Master Thesis

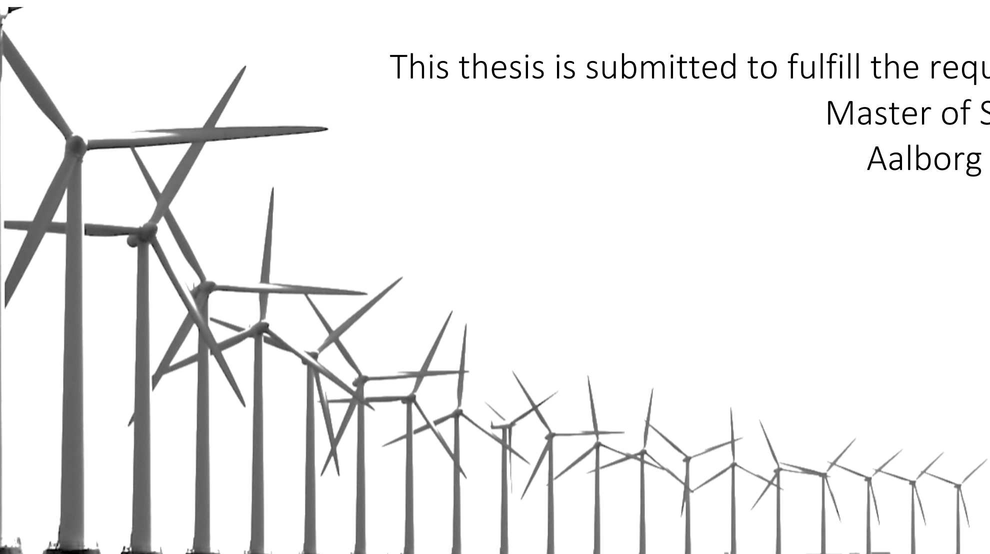
Forecasting DK1 Electricity Prices: Comparison and Performance Evaluation of ARIMAX and XGBoost.

Author:
Manik Chandra Prosad
ID: 20220898

Supervisor:
Frederik Steen Lundtofte
Professor, AAUBS

This thesis is submitted to fulfill the requirements for the degree of
Master of Science (MSc.) in Finance at
Aalborg University Business School.

December 1, 2024



Abstract

Electricity price forecasting has become increasingly critical in dynamic and volatile markets. Market like Denmark's DK1 zone is one of them which is mostly driven by the rising integration of renewable energy sources and external geopolitical influences. This thesis investigates the comparative performance of two forecasting models- ARIMAX (AutoRegressive Integrated Moving Average with Exogenous Variables) and XGBoost (Extreme Gradient Boosting) on electricity elspot prices. The analysis integrates both statistical and machine learning approaches, leveraging a dataset containing hourly electricity spot prices DK1 and energy production variables. The analysis begins with pre-model testing for stationarity, autocorrelation and normality of time series data for ARIMAX modeling. Afterwards, the ARIMAX and XGBoost models are optimized through AIC, BIC, grid search, cross-validation, optimal parameter selection and so on. Later, this thesis examines the performance of both models for forecasting electricity spot prices. Focusing on short-term and long-term spot price predictions, this paper highlights the necessity of robust forecasting models to capture the volatile characteristics of electricity prices. The empirical findings reveal that ARIMAX performs highly accurate for in-sample predictions but struggles with out-of-sample generalization where XGBoost outperforms ARIMAX in forecasting accuracy for unseen data.

Keywords: DK1, ARIMAX, XGBoost, Elspot forecasting.

Acknowledgment

I would like to express my gratitude to my supervisor, Professor Frederik Steen Lundtofte, for his invaluable guidance and continuous support throughout the completion of my thesis. His insightful feedback and direction have been instrumental in the success of this research.
I extend my thanks to Douglas Eduardo Turatti whose expertise and in depth lessons in the *Empirical Finance* module helped me understand the ARIMA time series model and its practical implementation in energy price prediction.

Special thanks go to my friend, Benjamin Ly, for his cooperation in developing the time-based dataset using Python and for helping me gain a foundational understanding of the Machine Learning approach-Extreme Gradient Boosting (XGBoost).

Finally, I would like to thank Aalborg University Business School for providing an excellent academic environment and resources that made this research possible.

Table of Contents

Abbreviations

ACER : Agency for the Cooperation of Energy Regulators (European regulatory authority ensuring energy market transparency and cooperation).

ACF : Autocorrelation Function (Measures correlation between time series and its lagged values).

ADF : Augmented Dickey-Fuller Test (Statistical test for checking stationarity in time series).

AIC : Akaike Information Criterion (Model selection metric penalizing complexity).

ANN : Artificial Neural Network (Machine learning model using brain neural structures for pattern recognition).

AR : Autoregressive (Uses past values to predict future values in time series).

ARIMA : Autoregressive Integrated Moving Average (Forecasting model using AR, integration, and MA components).

ARIMAX : Autoregressive Integrated Moving Average with Exogenous Variables (ARIMA model incorporating external explanatory variables).

BIC : Bayesian Information Criterion (Model selection metric balancing fit and simplicity).

DA: Day-Ahead (Electricity market where prices are determined a day in advance).

DK1 : Denmark-West (Electricity market region for western Denmark).

DK2 : Denmark-East (Electricity market region for eastern Denmark).

DM : Diebold-Mariano Test (Statistical test comparing forecasting accuracy of two models).

EDA : Exploratory Data Analysis (Initial step to summarize and understand data characteristics).

EIA : Energy Information Administration (US organization providing energy statistics and analysis).

EPAD : Electricity Price Area Differentials (Financial instruments for hedging price differences across zones).

EPEX : European Power Exchange (Market platform for trading electricity across Europe).

GARCH : Generalized Autoregressive Conditional Heteroskedasticity (Model for volatility prediction in time series).

GRU : Gated Recurrent Unit (Neural network architecture for handling sequence data efficiently).

IEA : International Energy Agency (Global authority offering policy advice on energy security and sustainability).

LB : Ljung-Box Test (Statistical test for detecting autocorrelation in residuals).

JB : Jarque-Bera Test (Test for normality in statistical distributions).

LNG : Liquefied Natural Gas (Natural gas cooled into liquid for easier storage and transportation).

LSTM : Long Short-Term Memory (Recurrent neural network model for sequence prediction).

MA : Moving Average (Model component smoothing time series data by averaging over time).

MAE : Mean Absolute Error (Evaluation metric calculating average absolute differences between predicted and actual values).

MSE : Mean Squared Error (Metric measuring the average squared difference between predicted and actual values).

MWh : Megawatt-hour (Unit of energy representing one megawatt of electricity used for one hour).

PACF : Partial Autocorrelation Function (Measures correlation between time series and lagged values excluding intermediate correlations).

PPA : Power Purchase Agreement (Contract between electricity buyer and seller defining pricing and volume terms).

RMSE : Root Mean Squared Error (Square root of MSE; measures the average prediction error magnitude).

R^2 : R-Squared or Coefficient of Determination (Statistical metric showing the proportion of variance explained by the model).

REMIT : Regulation of wholesale Energy Market Integrity and Transparency (EU regulation ensuring transparency and integrity in energy markets).

RW : Random Walk (A stochastic process where the current value is determined or predicted with the previous value plus a random step,)

SVM : Support Vector Machine (Machine learning model used for classification and regression tasks).

TSO : Transmission System Operator (Entity managing the high-voltage electricity grid).

WN : White Noise (Random data with constant mean and variance over time).

XGBoost : Extreme Gradient Boosting (Highly efficient machine learning algorithm for structured data regression and classification).

