The DELTA project has received funding from the EU's Horizon 2020 research and innovation programme under grant agreement No 773960

| | |
|---|---|
| Project Acronym: | **DELTA** |
| Project Full Title: | **Future tamper-proof Demand rEsponse framework through seLf-configured, self-opTimized and collAborative virtual distributed energy nodes** |
| Grant Agreement: | **773960** |
| Project Duration: | **36 months (01/05/2018 – 30/04/2021)** |

# DELIVERABLE D4.3
# DR, Flexibility Forecasting and Energy Market Simulation

| | |
|---|---|
| Work Package | **WP4 – Adaptive Aggregator DELTA Supervisory Engine** |
| Task | **T4.3 – DR Forecasting through DELTA Simulation (M10-28)** |
| Document Status: | **Final** |
| File Name: | **[DELTA]_D4.3_Final** |
| Due Date: | **31.08.2020** |
| Submission Date: | **01.09.2020** |
| Lead Beneficiary: | **JRC** |

| **Dissemination Level** | |
|---|---|
| Public | X |
| Confidential, only for members of the Consortium (including the Commission Services) | |

# Authors List

| Leading Authors | | | |
|---|---|---|---|
| **First Name** | **Last Name** | **Beneficiary** | **Contact e-mail** |
| Alexandre | Lucas | JRC | Alexandre.lucas@ext.ec.europa.eu |
| Evangelos | Kotsakis | JRC | Evangelos.kotsakis@ec.europa.eu |

| Co-Author(s) | | | | |
|---|---|---|---|---|
| **#** | **First Name** | **Last Name** | **Beneficiary** | **Contact e-mail** |
| 1 | Konstantinos | Pegios | Kiwi | kpegios@kiwipowered.com |
| 2 | Apostolos | Tsolakis | CERTH | tsolakis@iti.gr |
| 3 | Ioannis | Koskinas | CERTH | jkosk@iti.gr |
| 4 | Venizelos | Venizelou | UCY | venizelou.venizelos@ucy.ac.cy |
| 5 | George | Karagiannopoulos | HIT | g.karagiannopoulos@hit-innovations.com |

# Reviewers List

| Reviewers | | | |
|---|---|---|---|
| **First Name** | **Last Name** | **Beneficiary** | **Contact e-mail** |
| Phivos | Therapontos (PT) | EAC | PTherapo@Eac.com.cy |
| Ioannis | Papageogriou | EAC | IPapageo@Eac.com.cy |
| Carolina | Fernandes | Kiwi | cfernandes@kiwipowered.com |

# Legal Disclaimer

The DELTA has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 773960. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the Innovation and Networks Executive Agency (INEA) or the European Commission (EC). INEA or the EC are not responsible for any use that may be made of the information contained therein.

# Copyright

# Executive Summary

The role of electricity is expected to increase in the coming decade. The EU long-term vision to 2050 contains in all scenarios high end-use electrification which would see the share of electricity in final energy consumption grow to 53% by 2050, from 20% in 2018 of electricity in final energy consumption share. The need for a more adequate system has enhanced the importance of a flexible system both on the generation and demand side. Flexibility of consumption in some member states has been explored mostly in the Industry sector, taking advantage of their large power assets and predictable load profiles. There is however, a huge potential if the service and residential sector is explored. The fact that services and the residential sectors could be explored, smaller assets would be targeted. This implies many challenges, such as requiring accurate load forecasts, interaction with the assets, coordination, settlements, service tracking and interoperability just to mention a few. The Delta architecture tackles these issues and provides a solution to facilitate many of the tasks being performed by the Aggregator as an actor. One of the tasks is to understand when (market, settlement period) and what (assets, power) to bid in order to maximise potential revenues.

The importance of price forecasting has gained attention over the last few years, with the growth of Aggregators and the general opening of the European electricity markets. Market participants manage a tradeoff between, bidding in a lower price market (day-ahead), but with typically higher volume, and a lower volume market but with potentially higher returns (Balance energy market). Companies try to forecast the limits of revenues or prices, in order to manage risk and opportunity, assigning their assets in an optimized way. It is thought that in general, electricity markets have quasi-deterministic principles, rather than being based on speculation, hence the desire to forecast the price based on variables that can describe the outcome of the market.

Many studies address this problem from a statistical approach or by performing multiple-variable regressions, but they very often focus only on the time series analysis. In 2019, the Loss of Load Probability (LOLP) was made available in the UK for the first time for the full year. Taking this opportunity, this report focusses on five LOLP variables (with different time-ahead estimations) and other quasi-deterministic variables, to explain the price behavior of a multi-variable regression model. These include base production, system load, wind and solar generation, seasonality, day-ahead price and imbalance volume contributions. Three machine-learning algorithms were applied to test for performance, Gradient Boosting, Random Forest and XGBoost. The latter has a higher performance and so implemented for real time forecast. The model returns a mean absolute error (MAE) of 7.89 £/MWh, a coefficient of determination (R2 score) of 76.8% and a mean squared error (MSE) of 124.74. The variables that contribute the most to the model are the Net Imbalance Volume, the LOLP (aggregated), the month and the De-rated margins (aggregated) with 28.6% with 27.5%, 14.0%, and 8.9% of weight on feature importance respectively.

The goal of the report is to explain how the model can used as a support tool in the bidding strategy of an aggregator combining both modelling and statistical analysis to determine the higher price periods or the peaks of the day where the flexibility of a portfolio could be allocated.

# Table of Contents

## List of Figures

## List of Tables

# List of Acronyms and Abbreviations

| Term | Description |
|---|---|
| ARIMA | Autoregressive Integrated Moving Average |
| CEP | Clean Energy Package |
| BRP | Balance Responsible Party |
| DR | Demand Response |
| DSO | Distribution System Operator |
| TSO | Transmission System Operator |
| UK | United Kingdom |
| BSC | Balancing and Settlement Code |
| DA | Day Ahead |
| SP | Settlement Period |
| GB | Gradient Boosting |
| RF | Random Forest |
| XGBoost | Extreme Gradient Boosting |
| LOLP | Loss of Load Probability |
| MS | Member State |
| NETSO | National Electricity Transmission System Operator |
| SO | System Operator |
| NIV | Net Imbalance Volume |
| DRM | De-Rated Margin |
| ITSD | Initial Transmission System Demand |
| CR | Capacity Requirements |

# 1. Introduction

## 1.1 Scope and objectives of the deliverable

This deliverable is associated with Task 4.3 of the DELTA project and provides the modelling results of the price forecast techniques applied to the Balance Energy Market in the UK. Both Cyprus and UK pilot sites are the focus of the task, however since the Cyprus context cannot yet provide market data for the modelling activity due to its immature state, the study was developed only based on the UK data and context.

This report focusses on five LOLP variables (with different time-ahead estimations) and other quasi-deterministic variables, to explain the price behavior of a multi-variable regression model. These include base production, system load, wind and solar generation, seasonality, day-ahead price and imbalance volume contributions. Three algorithms were explored:

1. Gradient Boosting
2. Random Forest;
3. Extreme Gradient Boost;

The latter presents a higher overall performance and so implemented for real time forecast. The model returns a mean absolute error (MAE) of 7.89 £/MWh, a coefficient of determination (R2 score) of 76.8% and a mean squared error (MSE) of 124.74.

## 1.2 Structure of the deliverable

The work presented in this deliverable is structured as follows.

- Chapter 2 presents the context and state of the art in forecasting markets
- Chapter 3 introduces the methodological steps during the study, the details about the market in the study, which is focused on the UK reality, explaining some basic concepts about the Energy Balance Market and explains the LOLP variables. It focuses on the Net Imbalance variable mitigation technique, using queries to the historical dataset and the decomposition by quantiles.
- Chapter 4 introduces the decisions regarding data treatment, hyper parameters optimization, libraries used and the real data acquisition. It shows the results of the three algorithms used considering different metrics and variable impact. It also promotes a discussion on the use of the model and its shortcomings as it presents the results for a test day (based on real data acquisition) showing how a decision could be made to choose the best settlement period to allocate flexibility
- Chapter 5 concludes the report, stating the metrics and feature importance of the main variables

## 1.3 Relation to Other Tasks and Deliverables

The main achievement of this report is the delivery of modular service related to DR-related pricing forecasting that will be part of the DELTA Decision Support System (T4.4). The service will specifically be used by the Aggregator’s DSS towards identifying dynamically the optimal decision for participating within the imbalance market with the available flexibility using demand-response schemes. The inputs for this task were from the market itself, specifically from the ELEXON site. Elexon is responsible for the Balance Market and the Nord Pool site which runs the leading power market in Europe, offering the day-ahead and intraday markets.

## 2. Legal and Overall Framework

Over the past five years, the EU has made continuous progress in completing the internal electricity market, with most of the EU's borders and prices under market coupling, while increasing the interconnectivity with the periphery, including the Baltics, Turkey and North Africa. The EU has upgraded its market design to prepare it for the power system transformation it is experiencing. The institutional structures are in place to ensure harmonized network codes and rules for cross-border trading, along day-ahead, intraday and balancing market time frames, as well as enhanced system operation and security of supply rules. The Clean Energy Package (CEP) [1] was adopted in 2019 and has codified these new rules for the wholesale market architecture and as so the full implementation of the CEP has only just started. According to the IEA [2] it will bring along greater flexibility, including at the retail market level, allowing active consumer participation, greater distributed energy deployment and demand response. The new electricity market design in the EU is an inspiration for many large regional electricity markets around the world, and a source of many lessons learned and best practices in wholesale market integration. In no other region of the world do cross border electricity grids contribute so significantly to system integration of variable renewable generation as in the EU. With energy storage, Demand Response (DR), energy communities and Aggregators being now a reality there are new opportunities for market participation and to generate profits

The role of electricity is expected to increase in the coming decade. The EU long-term vision to 2050 [3], considers in all scenarios high end-use electrification which would see the share of electricity in final energy consumption grow to 53% by 2050, from 20% in 2018 of electricity in final energy consumption share. Figure 1 shows the trend of final electricity consumption by sector in the EU.

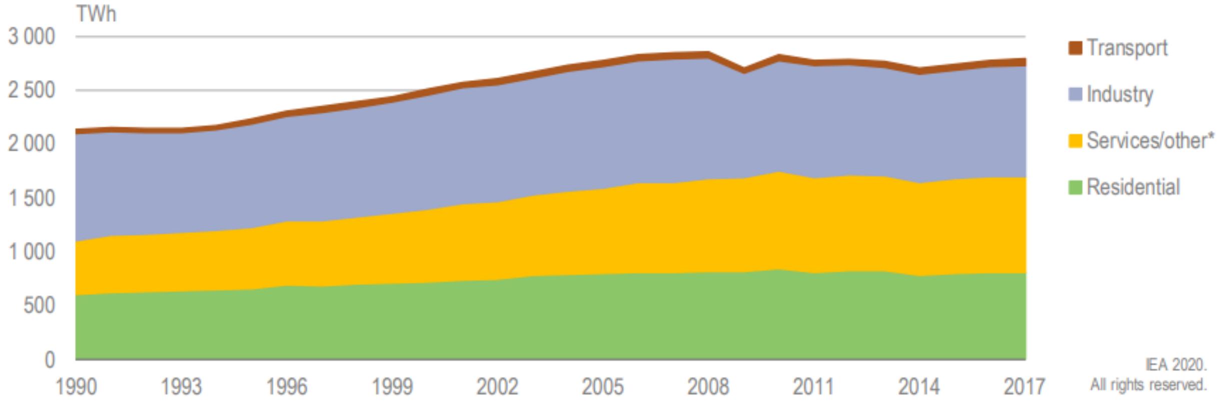


**Figure 1. Total Electricity consumption by sector in the EU, 1990-2017** [2]

Generation adequacy moved towards a resource adequacy concept, among others to include demand side management. Flexibility of consumption in some member states has been explored mostly in the Industry sector, taking advantage of their large power assets and predictable load profiles. There is however, a huge potential if the service and residential sectors are explored. Figure 2 shows what space demand response and smart charging could occupy, when providing services to the grid, especially to grid constraint management (procured by DSOs), but also to other users such as Aggregators, to satisfy or complement their portfolios, or BRPs. The fact that services and the residential sectors could be explored, smaller assets would be targeted. This implies many challenges, such as requiring load forecasts, interacting with the assets, coordination, settlements, service tracking and interoperability just to mention a few. The Delta architecture tackles these issues and provides a solution to facilitate many of the tasks being performed by the Aggregator actor. One of the tasks is to understand when (market, settlement period) and what (assets, power) to bid in order to maximise the revenues.

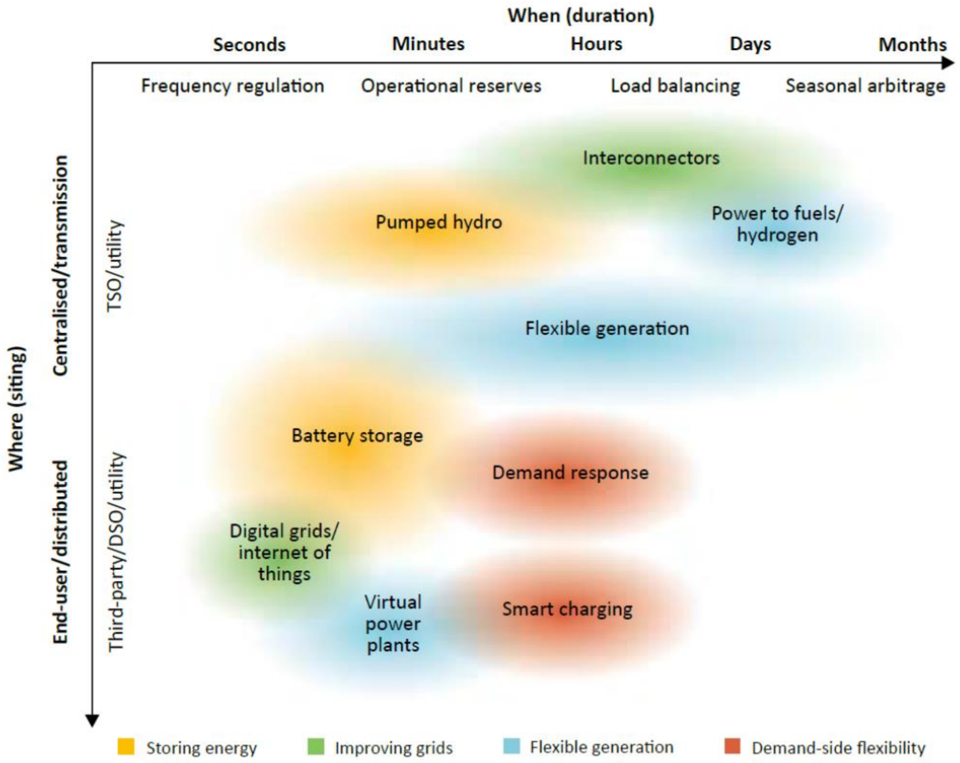


**Figure 2. Flexibility needs increase across a larger portfolio of services and time frames** [2]

Market forecasting in order to optimize resources and revenues are now in the forefront of research and discussions for power generators, power purchasing parties/consumers and regulators alike. Besides the dynamic and highly complex traded value of electricity as a commodity, which is subject to multiple intrinsic and extrinsic variables, trying to predict the market with a high degree of confidence has many benefits for all market participants. Such benefits arise from setting up their respective risk-adjusted bidding strategies and ensuring cost-optimal generation activities, to formulating evidence and insights on the economic development of a country's power sector, as well as ensuring the reliable security and safe operation of the whole electricity system [4], [5]. In order to maximize profits, market participants employ forecasting tools with different time horizons based on the different markets they are operating in. These typically vary from half-hourly or hourly settlement periods (real-time, balancing markets, imbalance exposure) to months ahead in the wholesale markets (intraday, day-ahead, spot, derivative markets) similar to those of a commodity.

