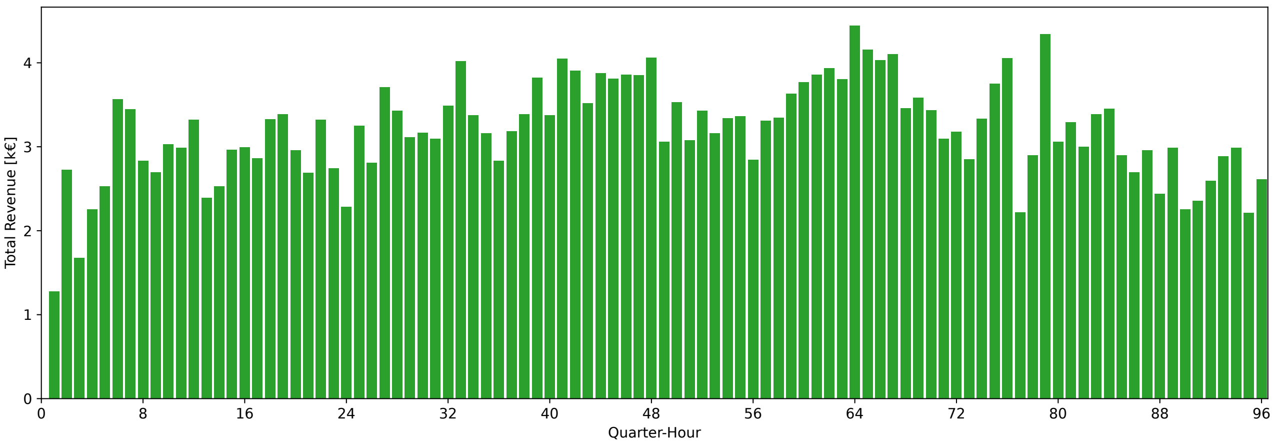

Figure 14: Total revenue generated by the RF model distributed across the 96 daily quarter-hour intervals.

6. Conclusions

The aim of this work was to provide a comparative benchmark of predictive models for near real-time forecasting of imbalance prices within the Italian electricity market, specifically focusing on the North macrozone. The framework was designed to be utilized by market participants requiring reliable price signals under strict temporal constraints, executing forecasts approximately 10 minutes before the delivery period ($t - 10$). To achieve this, the study implemented and systematically compared five distinct architectures: a statistical linear model (LEAR), three Machine Learning algorithms (Random Forest, XGBoost, SVR), and a Deep Learning network (LSTM). The objective was to assess not only the statistical accuracy of these models but also the potential economic value of their predictions through a simulated trading scenario, evaluating the effectiveness of the forecasting strategies in an operational context.

The empirical results showed that Machine Learning ensemble methods are the most effective architectures for this forecasting task. In particular, XGBoost emerged as the best-performing model in terms of Mean Absolute Error (MAE), achieving a 19.18% improvement over the Naive baseline — a benchmark model that simply uses the last available preliminary imbalance price (lagged by 60 minutes) as the forecast for the current interval. While XGBoost provided the most accurate typical predictions, RF recorded the lowest Root Mean Square Error (RMSE), with a 23.80% improvement compared to the baseline.
However, a closer look at the absolute values of these metrics reveals the extreme difficulty of the task. Even with the improvements over the baseline, the results are not entirely satisfactory: the best R^2 score (0.39 by RF) indicates that a large portion of the price variance remains unexplained. Furthermore, the sMAPE values, exceeding 43% even for the best model, underline a significant relative divergence between forecasts and actual prices, confirming that the high stochastic nature and volatility of the imbalance market severely limit the reliability of point forecasts.
Despite these statistical limitations, the economic assessment revealed a crucial insight. All tested models, including the Naive baseline, consistently predicted the sign of the imbalance, with directional accuracy ranging from 73.34% to 76.81%. This ability to identify the direction of the market explains why the models can be economically viable despite the difficulty in pinpointing exact price values. Within this context, RF generated the highest profit in the simulated trading scenario (+24.28% over the baseline). This success is primarily due to its superior performance in predicting the specific market scenarios that offer the highest profit potential, demonstrating that for a market operator, the strategic value of a forecast depends more on its performance during high-stakes intervals than on its point accuracy during stable conditions.
It can therefore be observed that even with simpler models, market participants can leverage these price signals to their economic advantage. However, it is important to emphasize that the practice of intentional imbalance is not permitted by market regulations. Consequently, these results should be interpreted as a measure of the models' predictive value rather than a suggested commercial strategy.