1. Introduction

1.1. Background

The global energy landscape is experiencing a profound transformation driven by climate change, technological advancements and regulatory frameworks. As countries move toward decarbonization, energy markets have become more complex in balancing supply and demand efficiently. As energy markets become more complex and competitive, accurate forecasting is also indispensable for hedging future positions in forward markets. Long-term contracts and derivative instruments such as options and futures are based on the anticipated future prices of electricity. Incorrect forecasting can lead to poor hedging strategies and financial losses. By incorporating machine learning and statistical models like XGBoost or ARIMAX, companies can predict price patterns, improving market performance and decision-making in real-time. Energy price forecasting also help to minimize price volatility and protect consumers from extreme price hikes and thus can make energy affordable to everyone. While many researchers focus on primary drivers like coal, wind, or temperature to forecast electricity prices, however, this paper emphasizes on post-production variables such as- central power production, local power production, onshore-offshore wind generation, solar power generation, etc. These features offer a refined view of how electricity is used and managed within the market after generation. Traditional variables may predict supply potential but production variables reveal actual supply and grid behavior which ultimately determines pricing.

1.2. Research Objectives

The primary objective of this research is to evaluate and compare between ARIMAX and XGBoost model and finalize which model forecast better on out-of-sample data of electricity prices in Denmark's DK1 region.

1.3. Problem Statement

Forecasting electricity prices presents unique challenges, particularly in regions like Denmark (DK1) where renewable energy sources form a significant share of the energy mix. Unlike fossil fuel-based markets, where price trends are more stable and driven by primary factors like oil, coal, and gas, however, renewable energy introduces high levels of price volatility due to weather-dependent production. This inherent variability makes it difficult to predict future electricity prices accurately. The volatile and unpredictable nature of DK1 electricity prices necessitates the use of advanced forecasting models capable of handling complex patterns. Therefore, selecting the most suitable forecasting model is critical for ensuring precise predictions and effective energy trading strategies.

This paper follows the following questions-

Primary research question-

1. How do ARIMAX and XGBoost models perform in forecasting electricity prices in the DK1 zone, and which model offers superior accuracy under volatile market conditions?

Secondary research questions-

a. How effective is the ARIMAX model in incorporating exogenous variables to predict electricity prices and what are its limitations?
b. How does XGBoost's ability to handle nonlinear relationships and high-dimensional data contribute to its forecasting performance?
c. How do forecasting errors vary across different time horizons?

1.4. Research Method

Research method includes a step-by-step approach to ensure that the research process is structured, transparent, and capable of addressing the research questions effectively (Creswell & Creswell, 2017). For this thesis, the research method is designed to explore and compare the performance of ARIMAX and XGBoost models encompassing all the steps from data collection to reporting findings following a structured process.

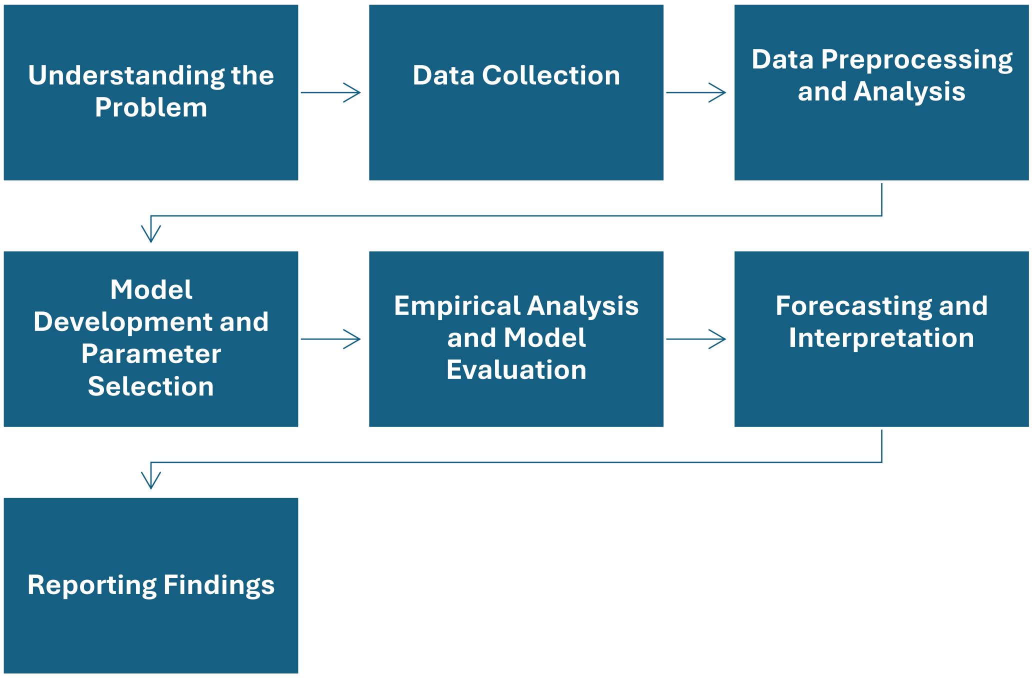


Figure 1: The research method used in the context of this thesis; adoptation from (Creswell & Creswell, 2017)

The research identifies the challenges of forecasting electricity prices in the DK1 zone particularly due to the high volatility caused by the reliance on renewable energy sources. The problem statement and research objectives are based on through literature review, research motivation toward energy sector and to meet the practical challenges of forecasting DK1 electricity prices to support trading and operations. Hourly electricity price data and relevant exogenous variables are collected from Energinet[1], the transmission system operator (TSO) in Danish energy market. The data are hourly because the electricity market operates and balances supply and demand on an hourly basis. This is a standard practice in energy markets worldwide as electricity cannot be easily stored and must be generated, distributed, and consumed in real-time.

Data preprocessing involves handling missing values, scaling features, and performing Exploratory Data Analysis (EDA) to understand patterns and trends. Statistical tests, including the Augmented Dickey-Fuller (ADF) test for stationarity and the Ljung-Box (LB) test for autocorrelation are conducted to validate the suitability of the data for time-series modeling.

[1] Energy Data Service, Build Report, Energinet, Link- https://www.energidataservice.dk/buildreport

After preprocessign of data both models are developed. The ARIMAX model is developed by using Auto-ARIMA and afterwards optimal parameters and exogenous variables are incorporated to improve the model's forecasting accuracy. The XGBoost model is developed using GridSearchCV to optimize hyperparameters such as tree depth, learning rate, and the number of estimators. The model's regularization features are utilized to address overfitting and improve generalization. The quantitative metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) are evaluated and lastly assessed the model with the Diebold-Mariano test to compare the performance of the two models on training and testing datasets. Finally compare the models visually and empirically in short-term (1-day and 2-day) and long-term (7-day) forecasting horizons. The results are visualized through time-series plots, residual distributions and error metrics. The findings then emphasize on discussing their implications for electricity price forecasting in volatile energy markets and providing recommendations for future research.

ChatGPT, as a generative AI tool, is utilized for generating some relevant coding and understanding of technical outputs but does not influence the core research findings, analysis, or interpretation.

2. Energy Market in Denmark

Denmark is one of the leading countries in renewable energy integration, particularly for wind power generation. In 2020, wind power accounted for over 50% of Denmark's electricity consumption, making it one of the highest shares of wind energy in the world (*Danish Energy Agency.* 2022). Denmark's commitment to sustainability and green energy is further demonstrated by its ambitious Energy Island project (an offshore hub for wind power generation) which aims to create artificial islands in the North Sea to harness offshore wind energy. This project would significantly expand country's renewable energy capacity and expecting to make it a major exporter of green electricity.

Figure 2: Electricity biding zone in Denmark (Source: Energinet)

Denmark has extensive grid interconnections with its neighboring countries including Norway, Sweden, Germany, Netherlands and UK. DK1's and DK2's connection with neighboring countries allow for efficient balancing of electricity across the regions. These interconnections allow for the export and import of electricity, helping to balance the grid during periods of excess or shortage in renewable energy generation. For instance, during times of high wind generation, Denmark can export excess electricity to its neighbors, conversely, while during periods of low wind, it can import hydropower from Norway or nuclear power from Sweden (Energinet, 2022).