Market participants are faced with a number of risks, including the constant requirement of maintaining the supply-demand equilibrium, the short-term inelastic demand, and the generation and load side uncertainties in the system. Moreover other quantifiable and non-quantifiable factors also introduce risk, such as fuel prices, cost of unit operation, the markets design and requirements and also the impact of weather conditions, just to mention a few. Other uncertainties regarding the complex process of forecasting the electricity price include extreme volatility, high frequency and price spiking behavior, non-constant mean and variance as well as multiple seasonality [6], [7].

In this context a plethora of studies has been developed over the years, in order to address the compound implications of electricity price forecasting. Aggarwal et al. (2009) [6] published a comprehensive first of its kind market forecasting review of 47 articles. It focused on the quantitative methodological approaches undertaken, and distinguished them based on the type of the models and their architectures, the input and output variables used, the prediction horizon of the forecasts, as well as the preprocessing and the exploratory analysis of the data and the results. According to the authors, the employed techniques were found to be similar to load forecasting models and were broken down into a) game theory models including stochastic and parsimonious models, b) time-series models, which incorporated artificial intelligence and neural-network based algorithms, and finally c) simulation techniques which accounted for regression or other causal models. The input factors which could have an impact on the electricity prices were grouped into: a) market characteristics (historical generation, supply, load etc.), b) nonstrategic uncertainties such as forecast load and reserves or weather parameters, c) other stochastic uncertainties such us generation outages and transmission congestions/contingencies, d) behavior indices which referred to historical price data, demand elasticity and market participants' bidding strategies, and e) temporal effect such as settlement period, day, month, public holidays, seasons etc.

The study's in-depth analysis, found that the time of the day variable and the more complex to model bidding strategies factor, were the most significant, suggesting a mathematical equation, which would also incorporate an additive residual term. This term intends to reflect the load and supply deviation from normally and randomly correlated short-term effects in the market. Despite their very systematic approach, they concluded that there was no systematic evidence of one model outperforming another on a consistent basis mainly due to the illiquidity and general paucity of historical electricity market data. The study also reported that some methods, i.e. multivariate dynamic regression, transfer loss models and nonlinear neural-network models, performed qualitatively better compared to univariate autoregressive integrated moving average (ARIMA) ones. However, the latter in combination with fuzzy logic or a wavelet transformation approach could hold promise in future developments. An even more comprehensive review of hundreds of relevant articles, proceedings and journals in the literature by Weron (2014) [8], suggested the categorization of models into similar categories shown in Figure 3. The main difference is the addition of the structural or fundamental models, which derive from the modelling of significant economic and physical factors in the power systems as well as hybrid solutions of the following techniques in the sub-branches.

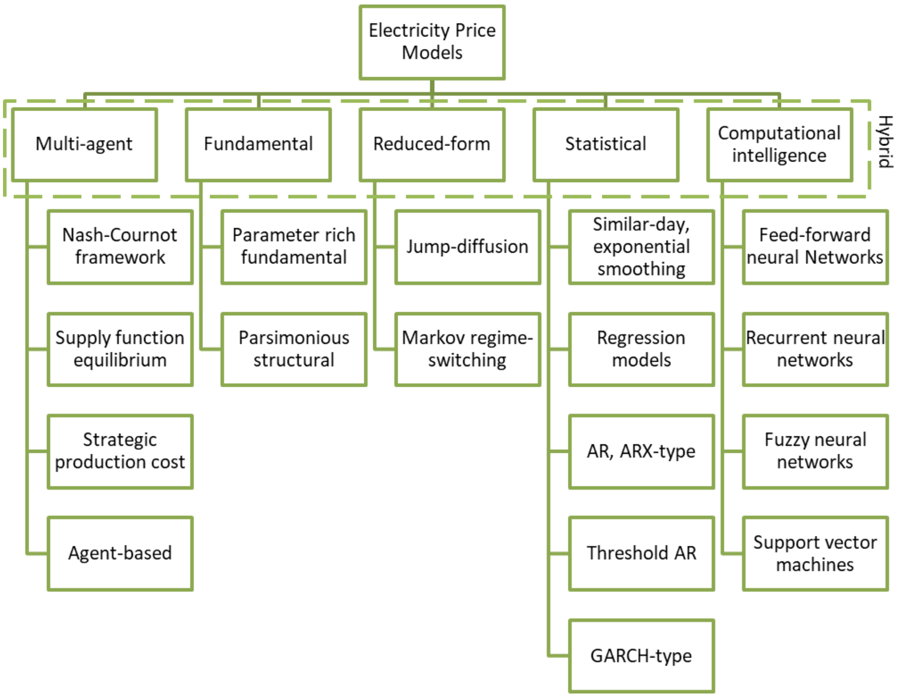


**Figure 3. Taxonomy of electricity price forecasting approaches based on** [8]

The author highlights in great detail the flexibility of multi-agent models [9]–[11][12][13], [14][15] with regards to the analysis and incorporation of the multi-dimensional strategic behaviors of the market participants as variables and agents. At the same time, this is one of the main caveats of this family of models since the underlying assumptions used in the model simulations introduce a lot of risk and uncertainty (i.e. a power generator can be either a buyer or a seller depending on his position or strategy). Evidence in the literature, also shows that another constraint of agent-based models is the prediction accuracy of the electricity price as an output variable, since the outcomes of such models have more qualitative implications (i.e. whether prices will be above marginal costs or not) rather than quantitative [8]. With regards to fundamental models, these are considered to be more suitable for medium-term forecasts and not so much for short-term predictions, due to data availability and resolution. Due to the nature of the fundamental data used on plant and transmission capacities and costs, they tend to overlook the hourly or half-hourly resolution, of the data needed in the case of short-term price forecasting, hence, they seem to be a better fit for describing market fundamentals. On a similar note, another challenge they face is their sensitivity to violations on significant assumptions made on the economical and physical relationships of the power entities of the market, therefore their optimization and calibration tend to be rather complex when incorporating stochastic fluctuations of fundamental factors. Similarly, purely reduced-form models, such as mean-reverting jump-diffusions and Markov regime-switching models [16]–[22] are expected to perform better on a daily horizon level and less well on an hourly or half-hourly short-term basis as evidence proves their poor performance [23], [24].

However, a hybrid model combining both a Markov-regime switching technique and vector auto regressions in a more macroeconomic context, as suggested by some authors [25] might turn out to be more effective. When reviewing the statistical methods employed in the literature, Weron (2014) [8] refers to the importance of the quality and efficiency of the methods used, highlighting the ability to incorporate filtered, and well-tested fundamental historical data (i.e. during normal days without unusual price movements or spikes). Many discussions have been promoted around the ability of statistical models to capture price volatility and sudden spikes and whether data should be filtered with a more comprehensive exploratory analysis of outlier detection prior to the application and comparison of the different methods. The majority of the literature however, tends to agree that they perform rather poorly to this extent making clear the substantial impact that extreme observations might have on the outcomes of a study and that an adequate stochastic model is essentially more suitable for detecting those price spikes. Many different methods have been suggested in the literature for addressing the issue of capturing those sudden price movements. These include variable price thresholds, regime-switching classification approaches, wavelet filtering and transformation techniques, recursive filters, and fixed price change thresholds which seemed to be the worst-performing method discussed, due to its inability to capture the long-term seasonal behavior of the market prices [16], [17], [24], [26]–[30].

Additional literature suggests the replacement of those spiky instances with various methods. These include finding instances in the historical data with similar patterns, taking the average/median of periods with matching temporal attributes such as the hour, the day, the month; replace spiking values with a chosen threshold; or simply deriving the mean of neighbouring settlement periods and essentially prices [18], [24], [31], [32]. With regards to artificial intelligence-based, non-parameter/linear techniques employed in the literature, there is a vast pool of them with both strengths and weaknesses. On one hand, they are found to be very flexible, powerful tools able to capture non-linear parameters, and potentially evolution and fuzziness making them more capable of adapting to complex dynamic systems and constraints. On the other hand, there is no systematic evidence they clearly outperform the previous families of models [8]. Their rich and complex architecture makes them hard to compare thoroughly, and the calibration of each one of them is so unique that it makes it very challenging to establish a common basis for comparison.

However, the combination of multilayer perceptron architectures into hybrid models with multiple types of neural networks such as long short-term memory, convolutional neural networks, or recurrent neural networks, or other types of algorithms such as clustering, ARIMA, trigonometric seasonal box-cox transformation, residuals trend and seasonal components approaches, show potential for useful and robust forecasting tools [4], [5], [33]–[35] primarily for day-ahead and spot markets. Less attention has been paid to the forecasting of real-time, balancing prices employing hybrid approaches again such as ARIMA and exponential smoothing approaches, and other combinations of multi-layer artificial neural networks perceptron with interfering deterministic and probabilistic techniques [36]–[39].

Across all of the literature, a key point for predicting electricity prices is the selection of the dependent variables, the predictors. Apart from seasonal attributes, which are easily derived from the temporal nature of the output variable (price), there is strong evidence for the fundamental factors that drive the price. These include system loads (demand, consumption and generation), climatic and weather variables, fuel costs, reserve margin variables such us surplus or deficit of generation, and most recently the data around planned maintenance or forced outages of plant trips [8], [19], [40], [41]. The aforementioned data however, is not always available or found to be significant, as shown in an indicative report for the United Kingdom (UK) market by Maciejowska [42], who used structural vector autoregressive models, in order to capture speculative electricity shocks. The study highlighted that expected major drivers such as wind generation and supply and demand, were not the ones explaining the extreme volatility of prices in earlier years. Even though the majority of the literature selects a combination of the main fundamental drivers of prices [43], there is not an optimal, fit-for-all, set of variables that can be established for all power markets. This is because the model category described in the previous paragraphs, the calibration and availability of the data as well as the objective of the research questions, need to be further explored in order to extract the most effective, minimum set of input variables that will not lead to under or over-fitting issues [8].

The literature review indicates that, price forecasting has gained a renewed attention, given the growing trend of aggregation activities and the market opening to demand response service providers. The main motivation remains the maximisation of revenues, taking advantage of the day-ahead (DA) and the Balance markets' most favourable moments. Aggregators manage portfolios of flexible assets, which given their finite available power, need to be assigned to the most advantageous settlement period (SP) and market, hence the need to predict the price.

This study presents the development of a multi-regression model, testing three machine learning algorithms, Gradient Boosting (GB), Random Forest (RF) and Extreme Gradient Boosting (XGBoost), presenting a combined approach of several categories according to Aggarwal et al. 2009 [6] classification. Market historical data is used for generation, supply, load, temporal effect such as settlement period, day, month, holiday, season and nonstrategic uncertainties, such as forecast load and probability of reserves plus generation to meet demand. For this latter variable, the Loss of Load Probability (LOLP) is used with different time horizons to capture this uncertainty. The model is a tool for short term forecasting, which can be used from 12 h ahead up to 1h before the gate closure. With resource adequacy methodologies being implemented and several metrics becoming available for the security of supply, value of loss load, loss of load expected and LOLP, new analysis are possible. In order to conduct the analysis, the ELEXON Balancing Energy Market in the UK is considered. To the best of our knowledge such approach has not been taken before, since the first full year with LOLP data included in this model, has only just become available for the year 2019.

# 3. Background and Data analysis context

## 3.1. Balancing Market functioning

In Europe, electricity markets in different zones or member states (MS) may still differ in their rules, terms and operation, but may also be typically found as sequence of year-ahead, month-ahead, day-ahead, intra-day markets and at the very end the energy balancing market (also called Imbalance Market). However, the design of the balancing market is more sophisticated, as it lies at the junction of financial transactions (the power market) and physical exchanges (the power system). It is the last opportunity for all parties to state a position (load/generation decrease or increase needs/availability), for each settlement period. After this stage only the Balancing Mechanism is left to balance the grid close to real time. In order to focus on a specific framework, in this study the UK Energy Balancing Market, which is managed by ELEXON [44], is address. ELEXON has the function to administer the Balancing and Settlement Code (BSC) and provide and procure the services needed to implement it. Essentially, ELEXON compares how much electricity generators and suppliers say they will produce or consume, with actual volumes and enables the imbalance settlement by managing the Balancing Market. ELEXON serves around 470 market participants and settled around 44 TWh in balancing actions and partys' imbalance volumes in 2018/2019 [45]. The balancing of the Transmission System is under the responsibility of National Electricity Transmission System Operator (NETSO), which acts as the System Operator (SO) and takes balancing actions. A balancing action is an instruction to a party, in accordance with agreed rules, to either increase or decrease generation, or increase or decrease demand.