Despite the framework established in this study, several limitations emerge, providing directions for future research. First, the analysis could be extended by testing more models, specifically designed to handle price spikes which remain the most challenging events to predict. Furthermore, the optimization process of these models could be refined by shifting the focus from purely statistical loss functions to strategies that directly incorporate economic objectives into the training phase. From a structural perspective, future works should transition from point forecasting to probabilistic forecasting, providing prediction intervals or density forecasts would allow market participants to better quantify uncertainty. Additionally, expanding the forecasting horizon beyond the immediate next quarter-hour would provide operators with a broader temporal perspective for strategic planning.
Finally, it is essential to consider the evolving nature of the Italian power system. The current imbalance regulations are relatively recent, and as market participants adapt their bidding behaviors, the dynamics of price formation may evolve. Moreover, the increasing penetration of intermittent renewable energy sources will likely introduce higher volatility and new dynamics. Future studies should therefore continuously re-evaluate these models to ensure their resilience against the ongoing structural transformation of the energy market.

7. Bibliography and citations

References

[1] Terna S.p.A. “Pubblicazioni statistiche”, 2025. Available: https://www.terna.it/it/sistema-elettrico/statistiche/pubblicazioni-statistiche.

[2] Ministero dell’Ambiente e della Sicurezza Energetica. “Piano Nazionale Integrato per l’Energia e il Clima (PNIEC)”, 2024.

[3] Autorità di Regolazione per Energia Reti e Ambiente (ARERA). “Testo Integrato del Dispacciamento Elettrico (TIDE)”, 2023.

[4] Rafał Weron. “Electricity price forecasting: A review of the state-of-the-art with a look into the future”. *International Journal of Forecasting*, 30(4):1030–1081, 2014.

[5] Jesus Lago, Fjo De Ridder, and Bart De Schutter. “Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms”. *Applied Energy*, 221:386–405, 2018.

[6] Jesus Lago, Grzegorz Marcjasz, Bart De Schutter, and Rafał Weron. “Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark”. *Applied Energy*, 293, 2021.

[7] Ciaran O’Connor, Mohamed Bahloul, Steven Prestwich, and Andrea Visentin. "A Review of Electricity Price Forecasting Models in the Day-Ahead, Intra-Day, and Balancing Markets". *Energies*, 18(12), 2025.

[8] Ciaran O’Connor, Joseph Collins, Steven Prestwich, and Andrea Visentin. “Electricity Price Forecasting in the Irish Balancing Market”. *Energy Strategy Reviews*, 54, 2024.

[9] Ali Dinler. “A Review of Balancing Price Forecasting in the Context of Renewable-Rich Power Systems, Highlighting Profit-Aware and Spike-Resilient Approaches”. *Energies*, 18(24), 2025.

[10] Alexandre Lucas, Konstantinos Pegios, Evangelos Kotsakis, and Dan Clarke. “Price Forecasting for the Balancing Energy Market Using Machine-Learning Regression”. *Energies*, 13(20), 2020.

[11] Sinan Deng, John Inekwe, Vladimir Smirnov, Andrew Wait, and Chao Wang. “Seasonality in deep learning forecasts of electricity imbalance prices”. *Energy Economics*, 137, 2024.

[12] Michał Narajewski. “Probabilistic Forecasting of German Electricity Imbalance Prices”. *Energies*, 15(14), 2022.

[13] Jonathan Dumas, Ioannis Boukas, Miguel Manuel de Villena, Sébastien Mathieu, and Bertrand Cornélusse. " Probabilistic Forecasting of Imbalance Prices in the Belgian Context". In *2019 16th International Conference on the European Energy Market (EEM)*, 2019.

[14] Daniele Carnevale, Mattia Cavaiola, and Andrea Mazzino. "A novel AI-assisted forecasting strategy reveals the energy imbalance sign for the day-ahead electricity market". *Energy Reports*, 11, 2024.

[15] Francesco Lisi and Enrico Edoli. "Analyzing and Forecasting Zonal Imbalance Signs in the Italian Electricity Market". *The Energy Journal*, 39(5), 2018.

[16] Valerio D’Elia. “Analysis and forecasting of the Italian electricity market imbalance”. Master’s thesis, Politecnico di Milano, 2024.

[17] GME - Gestore dei Mercati Energetici S.p.A. “Vademecum alla Borsa Elettrica”, 2026. Available: https://www.mercatoelettrico.org/it-it/Home/MediaGME/Library/GMEVolumes.

[18] Terna S.p.A. “Italian Grid Code”, 2026. Available: https://www.terna.it/en/electric-system/grid-codes/italian-grid-code.

[19] Terna S.p.A. “Fees”, 2026. Available: https://dati.terna.it/en/fees.

[20] GME - Gestore dei Mercati Energetici S.p.A. “Esiti”, 2026. Available: https://gme.mercatoelettrico.org/it-it/Home/Esiti/Elettricita.

[21] ENTSO-E - European Network of Transmission System Operators for Electricity. “Transparency Platform”, 2026. Available: `https://transparency.entsoe.eu/`.

[22] Terna S.p.A. “Download Center”, 2026. Available: `https://dati.terna.it/en/download-center`.