The Transmission System Operator (TSO), Energinet, plays a central role in this balancing act, managing the grid and facilitating the trading of balancing services. Balancing markets including the mFRR (manual Frequency Restoration Reserves) and aFRR (automatic Frequency Restoration Reserves) are crucial in maintaining grid stability, particularly during periods of renewable generation fluctuation. As renewable energy becomes more prevalent, new market mechanisms like Power Purchase Agreements (PPAs) have emerged as vital tools for hedging against price volatility. In Denmark, PPAs allow energy producers and consumers to agree on long-term contracts, securing stable electricity prices and supporting the financing of new renewable energy projects. These agreements are crucial for integrating more renewable energy into the grid while ensuring that energy producers have a stable revenue stream. In addition to PPAs, Danish energy companies engage in hedging strategies to manage the risks associated with price fluctuations. Hedging can take the form of futures and options contracts. It allows market participants to lock in prices for future electricity deliveries. This is particularly important in a market (e.g Denmark), where renewable energy generation can cause significant price swings due to its variability. Electricity prices are influenced by the Electricity Price Area Differences (EPADs) which are used to hedge against regional price differences between Denmark's two bidding zones, DK1 and DK2 (Spodniak et al., 2021). These two zones- Western Denmark (DK1) and Eastern Denmark (DK2) are physically separated but connected to the broader Nordic Energy Market via the Nord Pool electricity exchange. The Danish electricity market is also part of the wider European energy system through the European Power Exchange (EPEX) where cross-border trading opportunities are maximized by the ACER (Agency for the Cooperation of Energy Regulators). However, REMIT (Regulation on Wholesale Energy Market Integrity and Transparency) is an European Union regulation aimed at promoting transparency and integrity in the wholesale energy markets, including electricity and gas by preventing market manipulation. National Regulatory Authorities (NRAs) e.g. Danish Utility Regulator (*Forsyningstilsynet* in Danish) in each country enforce REMIT locally and these national regulatory bodies collaborate with ACER for cross-border enforcement. These collboration and participation enhance market liquidity and ensures price convergence across Europe. Regulatory efforts by the Danish government and the European Union aim to ensure market transparency, efficiency and security of supply.

Nord Pool power exchange allows participants from the Nordic region, such as- Denmark, Norway, Sweden, and Finland, and extends to the Baltic countries. This market provides a platform for buying and selling electricity across the region. It operates both day-ahead and intraday markets allowing for cross-border electricity trading. This power exchange are also responsible for setting up spot price (system price) of different bidding zones e.g. DK1 based on electricity supply and demand. The system price of DK1 zone, at which electricity is delivered to the market participants in a Day-ahead (DA) market is also known as area prices or Elspot price. So, elspot price indicates market price of electricity determined for each hour of the following day. Participants submit their bids a day in advance and the elspot price is determined by matching supply and demand curves to establish a market-clearing price. This price can vary by bidding zone (e.g., DK1 and DK2 for Denmark) depending on transmission constraints and market dynamics. The transmission constraints here refer to the physical limitations (capacity limits) of the electricity grid in transporting power from one region to another due to congestion in the transmission network, grid maintenance, or infrastructure. When demand for electricity in a certain region exceeds the capacity of the transmission lines, it can create price differences between bidding zones. Transmission constraints can lead to localized price spikes or surpluses depending on the supply-demand balance in each zone. This scenario can be clarified with an example for determinig biding price for DK2 zone-

We can consider the interconnection transmission congestion with an example. Assuming that DK2 can generate 200 MW locally at €70/MWh but has a demand of 500 MW, requiring it to import electricity. DK1 generates 600 MW from wind power at €40/MWh and has sufficient power to meet its own demand. Germany has excess supply at €50/MWh and is connected to DK2.

- Transmission from DK1 to DK2 is limited to 200 MW.
- Transmission from Germany to DK2 is limited to 100 MW.

Scenario 1: Without Congestion- If DK2 only needed 100 MW, the wind power from DK1 would fully meet the demand at €40/MWh. The prices in both zones would equalize at €40/MWh, as there is no congestion.

Scenario 2: With Congestion- Now, imagine DK2 needs 300 MW. DK1 can only send 200 MW due to transmission constraints, so the remaining 100 MW must be supplied from Germany at €50/MWh. This causes DK2's price to rise since part of the demand is met by Germany's more expensive electricity.

Here, congestion causes a price difference between DK1 and DK2 due to the transmission limitations, resulting in higher costs for DK2. In this case, congestion causes price divergence between zones because transmission lines are limited in capacity.

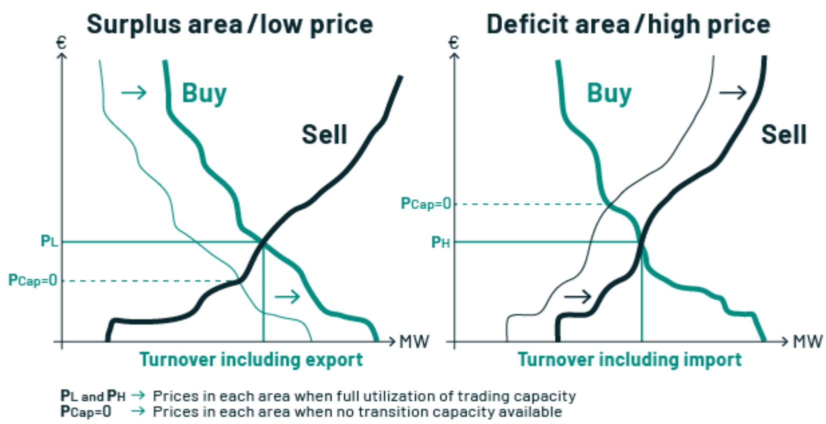


Figure 3: System price forecasting approach by Nord Pool

Market participants, including producers (such as wind farms, thermal plants) and consumers (large industries, distribution system operators) submit hourly bids for electricity delivery the next day (in the day-ahead market) or in real time (intraday). For instance, a wind farm might bid 100 MWh at €30/MWh for a particular hour based on its expected generation. Once all bids are collected, Nord Pool matches the supply and demand curves, which results in the market-clearing price (system price). This is the price at which electricity will be bought and sold for each hour of the next day. The system price reflects the equilibrium price for the entire Nordic market while area prices reflect congestion or transmission limitations between bidding zones. We can imagine the bidding process with an another hypothetical scenario.

Let’s imagine electricity needs to be supplied in the next day at 1:00 PM.

Producers' Offers: (electricity)	Consumers' Bids: (willing to pay)
Producer A: Offers to supply 100 MW at €50/MWh	Consumer X: Bids to buy 100 MW at €65/MWh
Producer B: Offers to supply 200 MW at €60/MWh	Consumer Y: Bids to buy 200 MW at €55/MWh
Producer C: Offers to supply 150 MW at €70/MWh	Consumer Z: Bids to buy 150 MW at €75/MWh

Market-Clearing Process:

Step 1: Nord Pool collects all offers from producers and bids from consumers.

Step 2: It matches the lowest offers from producers with the highest bids from consumers until supply meets demand.

Based on above scinario, consumer Z is willing to pay the most (€75/MWh), so they will definitely get their electricity and producer A is offering at the lowest price (€50/MWh), so they will definitely supply electricity. Nord Pool thus continues matching until supply meets demand.

Let's consider the market-clearing price is set at €60/MWh. So, Producers A and B will supply electricity because their offers are at or below €60/MWh. Consumers X and Z will buy electricity because they are willing to pay at or above €60/MWh. Contrary, producer C and consumer Y are excluded because their offers and bids don't match the market-clearing price.

However, the intraday market allows trading closer to real-time, enabling market participants to adjust their positions based on updated supply and demand information. This is especially important for renewable sources like wind, where generation forecasts can change frequently. Energy trading companies take advantage of market volatility by engaging in algorithmic trading to optimize their positions and profit from short-term price fluctuations.