All parties must submit details of their contracts to the BSC Systems. After the end of the settlement period, the BSC Systems compare a party's contracted (traded) volume, with its metered volume in order to determine its imbalance. If a party is in imbalance of its contracted volume then it will be subject to imbalance charges. After the energy balance and system balance actions (for system management reasons) are taken, adjustments for transmission losses are balanced, a volume-weighted average is taken to calculate the energy imbalance price or charge. Parties are first billed for imbalance charges approximately one month after the Settlement Day for which the charges were incurred. The BSC Systems carry out subsequent Reconciliation Runs over the next 13 months, which update the imbalance charges by replacing any estimated data with actual metered data. There are several reasons for imbalances, for example suppliers may not always accurately predict demand, or generators may not always be able to tightly control their generation as is the case of intermittent generation. In addition, problems can arise with transmission lines. The BSC does not require parties to meet their contracts and the market trades in half hour Settlement Periods, but the Transmission System must balance at every instant. After the Balance Market closes the Balancing Mechanism starts. The minimum capacity position is 1 MW. The capacity position is stated in power and minutes and a 1 minute is given for ramping up and down the asset.

## 3.2. Characterization and Predictors for the UK market

For the current analysis, the study focusses on the time window between the 1st of January 2019 and the 31st of December 2019, capturing 17520 observations, corresponding to 30 minutes time intervals. All the variables collected are also provided with 30 minutes time intervals, except for the day-ahead price given every hour, and so it was duplicated in each SP. The model uses 19 variables: LOLP with five ahead-of-time values, five corresponding De-rated Margins (DRM), Settlement Periods (SP), Production, Wind and Solar Generation, NIV or Net Imbalance Volume (NetImbVol), Weekdays, Months, Day-ahead Price (Price DA) and the Initial Transmission System Demand (Itsd), or simply system load.

The innovative part of the study is the focus on the LOLP variable. A LOLP value is a measure of scarcity in available surplus generation capacity that the NETSO will calculate for each Settlement Period. That is, for a given level of Capacity Requirement (CR) (measured in MW) on the Transmission

System, the associated LOLP indicates the probability that there will be insufficient Total Generation Capacity (Z) (measured in MW) to meet the CR. There are two types of LOLP values - indicative and final. For a given settlement period, the NETSO produces indicative LOLP values from the available data at defined lead times (at midday the day before and at 8, 4 and 2 hours ahead of gate closure for the SP). BSC parties use Indicative LOLP values as an indication of the level of scarcity anticipated ahead of gate closure for a SP. For the same SP, the NETSO produces final LOLP values from data available to it, at gate closure. The final LOLP is the best indication of expected scarcity during the SP. The Commission Interim Report of the Sector Inquiry on Capacity Mechanisms [46] refers to a calculation of a LOLP, as a more sophisticated method to measure generation adequacy.

Pursuant to the said document [46], LOLP quantifies the probability of a given level of unmet demand over a certain period of time. The Dynamic LOLP Function Method, is the one used by the NETSO to produce Indicative LOLP values from 1 May 2018, and final LOLP values from 1 November 2018. For a given settlement period, the dynamic model uses a direct relationship between the available generation (Z) and the Capacity Requirement (CR) as shown in Equation 1 [47]. The term Zj is the Combined Generation Forecast developed in in Equation 2, where Xj is the Conventional Generation Forecast shown in Equation 3.

$$LoLPj = P(Zj - CRj < 0), \qquad (1)$$

$$Zj = Xj + Wj, \qquad (2)$$

$$Xj = \sum(GCAPij \times AVi), \qquad (3)$$

In Equation 3, the GCAPji variable is the Generation Capacity of a conventional generator and AVi is an Availability Factor. The variable Wj in Equation 2 is the Total Wind Generation Forecast and CR in Equation 1 is the Capacity Requirement.

A crucial variable for any forecasting model is the Net Imbalance Volume. It refers to the resulting volume of positions, which were negotiated in the market for each SP. This volume is different from the one assigned to each party. A party’s imbalance position is simply its metered volumes compared to its contracted volumes. The contracted volumes are adjusted for any accepted bids and offers or delivery of Balancing Services. Energy imbalance volume = Energy – (Balancing Services + contracts). This results in a positive or negative volume of imbalance. A negative imbalance volume means that a party has under-contracted and is therefore short of energy. A positive imbalance volume means that a party has over-contracted and is therefore long on energy. The BSC Systems calculate the imbalance volumes for all parties for every settlement period. The NetImbVol is normally one of the variables used in most models. However, it cannot be a direct input to the model, as it cannot be foreseen ahead of time with sufficient accuracy. Another variable used in the model is the Initial Transmission System Demand variable (given in MW), which is the system load and refers to an average energy in each of the 48 SP of a day. The dataset used in the study is a time series, which was decomposed so as to provide information on weekdays and months. The production is kept separated as base generation (Production), distinct from Wind and Solar generation, with all values provided in MW.

# 4. Methodology

A multi variable regression was performed using each of the described predictors in section 3.2. The dataset initially had 17520 observations. The mean value of the target variable (Price) in the dataset is £ 41.99 with a minimum and maximum of -88 and +375 £/MWh respectively. Due to its disproportional value, such observations would introduce high variance in the model if not removed. For this reason they were considered as outliers and removed by applying a >99.75% and < 0.25% quantile exclusion, resulting in 17428 observations in the dataset. A summary of the main variables is presented in Table 1.

**Table 1. Dataset summary of numeric and target variable considered**

| Parameter | NetImbVol | Production | Wind | Solar | Price | Itsd | LOLP_12h* |
|---|---|---|---|---|---|---|---|
| min | -1534.00 | 0.00 | 0.00 | 0.00 | -60.00 | 18209.00 | 0.00 |
| 25% | -226.00 | 17266.00 | 2475.00 | 0.00 | 27.00 | 25559.00 | 0.00 |
| Std. | 314.11 | 6738.22 | 2860.97 | 1933.91 | 20.94 | 6426.33 | 199.08 |
| mean | -40.77 | 22375.83 | 4948.92 | 1256.96 | 41.65 | 30538.80 | 4.37 |
| 50% | -28.00 | 21712.00 | 4574.00 | 13.00 | 40.00 | 29843.00 | 0.00 |
| 75% | 146.00 | 26820.50 | 7069.00 | 2050.00 | 55.00 | 34680.50 | 0.00 |
| max | 2017.00 | 44493.00 | 14090.00 | 9712.00 | 136.00 | 48697.00 | 19615.00 |

* LOLP with a scientific notation of $10^{-6}$.

The model is designed to read the real data for the next day and provide a forecast for each SP. For this to occur, the model reads the forecasted predictors directly from the ELEXON website. This was possible for all variables except the NIV of each SP, which due to the uncertainty of maintenance, shortages of different sorts, failures, and unscheduled interventions is not provided by ELEXON. A strategy to estimate the value of this variable was developed, identifying patterns in the historical dataset and performing a regression to the quantile decomposition. As it can be observed in the Figure 4 the mean, maximum and minimum have high variance during the year and so other variables need be to included in the query.

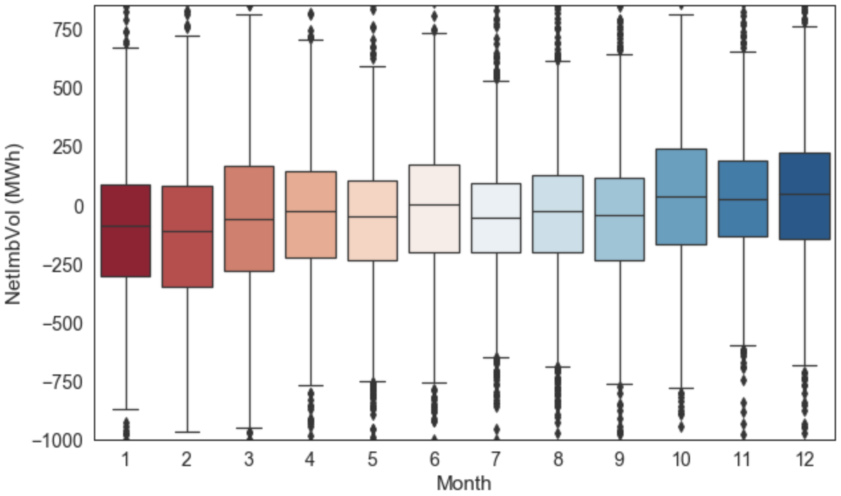
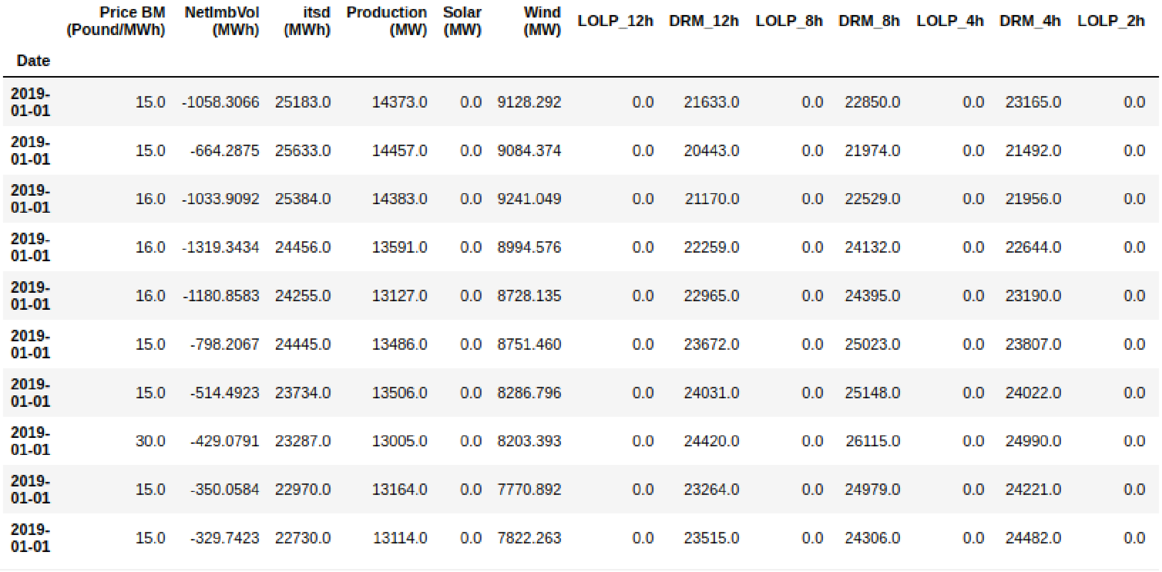


**Figure 4. Yearly variation boxplot of NIV by month**

## 4.1. ML Regression for Net Imbalance Volume estimation

NetImbVol is a crucial metric towards the forecasting of Imbalance Price as one of the highest correlated features. However, it is characterized by high volatility and the existence of many imponderables that affect its volume. As a result, forecasting this metric has many obstacles that need to be tackled and many features that need to be examined.

NetImbVol as a definition is the volume of the overall System energy imbalance for a specified 30 minutes Settlement Period. One of the possible approaches for the estimation of forecasted values for multiple steps ahead includes a recursive multi-step forecasting strategy. Specifically, the designed model forecasts the volume for the next settlement period and this information is utilized as input for further predictions. Our initial attempts to design a forecasting model included many features as input to the model: LOLP (loss of load probability), historical values of Net Imbalance Volume, historical Day ahead Market Prices, historical Intra Day Market Prices, weather conditions and generated features related with datetime like working/not working days, holidays, days of the week and months. The data that we had at our disposal concerns the year of 2019 with an example show in Figure 5.



| | Price BM (Pound/MWh) | NetImbVol (MWh) | itsd (MWh) | Production (MW) | Solar (MW) | Wind (MW) | LOLP_12h | DRM_12h | LOLP_8h | DRM_8h | LOLP_4h | DRM_4h | LOLP_2h |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Date | | | | | | | | | | | | | |
| 2019-01-01 | 15.0 | -1058.3066 | 25183.0 | 14373.0 | 0.0 | 9128.292 | 0.0 | 21633.0 | 0.0 | 22850.0 | 0.0 | 23165.0 | 0.0 |
| 2019-01-01 | 15.0 | -664.2875 | 25633.0 | 14457.0 | 0.0 | 9084.374 | 0.0 | 20443.0 | 0.0 | 21974.0 | 0.0 | 21492.0 | 0.0 |
| 2019-01-01 | 16.0 | -1033.9092 | 25384.0 | 14383.0 | 0.0 | 9241.049 | 0.0 | 21170.0 | 0.0 | 22529.0 | 0.0 | 21956.0 | 0.0 |
| 2019-01-01 | 16.0 | -1319.3434 | 24456.0 | 13591.0 | 0.0 | 8994.576 | 0.0 | 22259.0 | 0.0 | 24132.0 | 0.0 | 22644.0 | 0.0 |
| 2019-01-01 | 16.0 | -1180.8583 | 24255.0 | 13127.0 | 0.0 | 8728.135 | 0.0 | 22965.0 | 0.0 | 24395.0 | 0.0 | 23190.0 | 0.0 |
| 2019-01-01 | 15.0 | -798.2067 | 24445.0 | 13486.0 | 0.0 | 8751.460 | 0.0 | 23672.0 | 0.0 | 25023.0 | 0.0 | 23807.0 | 0.0 |
| 2019-01-01 | 15.0 | -514.4923 | 23734.0 | 13506.0 | 0.0 | 8286.796 | 0.0 | 24031.0 | 0.0 | 25148.0 | 0.0 | 24022.0 | 0.0 |
| 2019-01-01 | 30.0 | -429.0791 | 23287.0 | 13005.0 | 0.0 | 8203.393 | 0.0 | 24420.0 | 0.0 | 26115.0 | 0.0 | 24990.0 | 0.0 |
| 2019-01-01 | 15.0 | -350.0584 | 22970.0 | 13164.0 | 0.0 | 7770.892 | 0.0 | 23264.0 | 0.0 | 24979.0 | 0.0 | 24221.0 | 0.0 |
| 2019-01-01 | 15.0 | -329.7423 | 22730.0 | 13114.0 | 0.0 | 7822.263 | 0.0 | 23515.0 | 0.0 | 24306.0 | 0.0 | 24482.0 | 0.0 |