[23] Bartosz Uniejewski, Jakub Nowotarski, and Rafał Weron. "Automated Variable Selection and Shrinkage for Day-Ahead Electricity Price Forecasting". *Energies*, 9(8), 2016. doi: 10.3390/en9080621.

[24] Leo Breiman. "Random Forests". *Machine Learning*, 45(1), 2001.

[25] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

[26] Mariette Awad and Rahul Khanna. *"Support Vector Regression"*. Apress, 2015.

[27] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". *Neural Computation*, 9(8), 1997.

[28] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. "Algorithms for Hyper-Parameter Optimization". In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24, 2011.

[29] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. "Optuna: A Next-generation Hyperparameter Optimization Framework". In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2019.

[30] Scott M. Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

A. Appendix A

Feature	MAE [€/MWh]	RMSE [€/MWh]	sMAPE [%]	R^2
Naive Baseline ($P^{sb}_{mz,prel}(t-60)$)	54.16	86.50	52.12	-0.05
$P^{sb}_{mz,prel}$, P^{MGP}_{z}	51.21	82.71	50.39	0.25
$P^{sb}_{mz,prel}$, P^{mean}_{MI-A}	51.19	73.01	50.31	0.25
$P^{sb}_{mz,prel}$, P^{last}_{XBID}	50.70	72.79	50.05	0.25
$P^{sb}_{mz,prel}$, $S_{mz,prel}$	51.60	74.98	51.48	0.21
$P^{sb}_{mz,prel}$, Err^{sw}_{gen}	53.07	75.57	52.15	0.19
$P^{sb}_{mz,prel}$, L^{f}_{tot}	52.49	75.20	51.97	0.20
$P^{sb}_{mz,prel}$, G^{f}_{tot}	53.14	75.88	52.97	0.19
$P^{sb}_{mz,prel}$	52.73	75.58	51.88	0.19

Table 25: Forecasting performance of the LEAR model with the imbalance preliminary price and single exogenous variables using all available lags up to $t - 120$ min.

Feature	MAE [€/MWh]	RMSE [€/MWh]	sMAPE [%]	R^2
Naive Baseline ($P^{sb}_{mz,prel}(t-60)$)	54.16	86.50	52.12	-0.05
$P^{sb}_{mz,prel}$, P^{MGP}_{z}	51.52	73.35	50.85	0.24
$P^{sb}_{mz,prel}$, P^{mean}_{MI-A}	51.85	73.99	50.95	0.23
$P^{sb}_{mz,prel}$, P^{last}_{XBID}	50.89	72.93	50.33	0.25
$P^{sb}_{mz,prel}$, $S_{mz,prel}$	51.15	75.57	51.21	0.21
$P^{sb}_{mz,prel}$, Err^{sw}_{gen}	52.84	75.44	52.24	0.20
$P^{sb}_{mz,prel}$, L^{f}_{tot}	52.91	75.23	52.38	0.20
$P^{sb}_{mz,prel}$, G^{f}_{tot}	55.13	75.90	52.69	0.19
$P^{sb}_{mz,prel}$	52.83	75.39	52.08	0.19

Table 26: Forecasting performance of the LEAR model with the imbalance preliminary price and single exogenous variables using all available lags up to Day-1.

Abstract in lingua italiana

La crescente integrazione delle fonti rinnovabili non programmabili ha reso la gestione del bilanciamento della rete elettrica sempre più complessa. Nel contesto italiano è avvenuto un importante cambiamento per quanto riguarda la gestione e la valorizzazione economica delle risorse di bilanciamento della rete, creando nuove opportunità per gli operatori di mercato di sfruttare la propria flessibilità. In questo nuovo scenario, guidato dall'introduzione della riforma TIDE (Testo Integrato del Dispacciamento Elettrico), disporre di una previsione accurata e in tempo reale del prezzo di sbilanciamento diventa quindi rilevante per ottimizzare le strategie di mercato. Questo lavoro si pone l'obiettivo di sviluppare e confrontare diversi modelli per la previsione vicina al tempo reale dei prezzi di sbilanciamento nel mercato elettrico italiano. A tal fine, sono stati implementati e confrontati cinque algoritmi: Lasso Estimated AutoRegressive (LEAR), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Support Vector Regression (SVR) e Long Short-Term Memory (LSTM). I risultati sperimentali dimostrano che XGBoost ottiene la massima accuratezza con un MAE di 43,77 €/MWh e uno sMAPE del 43,57%. Al contempo, Random Forest si dimostra il più efficace nella gestione della volatilità dei prezzi, ottenendo il miglior RMSE pari a 65,91 €/MWh e un R^2 di 0,39. Infine, tutti i modelli implementati permetterebbero agli operatori di mercato di aumentare significativamente la redditività effettuando sbilanciamenti intenzionali basati sulle previsioni di prezzo, una pratica che rimane espressamente vietata dalle normative del mercato italiano.

Parole chiave: previsione prezzi di sbilanciamento, mercato elettrico italiano, previsione a breve termine, machine learning