3. Production-Consumption Dynamics in DK1

This chapter analyzes power generation trends over the past three years (2022 to 2024) in the DK1 zone focusing on various energy sources: solar, onshore wind, offshore wind, local production, and central production.

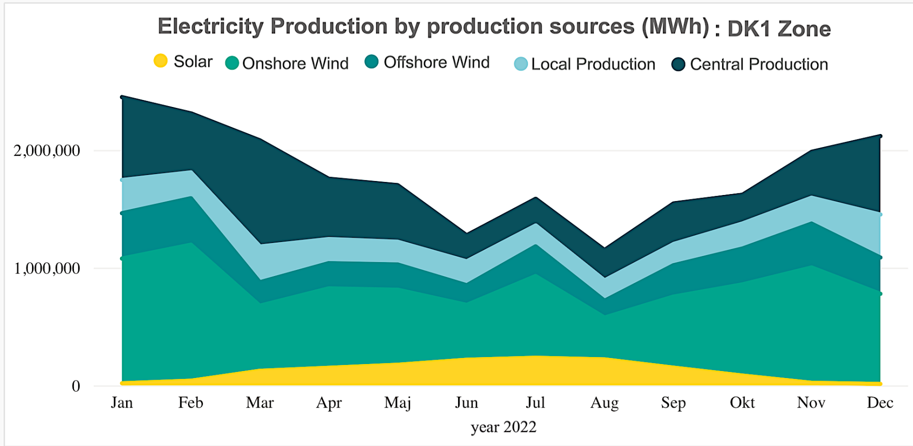


Figure 4: Power generation in DK1 zone in 2022 (source: Energinet)

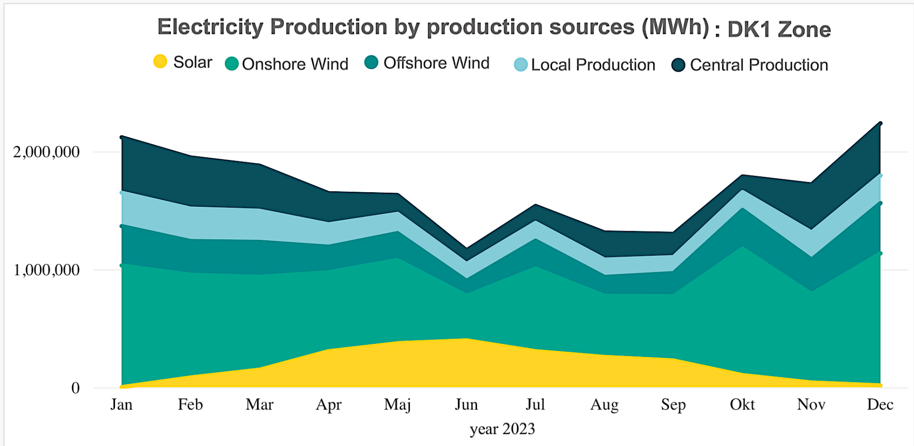


Figure 5: Power generation in DK1 zone in 2023 (source: Energinet)

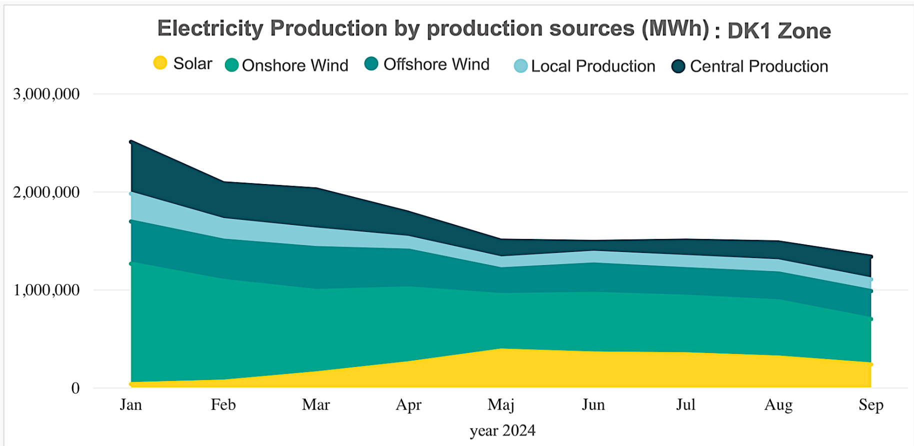


Figure 6: Power generation in DK1 zone in 2024 (source: Energinet)

Winter months typically has higher energy demand due to colder temperatures, longer nights and increased use of heating. The above three plots, figure 4 to 6, show a seasonal trend of electricity production where it reaches to the peak in the winter months and dips during the summer, particularly in June and July. In all years, wind energy, both onshore and offshore, dominates the energy mix while solar energy makes a noticeable increase in summer. The general trend highlights how renewable sources (wind and solar) shape the production patterns with local and central production acting as a stabilizing force across all months. In this context, local production refers to decentralized energy generation sources, often smaller in scale and typically situated closer to the point of consumption. These could be solar power systems, small-scale wind turbines, and smaller combined heat and power (CHP) plants, which often serve local communities (Wang, J. et al., 2017). On the other hand, central power production refers to larger, centralized power plants that generate electricity on a much larger scale. These include fossil fuels, biomass, or similar large power plants. These plants feed into the national or regional grids and provides a stable and large-scale electricity supply (Danish Energy Agency, 2021).

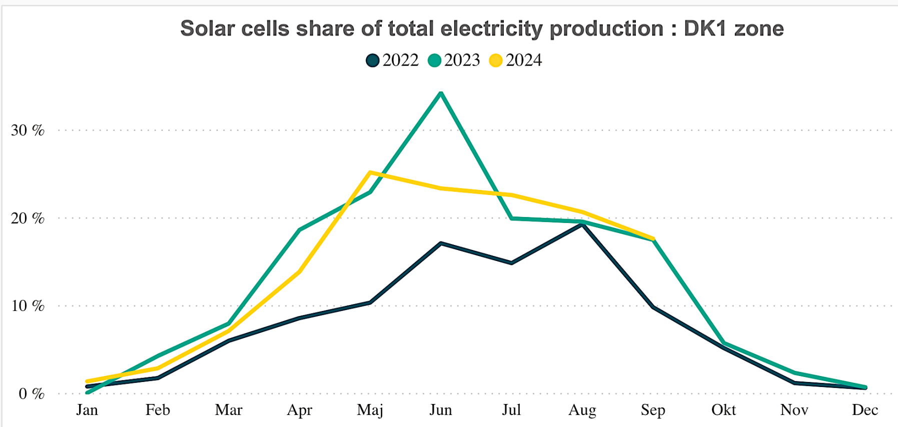


Figure 7: Solar cells power generation in DK1 zone in 2024 (source: Energinet)

The solar cell production share in DK1 (figure 7) shows notable seasonal patterns across 2022, 2023 and 2024. As depicted in the graph, solar energy production peaks during the summer months (May to August). If we pay special focus on the spike, we find the highest share of total production in june especially in 2023 where it surpassed 35% of total production. In contrast, solar contributions fall sharply in winter, nearing zero by December. However, 2022 experiences a smaller surge with a maximum production share reaching approximately 20% in June. This variability is directly tied to annual differences in weather patterns and perhaps variations in installed solar capacity.