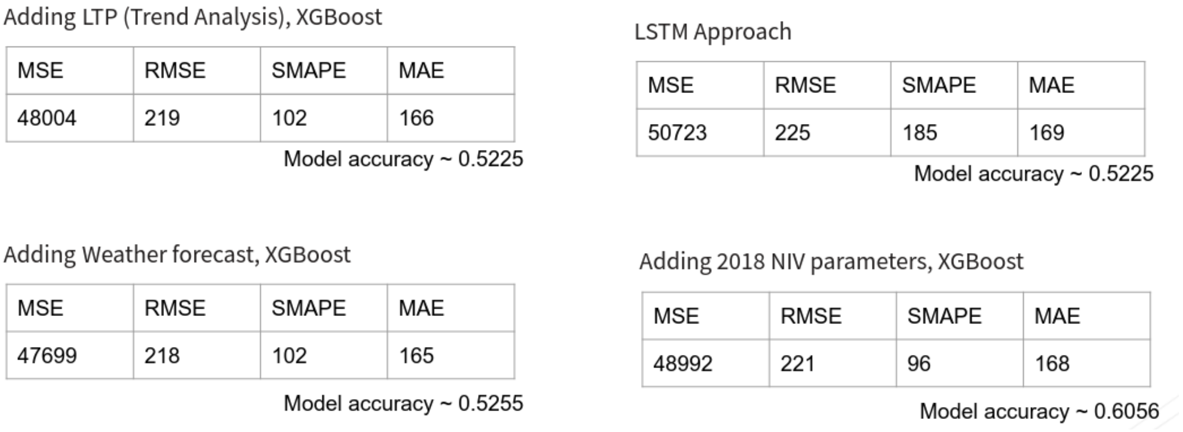
**Figure 5. Yearly variation boxplot of NIV by month**

As far as the selected machine learning algorithm is concerned, we chose Extreme Gradient Boosting algorithm to design the forecasting model, as a well-tested and time efficient algorithm. The attempt to train a deep learning model with the usage of a Long short-term Memory architecture did not accomplish to generalize towards our problem, as the model overfits, because of the limited amount of data. Furthermore, in the preprocessing phase, we applied Principal Component Analysis over the 0.95 of the variance of our data. In that way, we managed to retrieve the principal components of our data, throwing away noise and redundant information. Conducted experiments regarding the optimal parameters identification in the selected algorithms and features, achieved through grid search and cross validation.

In the effort of achieving higher accuracy, with particular market conditions, as already mentioned above, we endeavoured to incorporate more data related with forecasted photovoltaic generation and forecasted wind generation that slightly improved the forecasting error, while features like LOLP removed, because of their negligible impact on our model. Additionally, a feature engineering technique over the NetImbVol time-series data applied in terms of identifying the trend of the previous values. The chosen analysis applied a time window of the latest 10 values and estimated a metric that reflects the trend of NetImbVol. The results of the aforementioned extensions are presented in Figure 6.

Finally, we managed to incorporate more data from 2018 until the middle of 2020 leading to noticeable improvements of the indicative error metrics as they are presented below. However, the model remains unstable and inadequate to be exploitable from an Imbalance Price model.

.



Adding LTP (Trend Analysis), XGBoost

| MSE | RMSE | SMAPE | MAE |
|---|---|---|---|
| 48004 | 219 | 102 | 166 |

Model accuracy ~ 0.5225

LSTM Approach

| MSE | RMSE | SMAPE | MAE |
|---|---|---|---|
| 50723 | 225 | 185 | 169 |

Model accuracy ~ 0.5225

Adding Weather forecast, XGBoost

| MSE | RMSE | SMAPE | MAE |
|---|---|---|---|
| 47699 | 218 | 102 | 165 |

Model accuracy ~ 0.5255

Adding 2018 NIV parameters, XGBoost

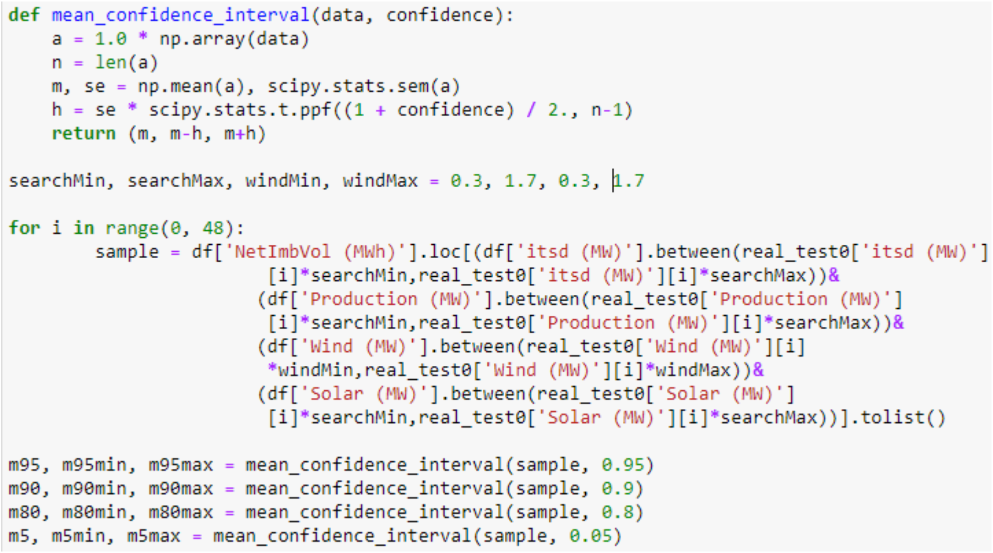
| MSE | RMSE | SMAPE | MAE |
|---|---|---|---|
| 48992 | 221 | 96 | 168 |

Model accuracy ~ 0.6056

Figure 6. Results for NIV estimation through regression

## 4.2. Quantile regression for Net Imbalance Volume

Since the NetImbVol cannot be accurately forecasted, and given the importance provided in the feature importance method contributing as the most important variable, this predictor deserves special attention. ELEXON has identified that the volume has increased over the last years steadily, however what the final value will be, is difficult to predict. Therefore, a range is set on the known predictors that are to be tested, and a query will determine the NetImbVol values for those particular observations already in the dataset. In practice, the python code resembles a few simple conditional selections as shown in Figure 7 for clarity:



```
def mean_confidence_interval(data, confidence):
    a = 1.0 * np.array(data)
    n = len(a)
    m, se = np.mean(a), scipy.stats.sem(a)
    h = se * scipy.stats.t.ppf((1 + confidence) / 2., n-1)
    return (m, m-h, m+h)

searchMin, searchMax, windMin, windMax = 0.3, 1.7, 0.3, 1.7

for i in range(0, 48):
        sample = df['NetImbVol (MWh)'].loc[(df['itsd (MW)'].between(real_test0['itsd (MW)']
                        [i]*searchMin,real_test0['itsd (MW)'][i]*searchMax))&
                       (df['Production (MW)'].between(real_test0['Production (MW)']
                        [i]*searchMin,real_test0['Production (MW)'][i]*searchMax))&
                       (df['Wind (MW)'].between(real_test0['Wind (MW)'][i]
                        *windMin,real_test0['Wind (MW)'][i]*windMax))&
                       (df['Solar (MW)'].between(real_test0['Solar (MW)']
                        [i]*searchMin,real_test0['Solar (MW)'][i]*searchMax))].tolist()

m95, m95min, m95max = mean_confidence_interval(sample, 0.95)
m90, m90min, m90max = mean_confidence_interval(sample, 0.9)
m80, m80min, m80max = mean_confidence_interval(sample, 0.8)
m5, m5min, m5max = mean_confidence_interval(sample, 0.05)
```

Figure 7. Core code for NetImbVol dataset selection

The above code looks back at historical data from the training dataset and filters the Net Imbalance Volumes observed, in days and settlement periods under similar market conditions, (production, demand, wind and solar generation). The user is able to set the 'sensitivity/tolerance' search limits in the historical data, which will subsequently return a smaller or larger sample/list of historical NIV data for each settlement period. As the dataset increases with more historical data (the reason for training the model), these limits can be adjusted in an iterative process in order to increase the accuracy. The query shown in Figure 5, returns a list of NetImbVol values to which a decomposition of quantiles (5%, 80%, 90% and 95%) is performed. The resulting sets are then input into the model, one at a time as possible NetImbVol values and different regressions are run, one for each of the quantiles. The goal is to incorporate the uncertainty related to this predictor and reflect it in the target variable prediction (Imbalance Price).

## 4.2. XGBRegressor

When choosing an algorithm, several factors must be considered depending on the problem to be solved. These factors can be the pre-processing requirements and, whether it is a time series set of data, or accuracy level acceptability. Moreover, the speed of running the model and how fast it is to train, or even its complexity, as well as the number of predictors also needs to be considered. In the current case, it is a time series with no heavy processing power required. It may take longer to train than to provide a prediction. Since the goal is to understand the dynamics and direction of the price, and not so much precisely forecast the absolute value of each SP price, we will consider an R2 score to be very low and unacceptable if under or 50% but would be very acceptable above 65%. A reasonably high number of predictors is being considered with reasonable complexity. Three algorithms were chosen, Random Forest [48], Gradient Boosting [49] and XGBoost [50]. The first two were tested but did not perform well on variable dependency and accuracy respectively, hence the XGBoost was used.

When compared to RF or GB the feature importance provided by the XGBoost presents larger variety of contributions. This is an advantage since the most important variable in the GB and RF algorithms is the NetImbVol, which is not predictable. XGBoost is a relatively recent development in machine learning, but follows the principle of gradient boosting, containing some differences in modeling details. XGBoost uses a more normalized model description to control over-fitting, which usually provides a better overall performance. Among all the hyperparameters there are typically five which are known to influence the model the most: Number of subtrees to be trained (n_estimators), maximum tree depth each tree can grow (max_depth), learning rate, reg_alpha and reg_lambda are regularization terms influencing the weight at the leaves and the scattering.

## 4.3. Data set Analysis

To check for variable independence, a correlation matrix was generated and can be seen in Figure 8. It can be observed that there is high correlation between some LOLP variables with different time horizons and also DRM variables as expected. However, since the model is to be run several times until 1h ahead of the closing gate, in order to provide as accurate estimations as soon as possible, the model will take into account all of these predictors. In addition, the high correlation between production and demand stands out, which is because one should match the other at all times. The reason for keeping both of the variables is to capture any deviations between the two, which could exist and maybe have an impact on the target variable.

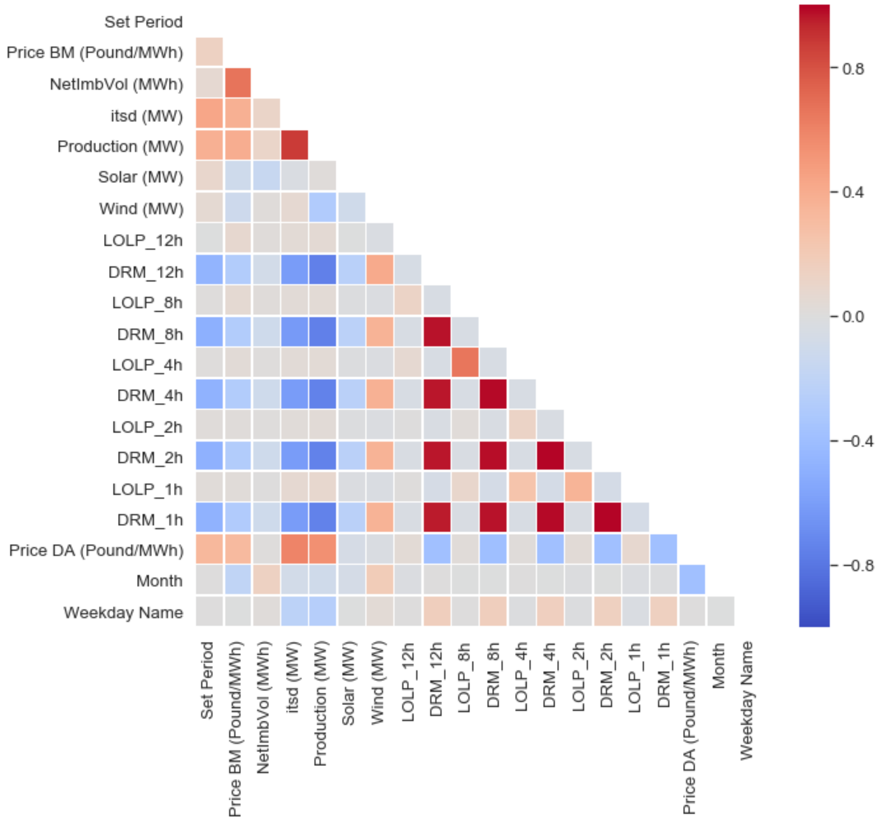


**Figure 8. Correlation Matrix of input Variables**

The dataset can be aggregated and observed in a static analysis with pivot tables, where monthly and daily profiles show clear trends. From each box plot, a statistical distribution can be derived and the corresponding parameters extracted, if such a statistical analysis approach is desired. Figure 9 shows the price variation per month, per SP and weekday, with an example of the price variation for the month of June on Tuesdays.

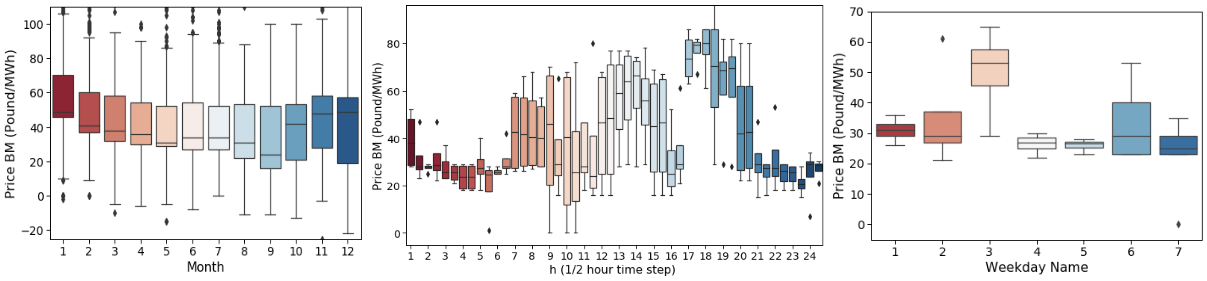


**Figure 9. Boxplots of Month (left), daily (center) and weekly (right) cycles of prices**

By analysing each weekday, the corresponding statistical distribution may be extracted. Figure 10 provides the example for a given Sunday, SP 32 and the month of June, fitting a gamma distribution, and showing also the corresponding parameters describing the distribution.

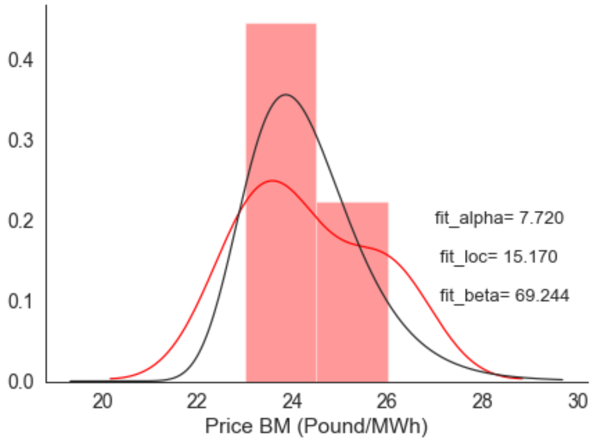


**Figure 10. Gamma Statistical distribution with parameters**

Regarding the regressions used, a 90% - 10% training and test linear split in time was performed. All three algorithms were analysed regarding its feature importance and metrics performance. The real time implementation is then developed with the best performing one.

# 5. Results and Discussion

In this study the Randomized Parameter Optimization was used, which is the randomized search cross validation (CV) method provided by the scikit-learn [51] library. The hyperparameter tuning is an intensive optimization problem, which can take several hours. Two main parameters have to be inserted as input for this exercise to be carried out, and these determine its accuracy and runtime: The number of iterations (N_iter) and the CV. N_iter is the number of parameter settings that are sampled, trading off runtime. The CV determines the cross-validation splitting strategy (for example 3 folds), which prevents the model from over fitting. The hyperparameters are provided in Table 2 for each of the algorithms run, so that the results can be replicated.

**Table 2. HyperParameters used in each algorithm**

| Methods | HyperParameters |
|---|---|
| RandomForrest | {'n_estimators': 100, 'min_samples_split': 10, 'min_samples_leaf': 2, 'max_features': 'sqrt', 'max_depth': 90, 'bootstrap': True} |
| GradientBoosting | {'subsample': 1, 'n_estimators': 642, 'min_samples_split': 7, 'min_samples_leaf': 1, 'max_depth': 14, 'learning_rate': 0.2, 'alpha':0.5} |
| XGBossting | {'subsample': 0.8, 'seed': 578, 'n_estimators': 4183, 'min_child_weight': 7, 'max_depth': 119, 'colsample_bytree': 0.5} |

Table 3 shows the three metrics assessed for each algorithm. The models show medium high R2 scores. The fact that some outliers were removed might have contributed to a low variance also visible in the mean absolute error (MAE). However when the model fails, it fails by a lot, which can be seen in the mean squared error (MSE).

**Table 3. Model Performance comparison between methods**

| Methods | $R^2$ | Mean absolute error (£) | Mean squared error (£) |
|---|---|---|---|
| RandomForrest | 80.4% | 6.97 | 105.38 |
| GradientBoosting | 78.3% | 7.49 | 116.30 |
| XGBossting | 76.8% | 7.89 | 124.74 |

It should be mentioned that the accuracies reported in Table 3, consider the ability to predict the test part of the dataset, having learnt from the training part of the dataset. To have the same accuracy with real time data would mean that the model could have access to all variables, which is not true because of the NetImbVol value being based on an estimation and quantile decomposition for regression. The feature importance reveals the variables contribution to the target variable estimation. Figure 11 shows all predictors.

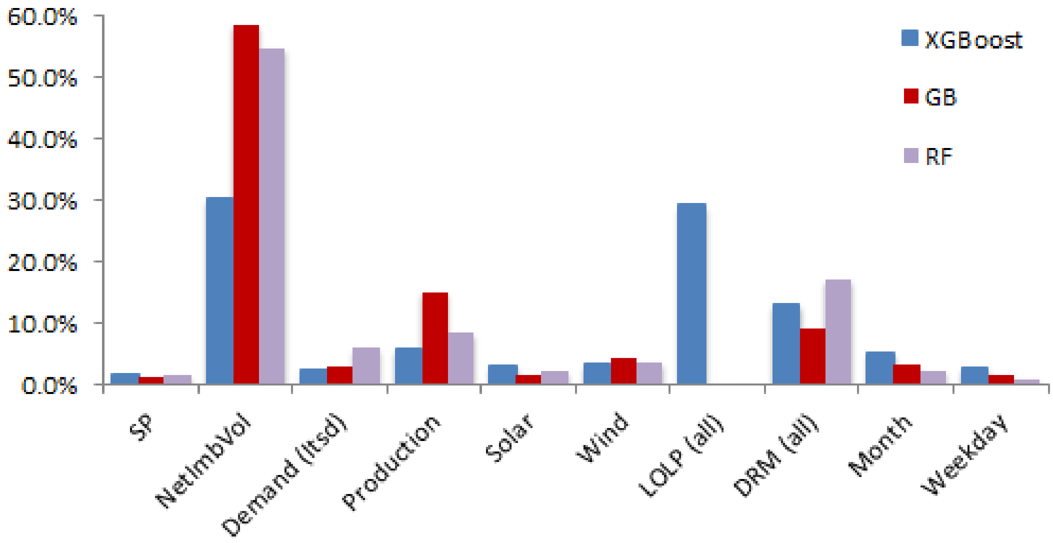


**Figure 11. Feature importance of each predictor for all three algorithms XGBoost, GB and RF**

It can easily be seen that the impact of the NetImbVol is the greatest. However, the direction in which it contributes to the model cannot be understood. Whether the price is positively or negatively impacted with the increase of the NetImbVol and at what values such change occurs, is not observable. For this reason, the partial dependences can be calculated, where one variable is observed, while maintaining the others at a constant mean value showing a sum of contributions. The SHAP library is used for this, which was built to model interpretability and used in XGBoost since it does not have by default a feature importance included. This can be seen in Figure 12, for three predictors.

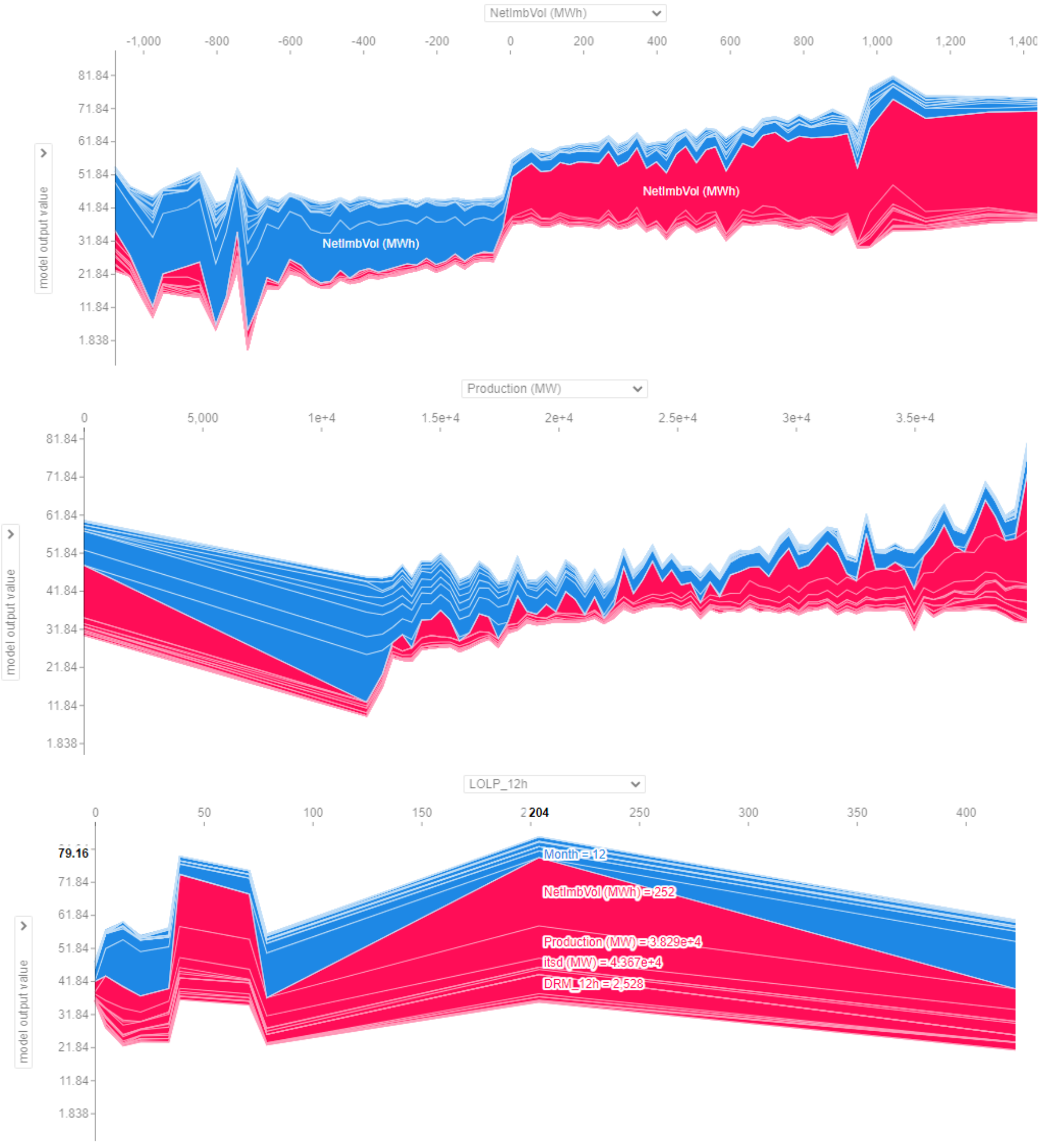


**Figure 12. Cumulative Partial dependency from the NetImbVol (upper), Production (center) and LOLP_12h predictors (bottom)**

The library also allows the representation of a dependence plot to show the effect of a single feature across the whole dataset as shown in Figure 13.

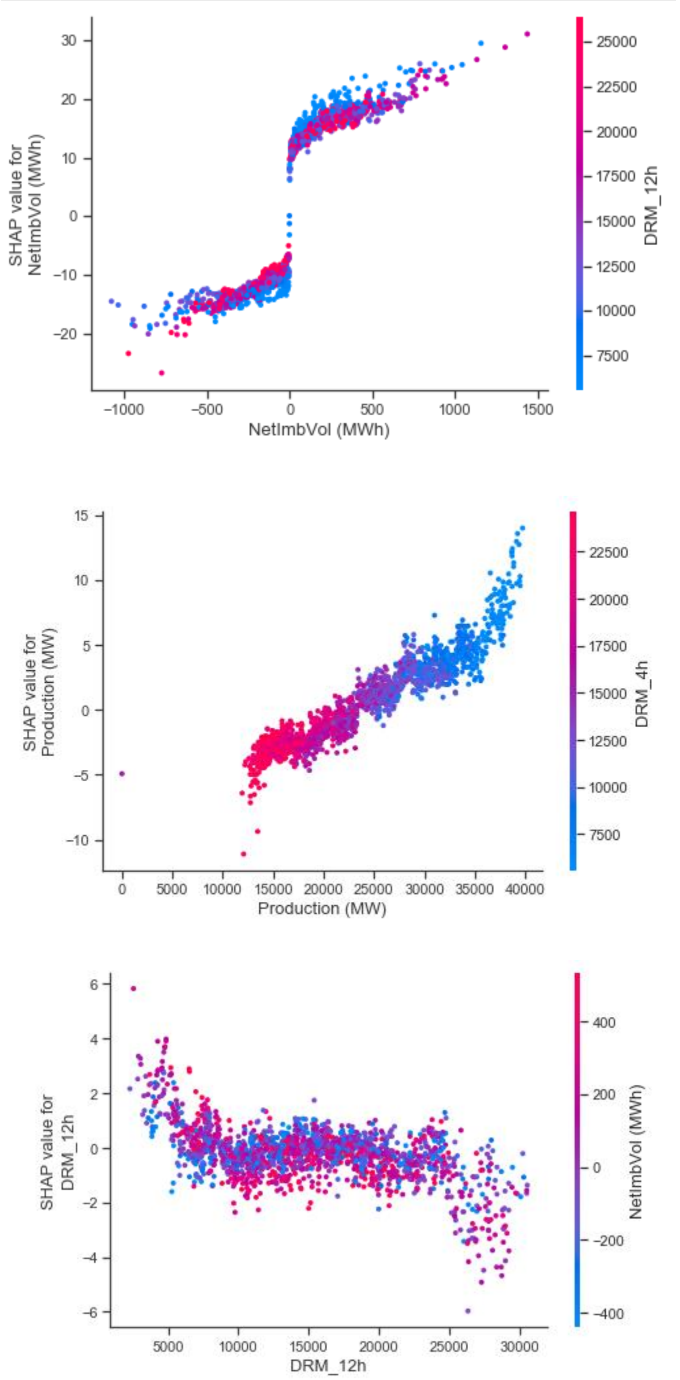


**Figure 13. Partial dependences from the NetImbVol (upper), Production (center) and DRM_12h predictors (bottom)**

The partial dependencies show non-linear behavior. Both NetImbVol and Production predictors show increasing steps in price in precise values. Such values should be monitored carefully as they tend to prompt sudden shifts in prices. Analysing the feature importance, given the high dependence on one variable (NetmbVol), which is the variable that cannot be predicted, the RF and GB algorithms were discarded. The XGBoost appears to be the most appropriate to continue the implementation and hence the real time forecast was performed using this algorithm.

Figure 14 presents the imbalance energy price trends for each of the 48 SP for the 23$^{rd}$ of June 2020. The forecast takes into account real data extracted from the ELEXON website. The only variable, which cannot be forecasted, is the Net Imbalance Volume (NetImbVol). To incorporate this uncertainty, the

95%, 90%, 80% and 5% quantiles of the query performed on the historical dataset, were subject to regressions and also shown in the Figure 14. The result is a variation of the mean prediction, which takes into account a possible fluctuation of the NetImbVol in case its value should be within the quantile range defined.