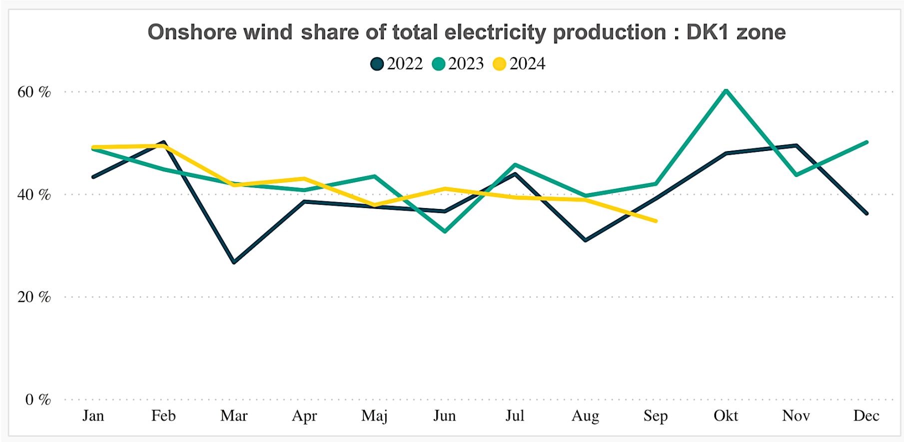


Figure 8: Onshore power generation in DK1 zone in 2024 (source: Energinet)

Onshore wind consistently contributes a significant portion of total electricity production in DK1, as shown in the above graph. Unlike solar, wind power exhibits a more even distribution throughout the year with a less pronounced seasonal dip. All years show relatively stable wind production with onshore wind accounting for 40-50% of the total production.

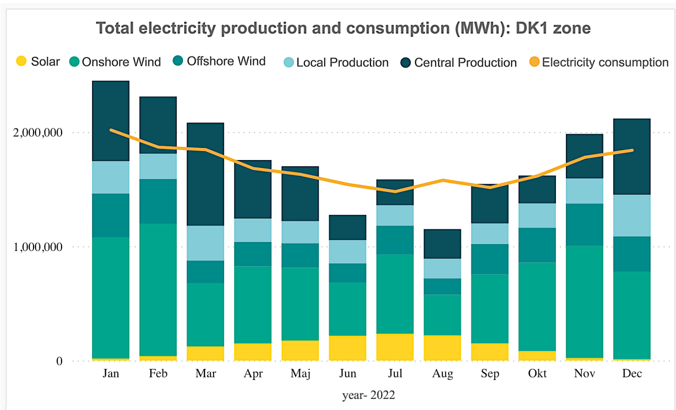


Figure 9: Total power generation and consumption in DK1 zone in 2022 (source: Energinet)

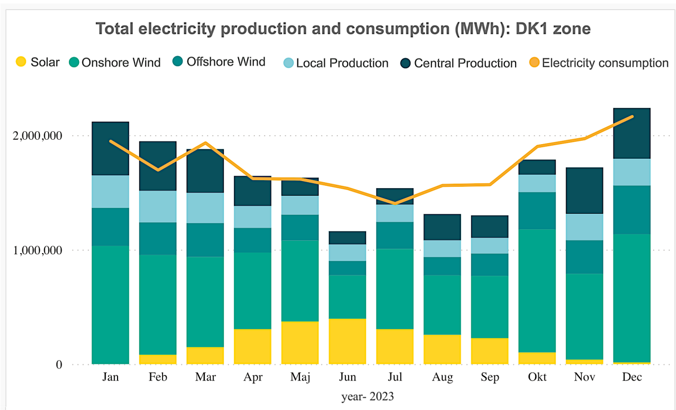


Figure 10: Total power generation and consumption in DK1 zone in 2023 (source: Energinet)

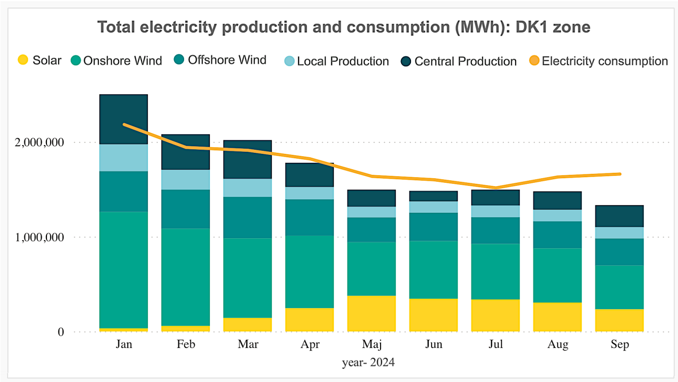


Figure 11: Total power generation and consumption in DK1 zone in 2024 (source: Energinet)

In each year, electricity consumption tends to follow a fairly predictable seasonal pattern with consumption. The gap between production and consumption narrows in summer months (e.g., June and July), where renewable sources are more abundant and the overall energy demand is lower. In 2022, the production appears slightly more variable, especially with a significant increase in January. In contrast, 2023 and 2024 show smoother production trends which indicates improved stability and perhaps more efficient grid integration of renewable energy. However, it is clearly depicted that production exceeds consumption over the past years. It suggests that DK1 may have surplus electricity at times, potentially allowing for energy exports or storage strategies, particularly when renewable power generation is high.

4. Literature Review

Forecasting electricity prices has been an ongoing challenge due to the inherent complexities of energy markets, especially with the increasing integration of renewable energy sources. The literature on this topic reveals a growing interest in both traditional statistical methods and advanced machine learning models. volume of research in electricity price forecasting is comparatively smaller than in sectors like finance, there has been a notable increase in studies as energy markets have gradually matured and evolved over recent decades.

Historically, traditional time-series models like ARIMA (Autoregressive Integrated Moving Average) have been applied to forecast energy prices. ARIMA's strength lies in its capacity to handle linear patterns in time series data, and it has been successfully used in electricity price forecasting in stable, fossil-fuel-based markets. For instance, (Contreras et al., 2003) used ARIMA models to forecast prices in the Spanish electricity market, demonstrating that the model could capture basic market dynamics. However, ARIMA and similar methods (ARMA, ARIMAX etc.) often fall short when handling the increasing volatility in markets influenced by renewable energy. Renewable energy, especially wind and solar power, introduces additional variability due to its dependence on weather conditions. This is particularly true for countries like Denmark, where renewables form a large portion of the energy mix. For example, wind power in Denmark's DK1 bidding zone has been shown to cause price fluctuations that traditional models struggle to predict accurately (Munoz et al., 2020). These models are primarily designed for stable markets driven by fossil fuels like coal and gas, where supply is more predictable. ARIMA models are particularly effective when dealing with time series data that are linear and stationary, where the relationships between past and future values are clear and stable over time (Hyndman, R. J., 2018). This is a major drawback in many real-world applications, such as energy and electricity price forecasting, where the relationships between variables may involve complex dynamics, including seasonality, volatility, and interactions between multiple factors (Tsay, 2005). Furthermore, ARIMAX is designed for multivariate time series, maximizing its use in cases where multivariate data (i.e., multiple predictors) are available, which could otherwise improve the accuracy of forecasts.

(Lucic & Xydis, 2023) focused on applying the ARIMAX (Autoregressive Integrated Moving Average with Exogenous Variables) model to forecast electricity prices in the Denmark-West (DK1) bidding zone. The study analyzed hourly price data for the DK1 intraday market over a two-year period from January 1, 2019, to December 31, 2020. The model was designed to use day-ahead prices as an exogenous input to improve forecasting accuracy for the intraday market. They found that the ARIMAX model significantly outperformed other forecasting techniques. The model could accurately predict volumeweighted average electricity prices up to 24 hours in advance. This performance suggests that ARIMAX, when combined with day-ahead prices as an external variable, can be a highly effective tool for short-term price forecasting. However, the study also highlighted certain limitations. While the ARIMAX model worked well with smaller datasets and specific external variables, its performance declines when applied to larger, more complex datasets. This suggests that ARIMAX may struggle to adapt to highly volatile or large-scale data, especially in comparison to more advanced machine learning models.

Recent studies have highlighted the limitations of traditional statistical models like ARIMAX in handling complex, nonlinear, and dynamic systems such as electricity price forecasting. (Fu et al., 2015) conducted a comparative study that evaluated ARIMAX against several machine learning approaches, including Support Vector Machine (SVM), Artificial Neural Networks (ANN), and Decision Tree models, for electricity load forecasting. Their results demonstrated that machine learning models, particularly A and ANN, consistently outperformed ARIMAX in terms of accuracy and robustness.