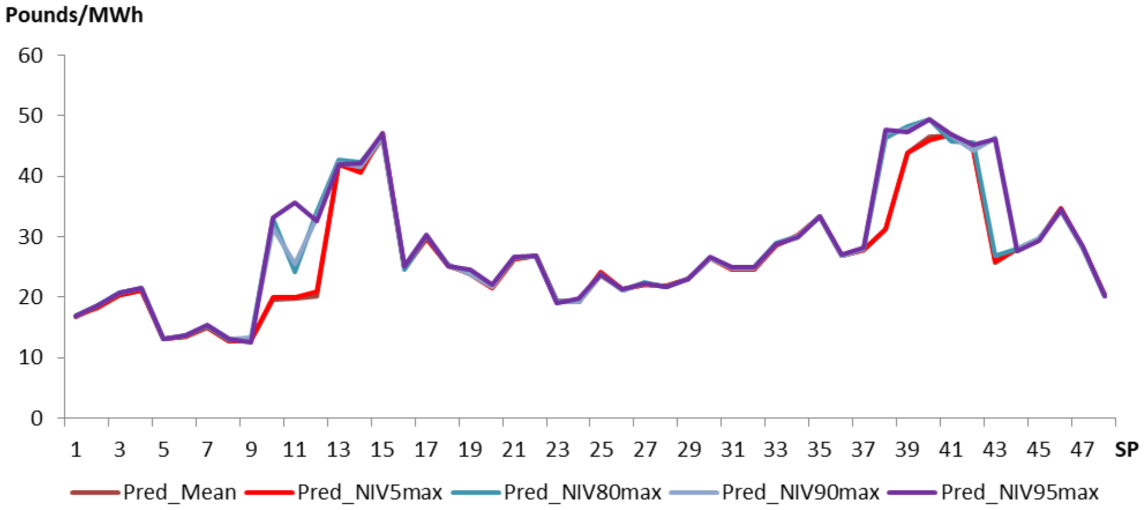


**Figure 14. Model price prediction with mean 5%, 80%, 90%, 95%, NetImbVol quantile regression**

All quantiles were compared with the real observation and checked for correlation. The highest correlation is with the 95% quantile, shown with the mean curve and real observation in Figure 15.

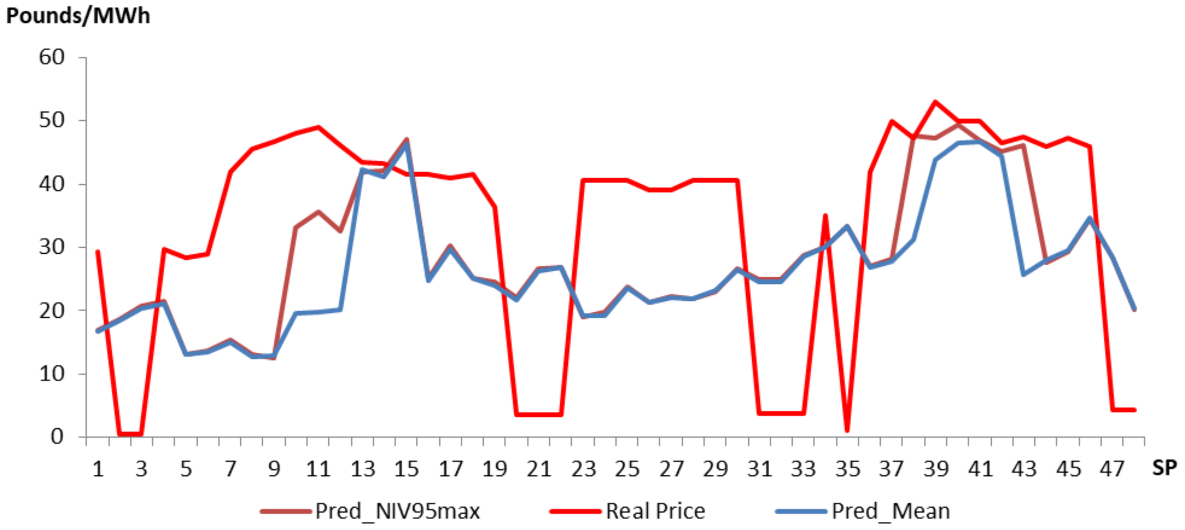


**Figure 15. Model price prediction mean and NetImbVol 95% quantile vs real data for 48 SP (test on the 23rd June 2020)**

Comparing the forecast and the real market price in both plots, one can observe two peaks of the real price, one in the morning period (SP 7 to 18) and one later in the evening (SP 36 to 46), and then a sudden drop at the end of the day also captured by the model. There is a lower high in the middle of the day from SP 23 to SP 30 and a small spike at SP 34, just before the last evening high. In terms of time span and precision of events, the dynamics of the prediction are acceptable in predicting the peaks. Regarding the amplitude, both peaks (morning and evening) are around 50 £/MWh with the second peak being slightly higher than the first, also predicted by the model. The middle peak is not predicted with a value of 23 £/MWh compared to 40.3 £/MWh in the real observation. The evening spike at SP 34, is predicts at SP35 with a 30.5 £/MWh value instead of 35 £/MWh in the real observation.

Regarding the bottom price instances throughout the day, minimum price forecasts predicted the market to clear at 12 £/MWh, while real price observations turned out to be just above 0 £/MWh. Such sudden drops while predicted in some cases, were unable to be followed by the model in terms of amplitude. However, since the interest is to have a fair sensitivity of the trend, the absolute values are less important, hence, this behaviour is acceptable. The ultimate goal is to know when to allocate the flexibility of the available assets. In this regard, the model can be used as a bidding strategy support tool. In the prediction shown in Figure 14, an aggregator should aim at allocating its DR flexible assets either from SP9 to SP15 or from SP 37 to 45. Moreover from the statistical analysis on Figure 9 (center), which refers to the month of June and the test day being a Tuesday (Figure 14), one can confirm that the evening period from 18h to 21h (SP 36 to SP 45) would be the most advantageous period to participate in the market. The accuracy of the model is sufficient for this exercise as well as the MAE. The statistical analysis approach is a useful one, especially when it comes to analysing seasonal patterns. The LOLP variable provided a useful contribution to the model accuracy, while the uncertainty generated by the NetImbVol variable, was well mitigated by the quantile approach and regression. The model will hence be provided to T4.4 in order to be integrated in the decision support tool.

# 6. Conclusion

It is very unlikely that a model can predict a market price with very high precision. Its likelihood and adoption would influence the very outcome of the market, which would make the same model useless. Instead, what can be done is an attempt to identify deterministic or quasi-deterministic variables, which may have an impact on the market. In this article, a forecasting model was developed to capture those dynamics and understand what influences the energy imbalance market price may endure. A total of 19 predictors were considered to develop a regression model using a machine learning algorithm, XGBoost. In terms of feature importance, the Net Imbalance Volume, the LOLP (aggregated), the De-rated margins (aggregated) and the month variables scored the highest, with 28.6% with 27.5%, 14.0%, and 8.9% of weight on feature importance respectively. The model has a MAE of 7.89 £/MWh, a R2 score 76.8% and a MSE of 124.74, which is acceptable for the problem being addressed. The study shows that the LOLPs are important predictors to be considered, while the uncertainty related to the NetImbVol variable can be mitigated with a quantile regression. A regression was also applied for the estimation of the NIV value, but it returned low R2 scores, hence the quantile approach was followed. Nevertheless, the NIV it remains a predictor which deserves further investigation. In the real example provided, the peaks of the daily price fluctuation were well predicted by the model and corroborated by the statistical analysis and hence one can assume that the correct SP could be potentially well identified in order to allocate the available DR flexibility. Furthermore, the amplitude of the price was predicted with an acceptable mean absolute error. However, the bottoms of the price fluctuation were far from the correct amplitude. Together with the statistical analysis, this approach could be indeed used as a support tool for market participants.

The outcome of the developed service is expected to aid the decision making process of the Aggregator, through its DSS, while trying to identify the optimal time slots for participating within the imbalance market with flexibility acquired through Demand Response schemes. As will be highlighted in D4.4 which is due M30, the DELTA Aggregator will explore potential participation in DR-markets during day-ahead and intra-day operation. Within that report, further details will be elaborated on the use of the developed service.

# References

[1] European Commission, Clean energy for all Europeans package - European Commission, Oct. 20, 2017. https://ec.europa.eu/energy/en/topics/energy-strategy/clean-energy-all- europeans (accessed Jun. 26, 2020).

[2] IEA. International Energy Agency. European Union 2020 Energy Policy Review, Paris, pp.310, 2020.

[3] Communication from the Commission to the European Parliament, the European Council, The Council, The European Economic and Social Committee and the Committee of the Regions. The European Green Deal COM/2019/640 final, Brussels, pp.24, 2019.

[4] P. H. Kuo and C. J. Huang, An electricity price forecasting model by hybrid structured deep neural networks, Sustain., vol. 10, no. 4, pp. 1–17, 2018, doi: 10.3390/su10041280.

[5] V. Sharma and D. Srinivasan, A hybrid intelligent model based on recurrent neural networks and excitable dynamics for price prediction in deregulated electricity market, Eng. Appl. Artif. Intell., vol. 26, no. 5–6, pp. 1562–1574, 2013, doi: 10.1016/j.engappai.2012.12.012.

[6] S. K. Aggarwal, L. M. Saini, and A. Kumar, Electricity price forecasting in deregulated markets: A review and evaluation, Int. J. Electr. Power Energy Syst., vol. 31, no. 1, pp. 13–22, 2009, doi: 10.1016/j.ijepes.2008.09.003.

[7] D. W. Bunn, Forecasting loads and prices in competitive power markets, Proc. IEEE, vol. 88, no. 2, pp. 163–169, Feb. 2000, doi: 10.1109/5.823996.

[8] R. Weron, Electricity price forecasting: A review of the state-of-the-art with a look into the future, Int. J. Forecast., vol. 30, no. 4, pp. 1030–1081, 2014, doi: 10.1016/j.ijforecast.2014.08.008.

[9] Hui Niu, R. Baldick, and Guidong Zhu, Supply function equilibrium bidding strategies with fixed forward contracts, IEEE Trans. Power Syst., vol. 20, no. 4, pp. 1859–1867, Nov. 2005, doi: 10.1109/TPWRS.2005.857272.

[10] C. M. Ruibal and M. Mazumdar, Forecasting the Mean and the Variance of Electricity Prices in Deregulated Markets, IEEE Trans. Power Syst., vol. 23, no. 1, pp. 25–32, Feb. 2008, doi: 10.1109/TPWRS.2007.913195.

[11] M. Borgosz-Koczwara, A. Weron, and A. Wyłomańska, Stochastic models for bidding strategies on oligopoly electricity market, Math. Methods Oper. Res., vol. 69, no. 3, pp. 579–592, 2009, doi: 10.1007/s00186-008-0252-7.

[12] K. C. Chatzidimitriou, A. C. Chrysopoulos, A. L. Symeonidis, and P. A. Mitkas, Enhancing Agent Intelligence through Evolving Reservoir Networks for Predictions in Power Stock Markets BT - Agents and Data Mining Interaction, 2012, pp. 228–247.

[13] P. Holmberg, D. Newbery, and D. Ralph, Supply function equilibria: Step functions and continuous representations, J. Econ. Theory, vol. 148, no. 4, pp. 1509–1551, 2013, doi: 10.1016/j.jet.2013.04.020.

[14] A. Kowalska-Pyzalska, K. Maciejowska, K. Suszczyński, K. Sznajd-Weron, and R. Weron, Turning green: Agent-based modeling of the adoption of dynamic electricity tariffs, Energy Policy, vol. 72, pp. 164–174, 2014, doi: 10.1016/j.enpol.2014.04.021.

[15] A. A. Ladjici, A. Tiguercha, and M. Boudour, Nash Equilibrium in a two-settlement electricity market using competitive coevolutionary algorithms, Int. J. Electr. Power Energy Syst., vol. 57, pp. 148–155, 2014, doi: 10.1016/j.ijepes.2013.11.045.

[16] M. Bierbrauer, S. Trück, and R. Weron, Modeling electricity prices with regime switching models, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 3039, pp. 859–867, 2004, doi: 10.1007/978-3-540-25944-2_111.

[17] Á. Cartea and M. G. Figueroa, Pricing in Electricity Markets: A Mean Reverting Jump Diffusion Model with Seasonality, Appl. Math. Financ., vol. 12, no. 4, pp. 313–335, 2005, doi: 10.1080/13504860500117503.

[18] M. Bierbrauer, C. Menn, S. T. Rachev, and S. Trück, Spot and derivative pricing in the EEX power market, J. Bank. Financ., vol. 31, no. 11, pp. 3462–3485, 2007, doi: 10.1016/j.jbankfin.2007.04.011.

[19] N. V. Karakatsani and D. W. Bunn, Forecasting electricity prices: The impact of fundamentals and time-varying coefficients, Int. J. Forecast., vol. 24, no. 4, pp. 764–785, 2008, doi: 10.1016/j.ijforecast.2008.09.008.

[20] J. Janczura and R. Weron, An empirical comparison of alternate regime-switching models for electricity spot prices, Energy Econ., vol. 32, no. 5, pp. 1059–1073, 2010, doi: 10.1016/j.eneco.2010.05.008.

[21] C. Albanese, H. Lo, and S. Tompaidis, A numerical algorithm for pricing electricity derivatives for jump-diffusion processes based on continuous time lattices, Eur. J. Oper. Res., vol. 222, no. 2, pp. 361–368, 2012, doi: 10.1016/j.ejor.2012.04.030.

[22] R. Weron and M. Zator, Revisiting the relationship between spot and futures prices in the Nord Pool electricity market, Energy Econ., vol. 44, pp. 178–190, 2014, doi: 10.1016/j.eneco.2014.03.007.

[23] A. Misiorek, S. Trueck, and R. Weron, Point and Interval Forecasting of Spot Electricity Prices: Linear vs. Non-Linear Time Series Models, Stud. Nonlinear Dyn. &amp; Econom., vol. 10, no. 3, pp. 1–36, 2006, [Online]. Available: https://econpapers.repec.org/RePEc:bpj:sndecm:v:10:y:2006:i:3:n:2.

[24] R. Weron and A. Misiorek, Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models, Int. J. Forecast., vol. 24, no. 4, pp. 744–763, 2008, doi: 10.1016/j.ijforecast.2008.08.004.

[25] M. Lanne, H. Lütkepohl, and K. Maciejowska, Structural vector autoregressions with Markov switching, J. Econ. Dyn. Control, vol. 34, no. 2, pp. 121–131, 2010, doi: 10.1016/j.jedc.2009.08.002.

[26] R. Weron, Modeling and forecasting electricity loads and prices: A statistical approach. 2006.

[27] R. Trück, Stefan; Weron, Rafal; Wolff, Outlier Treatment and Robust Approaches for Modeling Electricity Spot Prices, Munich Pers. RePEc Arch., no. 4711, 2007, [Online]. Available: https://mpra.ub.uni-muenchen.de/4711/.

[28] A. Boogert and D. Dupont, When Supply Meets Demand: The Case of Hourly Spot Electricity Prices, IEEE Trans. Power Syst., vol. 23, no. 2, pp. 389–398, May 2008, doi: 10.1109/TPWRS.2008.920731.

[29] E. Fanone, A. Gamba, and M. Prokopczuk, The case of negative day-ahead electricity prices, Energy Econ., vol. 35, no. May 2008, pp. 22–34, 2013, doi: 10.1016/j.eneco.2011.12.006.

[30] J. Janczura, S. Trück, R. Weron, and R. C. Wolff, Identifying spikes and seasonal components in electricity spot price data: A guide to robust modeling, Energy Econ., vol. 38, pp. 96–110, 2013, doi: 10.1016/j.eneco.2013.03.013.

[31] M. Shahidehpour, H. Yamin, and Z. Li, Market Operations in Electric Power Systems, Mark. Oper. Electr. Power Syst., no. May, 2002, doi: 10.1002/047122412x.

[32] H. Geman and A. Roncoroni, Understanding the Fine Structure of Electricity Prices, J. Bus., vol. 79, no. 3, pp. 1225–1261, Jun. 2006, doi: 10.1086/500675.

[33] I. P. Panapakidis and A. S. Dagoumas, Day-ahead electricity price forecasting via the application of artificial neural network based models, Appl. Energy, vol. 172, pp. 132–151, 2016, doi: 10.1016/j.apenergy.2016.03.089.

[34] D. Keles, J. Scelle, F. Paraschiv, and W. Fichtner, Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks, Appl. Energy, vol. 162, pp. 218–230, 2016, doi: 10.1016/j.apenergy.2015.09.087.

[35] O. A. Karabiber and G. Xydis, Electricity price forecasting in the Danish day-ahead market using the TBATS, ANN and ARIMA methods, Energies, vol. 12, no. 5, 2019, doi: 10.3390/en12050928.

[36] M. Olsson and L. Soder, Modeling Real-Time Balancing Power Market Prices Using Combined SARIMA and Markov Processes, IEEE Trans. Power Syst., vol. 23, no. 2, pp. 443–450, May 2008, doi: 10.1109/TPWRS.2008.920046.

[37] T. Jónsson, P. Pinson, H. A. Nielsen, and H. Madsen, Exponential smoothing approaches for prediction in real-time electricity markets, Energies, vol. 7, no. 6, pp. 3710–3732, 2014, doi: 10.3390/en7063710.

[38] Š. Kratochvíl and J. Bejbl, The system imbalance prediction, POSTER 2015 - 19th Int. Student Conf. Electr. Eng., pp. 1–6, 2015.

[39] J. Dumas, I. Boukas, M. M. De Villena, S. Mathieu, and B. Cornelusse, Probabilistic Forecasting of Imbalance Prices in the Belgian Context, Int. Conf. Eur. Energy Mark. EEM, vol. 2019-Septe, 2019, doi: 10.1109/EEM.2019.8916375.

[40] T. Brijs, K. De Vos, C. De Jonghe, and R. Belmans, Statistical analysis of negative prices in European balancing markets, Renew. Energy, vol. 80, pp. 53–60, 2015, doi: 10.1016/j.renene.2015.01.059.

[41] F. Ziel, R. Steinert, and S. Husmann, Efficient modeling and forecasting of electricity spot prices, Energy Econ., vol. 47, no. January 2015, pp. 98–111, 2015, doi: 10.1016/j.eneco.2014.10.012.

[42] K. Maciejowska, Fundamental and speculative shocks, what drives electricity prices?, in 11th International Conference on the European Energy Market (EEM14), May 2014, pp. 1–5, doi: 10.1109/EEM.2014.6861289.

[43] N. Amjady and M. Hemmati, Energy price forecasting - problems and proposals for such predictions, IEEE Power Energy Mag., vol. 4, no. 2, pp. 20–29, Mar. 2006, doi: 10.1109/MPAE.2006.1597990.

[44] ELEXON. https://www.ELEXON.co.uk/ (accessed Jun. 27, 2020).

[45] ELEXON. https://www.ELEXON.co.uk/about/about-ELEXON/ (accessed Jun. 27, 2020).

[46] European Commission. Final report of the sector inquiry on capacity mechanisms, Brussels, pp.18, 2016.

[47] ELEXON Limited 2019, Loss of Load Probability Calculation Statement, Balanc. Settl. Code, no. March, pp. 5–10, 2019.

[48] Breiman, L. Random Forest. Mach. Learn. 2001, 45, 5–32.

[49] Friedman, J.H. Greedy Function Approximation: A gradient Boosting Machine. Ann. Stat. 2001, *29*, 1189-1232

[50] Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD. International Conference on Knowledge Discovery and Data Mining, KDD16, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794

[51] SKlearn. http://scikit-learn.org/stable/ (accessed Oct. 31, 2018).