However another supervised ML approach, XGBoost, a gradient-boosting decision tree algorithm, has gained attention for its performance in this domain. It is particularly suitable for forecasting electricity prices in volatile markets because of its ability to process complex, multi-variable datasets efficiently.

The study, *"Balancing the Norwegian regulated power market anno 2016 to 2022"* provides an in-depth analysis of Norway's power market dynamics over recent years. The study offers a detailed statistical assessment of the Norwegian regulated power market, examining trends across various market metrics over six years. This analysis provides insights into changes in balancing market behaviors as renewable energy sources, especially wind power, have grown in significance. The study uses several models, including XGBoost, to assess predictability but notes that the effectiveness of these models is limited by the increasing complexity of market dynamics (Austnes et al., 2024).

(Xu et al., 2024) conducted a comparative study aimed at enhancing long-term solar energy hourly forecasting through the fusion of GRU (Gated Recurrent Units) and XGBoost models. Their work introduced an innovative hybrid approach where XGBoost played a crucial role in capturing nonlinear relationships among features, complemented by GRU's temporal sequence processing capabilities. (Xu et al., 2024) mentioned that XGBoost's ability to handle high-dimensional data and integrate exogenous variables contributed significantly to the hybrid model's success. This finding aligns with the broader recognition of XGBoost as a robust and scalable machine learning model, especially in scenarios requiring high adaptability and precision. By demonstrating consistent performance across varying forecasting lengths, this work underscores the flexibility and robustness of XGBoost in handling complex energy forecasting problems.

In addition to machine learning approaches, recent studies have also focused on feature selection as a critical component of electricity price forecasting models. For instance, (Lago et al., 2021) demonstrated that incorporating explanatory variables such as weather data, historical prices, and system load significantly improves the accuracy of machine learning models. This is particularly relevant for Denmark, where wind generation and demand patterns can change rapidly based on weather conditions.

From reviewing the existing literature, it's clear that while many studies have focused on using statistical models like ARIMAX or machine learning models like XGBoost for forecasting, direct comparisons between the two in the context of electricity price prediction are quite limited. Most research either works on improving one model or compares machine learning approaches without including traditional time series models like ARIMAX. This creates a gap in understanding how these two models perform against each other in real-world energy forecasting scenarios, especially in volatile markets like DK1. This research aims to fill that gap and provide valuable insights into their comparative strengths and weaknesses.

5. Research Philosophy

This chapter outlines the research philosophy and methodological structure underpinning this study. It follows a framework that guides the scientific approach to understanding research paradigms, data gathering techniques, and analysis methods essential for conducting and presenting research effectively. The methodology framework is presented and described step-by-step aligned with the structured approach recommended by (D O'Gorman & MacIntosh, 2015).

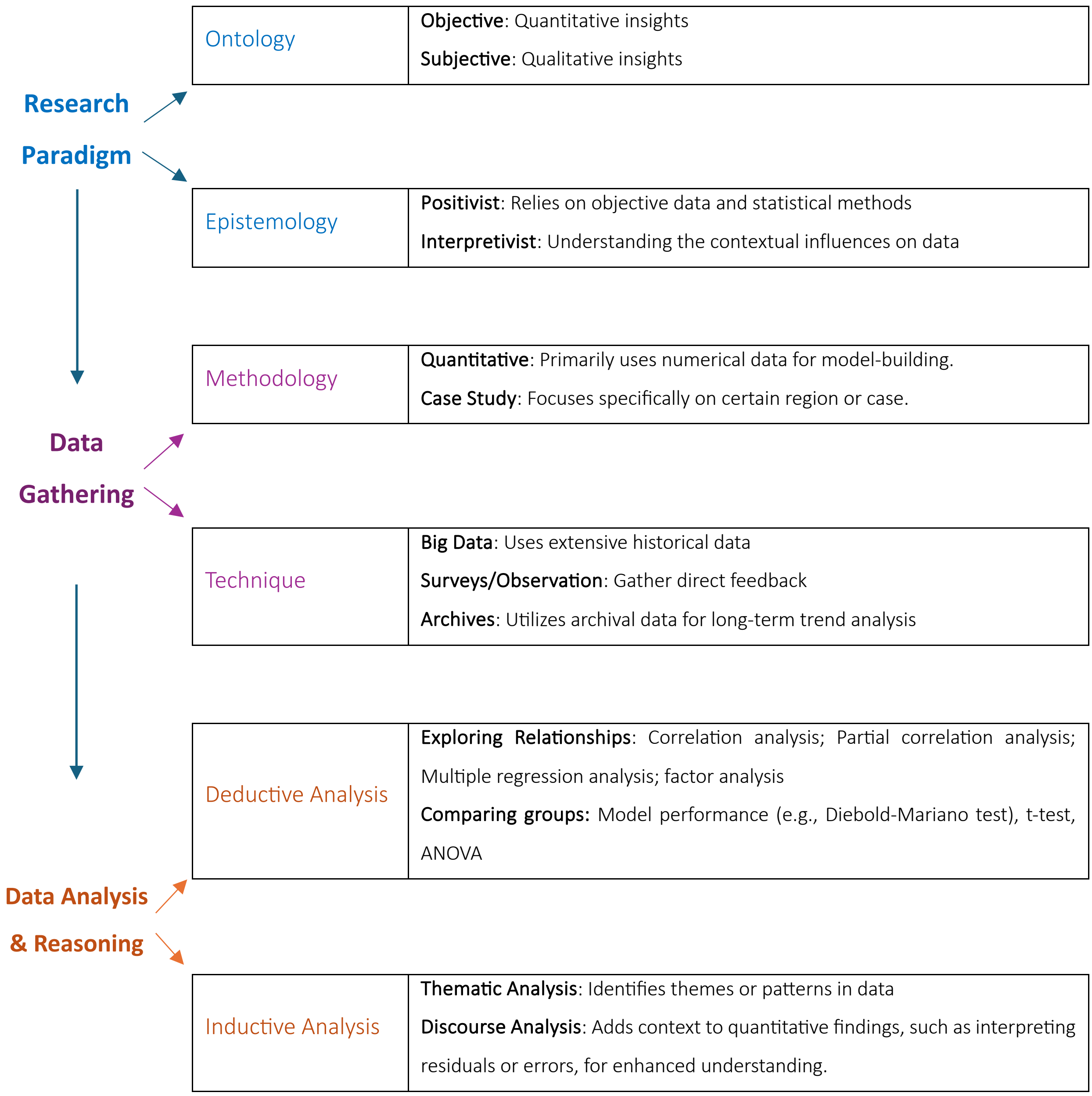


Figure 12: Research Methods Map

Source: Own creation with inspiration from (D O'Gorman & MacIntosh, 2015)

5.1. Research Paradigm

A research paradigm establishes the foundational beliefs about reality and knowledge that guide the research design (D O'Gorman & MacIntosh, 2015). This study considers both **Ontology** and **Epistemology** to clarify the perspectives applied to model development and electricity price forecasting.

5.1.1. Ontology

Ontology concerns the nature of reality. It illustrates whether the world is seen as objective or subjective. An objective perspective views reality as consisting of solid objects that can be measured and tested which exists independently on our perception or experience. In contrast, a subjective perspective sees reality as shaped by the perceptions and interactions of living subjects (D O'Gorman & MacIntosh, 2015). In this study, the ontology is largely objective, grounded in the assumption that electricity prices and energy market behaviors can be analyzed and forecasted through empirical, quantifiable data. However, there may also be subjective elements if qualitative insights or expert perspectives on energy trends are incorporated.