## ANNEX A: Python Code for the Price Imbalance Forecast

```
#Import the following libraries:
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
from sklearn.metrics import accuracy_score
from sklearn.metrics import r2_score
from sklearn.metrics import adjusted_rand_score
from sklearn.metrics import explained_variance_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import precision_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn import ensemble
import xgboost as xgb
from xgboost import XGBRegressor
from sklearn.model_selection import RandomizedSearchCV
import timeit
import seaborn as sns
from pandas_profiling import ProfileReport

df=pd.read_csv("C:\BalanceUKPrice2019.csv", infer_datetime_format=True, decimal='.',
index_col=0, parse_dates=True)
df.info()
```

```
#Output
""
DatetimeIndex: 17520 entries, 2019-01-01 to 2019-12-31
Data columns (total 34 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Set Period            17520 non-null  int64
 1   Price BM (Pound/MWh)  17520 non-null  float64
 2   SBP                   17520 non-null  float64
 3   BD                    17520 non-null  object
 4   PDC                  17520 non-null  object
 5   RSP                  6913 non-null   float64
 6   NetImbVol (MWh)       17520 non-null  float64
 7   SPA                  17520 non-null  int64
 8   BPA                  17520 non-null  float64
 9   RP                   1867 non-null   float64
 10  RPRV                 1867 non-null   float64
 11  OV                   17520 non-null  float64
 12  BV                   17520 non-null  float64
 13  TotOfferVol          17520 non-null  float64
 14  TotBidVol            17520 non-null  float64
 15  ASV                  17520 non-null  float64
 16  ABV                  17520 non-null  float64
 17  TotAccSellVol        17520 non-null  float64
 18  TotAccBidVol         17520 non-null  float64
 19  itsd (MW)            17520 non-null  int64
 20  Production (MW)      17520 non-null  int64
 21  Solar (MW)           17520 non-null  int64
 22  Wind (MW)            17520 non-null  float64
 23  LOLP_12h             17520 non-null  float64
 24  DRM_12h              17520 non-null  int64
 25  LOLP_8h              17520 non-null  float64
 26  DRM_8h               17520 non-null  int64
 27  LOLP_4h              17520 non-null  float64
 28  DRM_4h               17520 non-null  int64
 29  LOLP_2h              17520 non-null  float64
 30  DRM_2h               17520 non-null  int64
 31  LOLP_1h              17520 non-null  float64
 32  DRM_1h               17520 non-null  int64
 33  Price DA (Pound/MWh)  17520 non-null  float64
""
df.drop(df.columns[[2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18]], axis=1, inplace=True)
df["Date"] = pd.to_datetime(df.index,infer_datetime_format=True, dayfirst=True)
df.fillna((0), inplace=True)
df["Price BM (Pound/MWh)"]=df["Price BM (Pound/MWh)"].astype('int32')
df["NetImbVol (MWh)"]=df["NetImbVol (MWh)"].astype('int32')
df["Production (MW)"]=df["Production (MW)"].astype('int32')
df["Solar (MW)"]=df["Solar (MW)"].astype('int32')
df["Wind (MW)"]=df["Wind (MW)"].astype('int32')
df["LOLP_12h"]=df["LOLP_12h"].astype('int32')
df["DRM_12h"]=df["DRM_12h"].astype('int32')
df["LOLP_8h"]=df["LOLP_8h"].astype('int32')
df["DRM_8h"]=df["DRM_8h"].astype('int32')
df["LOLP_4h"]=df["LOLP_4h"].astype('int32')
df["DRM_4h"]=df["DRM_4h"].astype('int32')
df["LOLP_2h"]=df["LOLP_2h"].astype('int32')
df["DRM_2h"]=df["DRM_2h"].astype('int32')
df["LOLP_1h"]=df["LOLP_1h"].astype('int32')
```

```
df["DRM_1h"]=df["DRM_1h"].astype('int32')
df["Price DA (Pound/MWh)"]=df["Price DA (Pound/MWh)"].astype('int32')

df['Year'] = df.index.year
df['Month'] = df.index.month
df['Weekday Name'] = df.index.day_name()
df.replace({'Weekday Name': {'Monday': 1, 'Tuesday': 2, 'Wednesday': 3, 'Thursday': 4,'Friday': 5, 'Saturday': 6, 'Sunday': 7}}, inplace=True)
df.drop("Date", axis=1, inplace=True)

q_low = df["Price BM (Pound/MWh)"].quantile(0.0025)
q_hi  = df["Price BM (Pound/MWh)"].quantile(0.9975)

df= df[(df["Price BM (Pound/MWh)"] < q_hi) & (df["Price BM (Pound/MWh)"] > q_low)]
df.drop("Year", axis=1, inplace=True)

y=df.pop("Price BM (Pound/MWh)")
y=pd.DataFrame(y)
y=y.values.reshape(-1,1)
y.shape

X=df
X=pd.DataFrame(X)

train_X, test_X= np.split(X, [int(.90 *len(X))])
train_y, test_y= np.split(y, [int(.90 *len(y))])


%%time
parameters = {'subsample': 0.8,
 'seed': 578,
 'n_estimators': 4183,
 'min_child_weight': 7,
 'max_depth': 11,
 'colsample_bytree': 0.5}
clf=XGBRegressor(**parameters)
clf.fit(train_X, train_y)

%%time
test_data=(test_X)
prediction=clf.predict(test_data)
print("The expected balancing energy price is:", prediction, "£/MWh")

prediction=prediction.astype(int)
prediction=pd.DataFrame(prediction)
test_y=pd.DataFrame(test_y)

prediction.astype(int)
```

```
clf.feature_importances_

import matplotlib

sns.set(font_scale=1.5, style='white')
feature_importances = pd.Series(clf.feature_importances_, index=X.columns)
feature_importances.sort_index
feature_importances.plot(kind="barh", figsize=(7,6))

import shap

# load JS visualization code to notebook
shap.initjs()

# train XGBoost model
#X,y = shap.datasets.boston()
#model = clf.train({"learning_rate": 0.01}, clf.DMatrix(X, label=y), 100)

# explain the model's predictions using SHAP
# (same syntax works for LightGBM, CatBoost, scikit-learn and spark models)
explainer = shap.TreeExplainer(clf)
shap_values = explainer.shap_values(test_X)

# visualize the first prediction's explanation (use matplotlib=True to avoid Javascript)
shap.force_plot(explainer.expected_value, shap_values[0,:], test_X.iloc[0,:])
shap.force_plot(explainer.expected_value, shap_values, test_X)
shap.dependence_plot("NetImbVol (MWh)", shap_values, test_X)
shap.dependence_plot("Production (MW)", shap_values, test_X)
shap.dependence_plot("DRM_12h", shap_values, test_X)
shap.summary_plot(shap_values, test_X)
shap.summary_plot(shap_values, test_X, plot_type="bar")

#Metrics Output
r2_score(test_y, prediction, multioutput='variance_weighted')
clf.score(test_X, test_y)
print(explained_variance_score(test_y, prediction))
print(mean_absolute_error(test_y, prediction))
mse = mean_squared_error(test_y, prediction)
print("MSE: %.4f" % mse)

#Hyper Parameter Calculation
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 500, stop = 10000, num = 50)]
# Maximum number of levels in tree
max_depth = [3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33]
# Minimum number of samples required
subsample = [0.5, 0.6, 0.7, 0.8, 0.9]
colsample_bytree = [0.5, 0.6, 0.7, 0.8, 0.9]
```

```
min_child_weight= [1,3,4,5,7]
seed=[int(i) for i in np.linspace(start = 0, stop = 1000, num = 20)]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
            'max_depth': max_depth,
            "subsample":subsample,
            'colsample_bytree': colsample_bytree,
            'min_child_weight': min_child_weight, "seed":seed}

#Include the output parameters in the model parameters
clf_random.best_params_

#Use the API or Scrapping tools to retrieve the data from the Elexon Website
from LoLP_Elexon_Scrapping1 import df3_result
from SystemDemandandProductionElexonScrapping import df5_result, df4_result
from WindSolarForecast_ElexonScrapping2 import df6_solar, df6_wind
from DA_Price_Scrapping2 import df7

df3_result["Settlement Period"]=pd.to_numeric(df3_result["Settlement Period"])
df5_result.dtypes
df6_solar["Settlement Period"]=pd.to_numeric(df6_solar["Settlement Period"])
df6_wind.dtypes
real_test3 = pd.merge(df3_result, df5_result, on='Settlement Period')
real_test2=pd.merge(real_test3, df6_solar,on='Settlement Period')
real_test1=pd.merge(real_test2, df6_wind, on='Settlement Period')

real_test00=pd.merge(real_test1, df4_result, on='Settlement Period')
real_test0=pd.merge(real_test00, df7, on='Settlement Period')
real_test0["Date"] = pd.to_datetime(real_test0["Date"],infer_datetime_format=True)

real_test0["Date"].dtype
real_test0.set_index("Date", inplace=True)

real_test0['Month'] = real_test0.index.month
real_test0['Weekday Name'] = real_test0.index.day_name()
real_test0.replace({'Weekday Name': {'Monday': 1, 'Tuesday': 2, 'Wednesday': 3, 'Thursday': 4,'Friday':
5, 'Saturday': 6, 'Sunday': 7}}, inplace=True)
real_test0.drop(real_test0.columns[[11,13,16,17]], axis=1, inplace=True)
real_test0.head(3)

real_test0.rename(columns={"Settlement Period":"Set Period","12h LoLP":"LOLP_12h","12h
DRM":"DRM_12h","8h LoLP":"LOLP_8h",
                "8h DRM":"DRM_8h","4h LoLP":"LOLP_4h","4h DRM":"DRM_4h","2h
LoLP":"LOLP_2h",
                 "2h DRM":"DRM_2h","1h LoLP":"LOLP_1h","1h DRM":"DRM_1h","Quantity
(MW)":"Production (MW)",
                 "Day Ahead (MW)_x":"Solar (MW)","Day Ahead (MW)_y":"Wind (MW)","TSDF
(MW)":"itsd (MW)", "Price DA (Pound/MWh)":"Price DA (Pound/MWh)"}, inplace=True)
```

```
real_test0["Solar (MW)"]=real_test0["Solar (MW)"].astype('int32')

real_test0=pd.DataFrame(real_test0)

#NET Imb Vol Estimation based on quantiles

import numpy as np
import scipy.stats
from scipy import stats


def mean_confidence_interval(data, confidence):
    a = 1.0 * np.array(data)
    n = len(a)
    m, se = np.mean(a), scipy.stats.sem(a)
    h = se * scipy.stats.t.ppf((1 + confidence) / 2., n-1)
    return (m, m-h, m+h)

searchMin = 0.3
searchMax = 1.7
windMin = 0.3
windMax = 1.5

a=pd.DataFrame()

real_test0['NIVmean']=0.0
real_test0['NIV95max']=0.0
real_test0['NIV95min']=0.0
real_test0['NIV90max']=0.0
real_test0['NIV90min']=0.0
real_test0['NIV80max']=0.0
real_test0['NIV80min']=0.0
real_test0['NIV5max']=0.0
real_test0['NIV5min']=0.0

for i in range(0, 48):
        sample = df['NetImbVol (MWh)'].loc[(df['itsd (MW)'].between(real_test0['itsd (MW)']
                        [i]*searchMin,real_test0['itsd (MW)'][i]*searchMax))&
                       (df['Production (MW)'].between(real_test0['Production (MW)']
                        [i]*searchMin,real_test0['Production (MW)'][i]*searchMax))&
                       (df['Wind (MW)'].between(real_test0['Wind (MW)'][i]
                        *windMin,real_test0['Wind (MW)'][i]*windMax))&
                       (df['Solar (MW)'].between(real_test0['Solar (MW)']
                        [i]*searchMin,real_test0['Solar (MW)'][i]*searchMax))].tolist()

        m95, m95min, m95max = mean_confidence_interval(sample, 0.95)
        m90, m90min, m90max = mean_confidence_interval(sample, 0.9)
        m80, m80min, m80max = mean_confidence_interval(sample, 0.8)
        m5, m5min, m5max = mean_confidence_interval(sample, 0.05)

        real_test0['NIVmean'][i] = m95
        real_test0['NIV95max'][i] = m95max
        real_test0['NIV95min'][i] = m95min
        real_test0['NIV90max'][i] = m90max
        real_test0['NIV90min'][i] = m90min
        real_test0['NIV80max'][i] = m80max
        real_test0['NIV80min'][i] = m80min
        real_test0['NIV5max'][i] = m5max
        real_test0['NIV5min'][i] = m5min

        p5 = np.percentile(sorted(sample), 5)
        p25 = np.percentile(sorted(sample), 25)
        p50 = np.percentile(sorted(sample), 50)
        p75 = np.percentile(sorted(sample), 75)
        p95 = np.percentile(sorted(sample), 95)
        meann = np.mean(sorted(sample))

real_test_mean=real_test0.drop(real_test0.columns[[19,20,21,22,23,24,25,26]], axis=1)
real_test_mean.rename(columns={"NIVmean":"NetImbVol (MWh)"},inplace=True)
real_test_mean = real_test_mean[['Set Period', 'NetImbVol (MWh)', 'itsd (MW)', 'Production (MW)',
'Solar (MW)', 'Wind (MW)', 'LOLP_12h', 'DRM_12h', 'LOLP_8h', 'DRM_8h', 'LOLP_4h', 'DRM_4h',
'LOLP_2h', 'DRM_2h', 'LOLP_1h', 'DRM_1h','Price DA (Pound/MWh)', 'Month', 'Weekday Name']]

real_test_NIV95max=real_test0.drop(real_test0.columns[[20,21,22,23,24,25,26]], axis=1)
real_test_NIV95max.rename(columns={"NIV95max":"NetImbVol (MWh)"},inplace=True)
real_test_NIV95max = real_test_NIV95max[['Set Period', 'NetImbVol (MWh)', 'itsd (MW)',
'Production (MW)', 'Solar (MW)', 'Wind (MW)', 'LOLP_12h', 'DRM_12h', 'LOLP_8h', 'DRM_8h',
'LOLP_4h', 'DRM_4h', 'LOLP_2h', 'DRM_2h', 'LOLP_1h', 'DRM_1h','Price DA (Pound/MWh)',
'Month', 'Weekday Name']]

real_test_NIV95min=real_test0.drop(real_test0.columns[[19,21,22,23,24,25,26]], axis=1)
real_test_NIV95min.rename(columns={"NIV95min":"NetImbVol (MWh)"},inplace=True)
real_test_NIV95min = real_test_NIV95min[['Set Period', 'NetImbVol (MWh)', 'itsd (MW)', 'Production
(MW)', 'Solar (MW)', 'Wind (MW)', 'LOLP_12h', 'DRM_12h', 'LOLP_8h', 'DRM_8h', 'LOLP_4h',
'DRM_4h', 'LOLP_2h', 'DRM_2h', 'LOLP_1h', 'DRM_1h','Price DA (Pound/MWh)', 'Month', 'Weekday
Name']]

real_test_NIV90max=real_test0.drop(real_test0.columns[[19,20,22,23,24,25,26]], axis=1)
real_test_NIV90max.rename(columns={"NIV90max":"NetImbVol (MWh)"},inplace=True)
real_test_NIV90max = real_test_NIV90max[['Set Period', 'NetImbVol (MWh)', 'itsd (MW)',
'Production (MW)', 'Solar (MW)', 'Wind (MW)', 'LOLP_12h', 'DRM_12h', 'LOLP_8h', 'DRM_8h',
'LOLP_4h', 'DRM_4h', 'LOLP_2h', 'DRM_2h', 'LOLP_1h', 'DRM_1h','Price DA (Pound/MWh)',
'Month', 'Weekday Name']]

real_test_NIV90min=real_test0.drop(real_test0.columns[[19,20,21,23,24,25,26]], axis=1)
```

```
real_test_NIV90min.rename(columns={"NIV90min":"NetImbVol (MWh)"},inplace=True)
real_test_NIV90min = real_test_NIV90min[['Set Period', 'NetImbVol (MWh)', 'itsd (MW)', 'Production (MW)', 'Solar (MW)', 'Wind (MW)', 'LOLP_12h', 'DRM_12h', 'LOLP_8h', 'DRM_8h', 'LOLP_4h', 'DRM_4h', 'LOLP_2h', 'DRM_2h', 'LOLP_1h', 'DRM_1h','Price DA (Pound/MWh)', 'Month', 'Weekday Name']]

real_test_NIV80max=real_test0.drop(real_test0.columns[[19,20,21,22,24,25,26]], axis=1)
real_test_NIV80max.rename(columns={"NIV80max":"NetImbVol (MWh)"},inplace=True)
real_test_NIV80max = real_test_NIV80max[['Set Period', 'NetImbVol (MWh)', 'itsd (MW)', 'Production (MW)', 'Solar (MW)', 'Wind (MW)', 'LOLP_12h', 'DRM_12h', 'LOLP_8h', 'DRM_8h', 'LOLP_4h', 'DRM_4h', 'LOLP_2h', 'DRM_2h', 'LOLP_1h', 'DRM_1h','Price DA (Pound/MWh)', 'Month', 'Weekday Name']]

real_test_NIV80min=real_test0.drop(real_test0.columns[[19,20,21,22,23,25,26]], axis=1)
real_test_NIV80min.rename(columns={"NIV80min":"NetImbVol (MWh)"},inplace=True)
real_test_NIV80min = real_test_NIV80min[['Set Period', 'NetImbVol (MWh)', 'itsd (MW)', 'Production (MW)', 'Solar (MW)', 'Wind (MW)', 'LOLP_12h', 'DRM_12h', 'LOLP_8h', 'DRM_8h', 'LOLP_4h', 'DRM_4h', 'LOLP_2h', 'DRM_2h', 'LOLP_1h', 'DRM_1h','Price DA (Pound/MWh)', 'Month', 'Weekday Name']]

real_test_NIV5max=real_test0.drop(real_test0.columns[[19,20,21,22,23,24,26]], axis=1)
real_test_NIV5max.rename(columns={"NIV5max":"NetImbVol (MWh)"},inplace=True)
real_test_NIV5max = real_test_NIV5max[['Set Period', 'NetImbVol (MWh)', 'itsd (MW)', 'Production (MW)', 'Solar (MW)', 'Wind (MW)', 'LOLP_12h', 'DRM_12h', 'LOLP_8h', 'DRM_8h', 'LOLP_4h', 'DRM_4h', 'LOLP_2h', 'DRM_2h', 'LOLP_1h', 'DRM_1h','Price DA (Pound/MWh)', 'Month', 'Weekday Name']]

real_test_NIV5min=real_test0.drop(real_test0.columns[[19,20,21,22,23,24,25]], axis=1)
real_test_NIV5min.rename(columns={"NIV5min":"NetImbVol (MWh)"},inplace=True)
real_test_NIV5min = real_test_NIV5min[['Set Period', 'NetImbVol (MWh)', 'itsd (MW)', 'Production (MW)', 'Solar (MW)', 'Wind (MW)', 'LOLP_12h', 'DRM_12h', 'LOLP_8h', 'DRM_8h', 'LOLP_4h', 'DRM_4h', 'LOLP_2h', 'DRM_2h', 'LOLP_1h', 'DRM_1h','Price DA (Pound/MWh)', 'Month', 'Weekday Name']]

#Predicting mean and quantiles
predict_mean=clf.predict(real_test_mean)
print("The expected mean balancing energy price for the Day", real_test3.Date[0], "per SP is:\n\n", predict_mean, "£/MWh \n")
predict_mean=pd.DataFrame(predict_mean)
predict_mean.to_csv("predict_mean.csv")

predict_NIV95max=clf.predict(real_test_NIV95max)
print("The expected NIV95max balancing energy price for the Day", real_test3.Date[0], "per SP is:\n\n", predict_NIV95max, "£/MWh \n")
predict_NIV95max=pd.DataFrame(predict_NIV95max)
predict_NIV95max.to_csv("predict_NIV95max.csv")

predict_NIV95min=clf.predict(real_test_NIV95min)
```

```
print("The expected NIV95min balancing energy price for the Day", real_test3.Date[0], "per SP is:\n\n",
predict_NIV95min, "£/MWh \n")
predict_NIV95min=pd.DataFrame(predict_NIV95min)
predict_NIV95min.to_csv("predict_NIV95min.csv")

predict_NIV90max=clf.predict(real_test_NIV90max)
print("The expected NIV90max balancing energy price for the Day", real_test3.Date[0], "per SP is:\n\n",
predict_NIV90max, "£/MWh \n")
predict_NIV90max=pd.DataFrame(predict_NIV90max)
predict_NIV90max.to_csv("predict_NIV90max.csv")

predict_NIV90min=clf.predict(real_test_NIV90min)
print("The expected NIV90min balancing energy price for the Day", real_test3.Date[0], "per SP is:\n\n",
predict_NIV90min, "£/MWh \n")
predict_NIV90min=pd.DataFrame(predict_NIV90min)
predict_NIV90min.to_csv("predict_NIV90min.csv")

predict_NIV80max=clf.predict(real_test_NIV80max)
print("The expected NIV80max balancing energy price for the Day", real_test3.Date[0], "per SP is:\n\n",
predict_NIV80max, "£/MWh \n")
predict_NIV80max=pd.DataFrame(predict_NIV80max)
predict_NIV80max.to_csv("predict_NIV80max.csv")

predict_NIV80min=clf.predict(real_test_NIV80min)
print("The expected NIV80min balancing energy price for the Day", real_test3.Date[0], "per SP is:\n\n",
predict_NIV80min, "£/MWh \n")
predict_NIV80min=pd.DataFrame(predict_NIV80min)
predict_NIV80min.to_csv("predict_NIV80min.csv")

predict_NIV5max=clf.predict(real_test_NIV5max)
print("The expected NIV5max balancing energy price for the Day", real_test3.Date[0], "per SP is:\n\n",
predict_NIV5max, "£/MWh \n")
predict_NIV5max=pd.DataFrame(predict_NIV5max)
predict_NIV5max.to_csv("predict_NIV5max.csv")

predict_NIV5min=clf.predict(real_test_NIV5min)
print("The expected NIV5min balancing energy price for the Day", real_test3.Date[0], "per SP is:\n\n",
predict_NIV5min, "£/MWh \n")
predict_NIV5min=pd.DataFrame(predict_NIV5min)
predict_NIV5min.to_csv("predict_NIV5min.csv")

predict_mean=pd.DataFrame(predict_mean)
real_test_mean.rename(columns={0:'Predicted Price for Today'},inplace=True)

plt.figure(figsize=(15, 7))
sns.lineplot(data=predict_mean)
sns.lineplot(data=predict_NIV95max)
sns.lineplot(data=predict_NIV95min)
sns.lineplot(data=predict_NIV90max)
sns.lineplot(data=predict_NIV90min)
sns.lineplot(data=predict_NIV80max)
sns.lineplot(data=predict_NIV80min)
sns.lineplot(data=predict_NIV5max)
sns.lineplot(data=predict_NIV5min)
```