5.1.Epistemology

Epistemology addresses how valid knowledge is obtained.(D O'Gorman & MacIntosh, 2015). This research is positivist in its epistemology, assuming that knowledge about the electricity market can be derived from objective data and statistical methods. A positivist approach aligns well with quantitative models, like ARIMAX and XGBoost, used for forecasting. However, there may be a slight interpretivist element if qualitative data, like geopolitical impacts, demand-supply mismatch, volatility and negative pricing impact are considered to contextualize results.

5.2. Data Gathering

The methodology for data gathering is structured into Quantitative, Qualitative, and Case Study methods, though this study primarily uses quantitative methods with potential elements of case study analysis to contextualize findings.

Methodology

The primary methodology is quantitative, focusing on numerical data from electricity prices and energy production variables of Denmark’s DK1 bidding zone.

Techniques

The techniques used for data collection are rooted in big data, survey or observation.

This study leverages a large dataset comprising 40,926 observations of hourly electricity prices and production capacities influencing the energy market. The target variable is 'dk1' (representing spot electricity prices in the DK1 bidding zone), and there are 15 exogenous features. These relevant exogenous variables include production data, such as central production, local production, and production from different energy sources—specifically onshore and offshore wind, hydropower, and solar power generation, all are in megawatts (MW). This comprehensive dataset allows for a detailed analysis of electricity price dynamics and influencing factors across Denmark's energy landscape.

Though no surveys or observational techniques directly applied here but it could support understanding how market participants react to price changes. So, this could be suggested for future research to understand market participant behavior.

The data used in developing the model are-

Features (DK1 Zone)	Full Form	Description
dk1	DK1 Electricity Spot Price	The hourly elspot price of electricity in the DK1 zone, measured in Euro/MWh.
LocalPowerMWhDK1	Local Power Production	Total electricity generated by local power plants (MWh).
LocalPowerSelfConMWhDK1	Local Power Self-Consumption	Electricity produced and consumed locally without entering the grid (MWh).
CentralPowerMWhDK1	Central Power Production	Electricity generated by large-scale, centralized power plants in DK1 (MWh).
CommercialPowerMWhDK1	Commercial Power Production	Power produced for commercial purposes, often from large private operators (MWh).
HydroPowerMWhDK1	Hydropower Production	Electricity generated from hydroelectric plants in the DK1 zone (MWh).
OffshoreWindGe100MW_MWhDK1	Offshore Wind (≥100 MW)	Electricity generated by offshore wind farms with a capacity of 100 MW or more (MWh).
OffshoreWindLt100MW_MWhDK1	Offshore Wind (<100 MW)	Electricity generated by offshore wind farms with a capacity below 100 MW (MWh).
OnshoreWindGe50kW_MWhDK1	Onshore Wind (≥50 kW)	Electricity generated by onshore wind farms with a capacity of 50 kW or more (MWh).

OnshoreWindLt50kW_MWhDK1	Onshore Wind (<50 kW)	Electricity generated by small-scale onshore wind farms with less than 50 kW (MWh).
SolarPowerGe10Lt40kW_MWhDK1	Solar Power (10–40 kW)	Solar electricity generated from systems with a capacity between 10 kW and 40 kW (MWh).
SolarPowerGe40kW_MWhDK1	Solar Power (>40 kW)	Solar electricity generated from systems with a capacity greater than 40 kW (MWh).
SolarPowerLt10kW_MWhDK1	Solar Power (<10 kW)	Electricity from small-scale residential or small business solar systems (<10 kW) (MWh).
SolarPowerSelfConMWhDK1	Solar Power Self-Consumption	Solar power generated and consumed on-site without exporting to the grid (MWh).
PowerToHeatMWhDK1	Power-to-Heat Conversion	Electricity converted to heat, often for district heating systems in DK1 (MWh).
GrossConsumptionMWhDK1	Gross Electricity Consumption	Total electricity consumed, including losses and self-consumption (MWh).

Table 1: Features used for the modeling
(Description from Energinet)

5.3. Data Analysis

Python programming language is used as the primary tool for data analysis in this thesis . Key libraries such as *Pandas* and *NumPy* were utilized for data manipulation and preprocessing, while *Matplotlib* and *Seaborn* enabled effective data visualization. Model development and evaluation were conducted using Statsmodels for statistical modeling (ARIMAX) and XGBoost for gradient boosting models.

5.4. Research Reasoning

The analysis strategy of this report includes both **deductive** and **inductive** approaches which enables a comprehensive examination of the forecasting models' effectiveness and interpretation of the results.

Deductive Approch

A deductive approach is employed to validate existing theories and methods. This approach begins with a general theory or hypothesis with specific predictions. This reasoning starts with established theories and hypotheses which are then tested against real-world data. It focuses on testing predefined hypotheses whether there is significant differences between two models and uses quantitative analysis methods to verify or reject them.

In this study, several deductive analytical methods were applied:

Feature Analysis: To examine the relationships between electricity prices (target variable, dk1) and exogenous variables.

Regression Analysis: To integrate within the XGBoost algorithm, this approach is used to assess the influence of multiple variables on electricity price predictions.

Residuals Analysis: To compare the performance of both forecasting models ARIMAX and XGBoost across different forecasting horizons, such as short-term (1-day and 2-day) versus long-term (7-day).

Time-Based Error Analysis: To analyze hourly and daily MAE and RMSE trends to identify model strengths and weaknesses over specific times

Diebold-Mariano Test: Tested if XGBoost's performance is significantly better than ARIMAX's

Inductive Approach

Inductive reasoning involves observing data and deriving generalized insights. Under this approach, data is used to generate a theory or conceptual framework. Researchers collect data and observations and then build a general principle or theory based on those observations. Inductive reasoning is particularly useful when little existing literature or theoretical frameworks exist on a topic (Saunders, 2009). This study mostly concentrated on deductive approach rather than inductive approach.

6. Model Specification

This section focuses on pre-estimation testing for stationarity, autocorrelation, normality and finally the implementation of time series forecasting models ARIMAX and XGBoost to predict electricity prices in the DK1 zone. This section explores the setup, tuning, and application of these models as well as their evaluation metrics.

6.1. Pre-Estimation Testing for Model Suitability

Before implementing the forecasting models, it is essential to evaluate the characteristics of the dataset to ensure the validity of the chosen methods. This chapter outlines three key statistical tests: the Augmented Dickey-Fuller (ADF) test for stationarity, the Ljung-Box (LB) test for autocorrelation, and the Jarque-Bera (JB) test for normality. These tests are vital for assessing the assumptions underpinning time series models.

6.1.1. Augmented Dickey-Fuller (ADF) Test for Stationarity

The Augmented Dickey-Fuller (ADF) test determines whether a time series is stationary. It is a crucial requirement for many time series models like ARIMA. The test evaluates the null hypothesis (H_0) that the series contains a unit root, implying non-stationarity. The mathematical form of the ADF test in its most general form can be written as:

$$\Delta Y_t = \alpha + \beta_t + \gamma Y_{t-1} + \Sigma \delta_i \Delta Y_{t-1} + \varepsilon_t \quad (1)$$

Where, $\mathrm{Y_t}$ is the time series, $\Delta\mathrm{Y_t}$ is the first difference of the time series, α is the intercept, t is the time trend, α, β, γ and δ are parameters to be estimated, and ϵ_t is the error term. If γ is significantly less than zero, the series is deemed stationary.

The null hypothesis of the test is that $\gamma = 0$, indicating the presence of a unit root, i.e. the series is non-stationary. The alternative hypothesis is that $\gamma < 0$, indicating the series is stationary. If the p-value of the test statics is close to 0 (typically $p < 0.05$), it indicates that the time series stationary.

Non-stationarity in the data can lead to misleading and untrustworthy model results, and in such cases, difference or other transformations may be employed to achieve stationarity (Dickey & Fuller, 1979).

6.1.2. Ljung-Box (LB) Test for Autocorrelation

The Ljung-Box test checks for the presence of autocorrelation within a time series, which can indicate dependencies across observations. It tests the null hypothesis that refers the time series lags are independent and uncorrelated (close to zero).

The Ljung–Box test is built on the following hypotheses:

H_0: $The\ data\ are\ independently\ distributed\ (No\ autocorrelation\ exist)$.
H_1: $The\ data\ are\ not\ independently\ distributed\ (serial\ correlation\ or\ dependency\ in\ the\ lags)$.

The test statistic of Ljung-Box test is:

$$Q = n(n+2)\sum_{k=1}^{m}\frac{\widehat{\rho_k}^2}{n-k} \quad (2)$$

Where, Q is the Ljung-Box test statistic, n is the number of observations, $\widehat{\rho_k}$ is the sample autocorrelation at lag k and m is the number of lags being tested.

If the p-value of the Ljung-Box test is low (typically $p < 0.05$), it indicates that there is significant autocorrelation in the lags, implying that the there in some pattern in the time series that can be capture by the ARIMA model and vice versa (Ljung & Box, 1978). Same way, we can check the residuals of the model to diagnosis the model. If p-value of the test statics for residuals is lower than 0.05 ($p < 0.05$), it refers there is autocorrelation in the residuals, implying that there is still some pattern which is not capture by the model. Hence, the ARIMA model needs to be refined.

6.1.3. Jarque-Bera (JB) Test for Normality

Although XGBoost models are flexible to handle non-normal distributions, a normality test would still be conducted as it helps in determining the distribution of the data and inform any necessary adjustments to the model. The JB test is built on the following hypotheses:

H_0: $Skewness\ and\ Kurtosis\ match\ a\ normal\ distribution.$
H_1: $Skewness\ and\ Kurtosis\ does\ not\ match\ a\ normal\ distribution.$

The JB test statistic is:

$$JB = n\left(\frac{S^2}{6} + \frac{(K-3)^2}{24}\right) \quad (3)$$

Where n is the number of observations in the data set, S is the sample skewness, and K is the sample kurtosis.

If the JB statistic is significantly different from 0, then the null hypothesis is rejected, indicating that the data do not have a normal distribution. Other way, if p-value of the test statics is lower than 0.05 ($p < 0.05$), then it rejects null hypothesis and refers the data does not normally distributed. We can also diagnosis model based on JB test. Normality in residuals is essential for ensuring unbiased predictions in time series models (Jarque & Bera, 1987).

6.2. ARIMAX Model

ARIMA (AutoRegressive Integrated Moving Average) is a univariate model which forecast future values based solely on past values of the target variable whereas ARIMAX (AutoRegressive Integrated Moving Average with Exogenous Variables) extends ARIMA by incorporating external (exogenous) variables into the forecasting process. This inclusion of additional explanatory variables on the target variable makes ARIMAX suitable for multivariate time series analysis (Pankratz, 2012).

The ARIMAX model is estimated based on three key components AutoRegressive(AR), Integrated (I) Moving Average (MA) and Exogenous(X) . Generically the model is denoted as model (p, d, q) where (p) is the number of autoregressive term in the model, q is the number of moving average terms and d is the number of differences required to make the time series stationary (this is the "Integrated" part). So an autoregressive (AR) model of order p, abbreviated as AR(p) refers to using past values to predict the current value of the time series. In an AR model of order p, the current value $\mathrm{x_t}$ is expressed as a linear combination of the past p values of the series. So, it an be explained as a function of p past values such as $\mathrm{x_{t-1}, x_{t-2} \dots x_{t-p}}$, where p determines the number of steps into the past needed to forecast the current value.

An autoregressive model can be written as (Shumway et al., 2000)

$$x_t = \phi_1\, x_{t-1} + \phi_2\, x_{t-2} + \cdots + \phi_p\, x_{t-p} + \varepsilon_t \quad ; \quad \varepsilon_t \sim \text{WN}\,(0, \sigma^2) \tag{4}$$

where x_t is the current value of which is assumed stationary, ε_t is the error term or residual, which is the white noise process (assumed to be normally distributedthe) and $\phi_1, \phi_2, \ldots, \phi_p$ are the parameters (or model coefficients) for each lag $(\phi_p \neq 0)$. The mean of x_t in (4) is zero. If the mean, μ, of x_t is non-zero, we can replace x_t by $x_t - \mu$ in (4). So, with a non-zero mean, we can rewrite the equation as:

$$x_t = \phi_1\,(x_{t-1} - \mu) + \phi_2\,(x_{t-2} - \mu) + \cdots + \phi_p\,(x_{t-p} - \mu) + \varepsilon_t \tag{5}$$

Or,

$$x_t = \alpha_0 + \phi_1\, x_{t-1} + \phi_2\, x_{t-2} + \cdots + \phi_p\, x_{t-p} + \varepsilon_t \tag{6}$$

Where, $\alpha = \mu(1 - \phi_1 - \phi_2 - \ldots - \phi_p)$

So, autoregresive first-order model, AR(1), could be written as

$$x_t = \alpha_0 + \phi_1\, x_{t-1} + \varepsilon_t \tag{7}$$

And second order-order mode, AR(2)-

$$x_t = \alpha_0 + \phi_1\, x_{t-1} + \phi_2\, x_{t-2} + \cdots + \phi_p\, x_{t-p} + \varepsilon_t \tag{8}$$

So, The AR order (p) represents how many previous time points (lags) we can use to predict the current value. p can be initially determined by looking at the Partial Autocorrelation Function (PACF) plot. The PACF plot helps to find the point (lag) after which the correlation sharply decreases (crossing the confidence interval).

The "Integrated (I)" part of the model refers to making the time series stationary by differencing. A series is stationary if its statistical properties (like mean and variance) do not change over time (Shumway et al., 2000). If the data has trends (upward or downward), differencing can remove the trend and stabilize the series. The number of differences needed to make the data stationary is generally represented by d. If d=1, the first-order differenced series would be:

$$y_t = x_t - x_{t-1} \tag{9}$$

Where y_t is the differenced series and x_t is the original series.

The moving average (MA) model is considered as an alternative to the autoregressive representation (Shumway et al., 2000). The MA part of ARIMA involves modeling the current value based on past errors or "shocks" in the system. It means that instead of using past values (e.g. x_{t-1}) of the series directly, the current value (x_t) is then considered as a linear combination of past forecasted errors(e.g. ε_{t-1}). Moving average model of order q, abbreviated as MA(q). A moving average model can be written as-

$$x_t = \varepsilon_t + \theta_1 \, \varepsilon_{t-1} + \theta_2 \, \varepsilon_{t-2} \cdots + \theta_t \, \varepsilon_{t-p} \quad ; \; \varepsilon_t \sim \; WN\,(0, \sigma^2) \tag{10}$$

Where, x_t is the current value, ϵ_t is the current error term (residual), $\varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-p}$ are the past error terms, $\theta_1, \theta_2, \dots, \theta_q \; (\theta_p \neq 0)$ are the MA parameters.

So, The MA order (q) represents how many past errors are considered in the model. Here, q is determined by looking at the Autocorrelation Function (ACF) plot. The ACF plot shows how current values of the series are correlated with past values. The value of q is the point (lag) after which the autocorrelation drops off (crosses the confidence interval).

The exogenous component incorporates external $m_{k,t}$ variables that are assumed to have a causal relationship with the target variable (x_t). The exogenous eqution is-

$$x_t = \beta_1 m_{1,t} + \beta_2 m_{2,t} + \cdots + \beta_k m_{k,t} \tag{11}$$

Where, $m_{k,t}$ is the exogenous variables at time t and β_k is the coefficients for the exogenous variables.

Ater combining all the components (p,d,q and x), the ARIMAX model can be written as-

$$x_t = \alpha_0 + \phi_1 \, x_{t-1} + \cdots + \phi_t \, x_{t-p} + \; \beta_1 m_{1,t} + \beta_2 m_{2,t} + \cdots + \beta_k m_{k,t} + \theta_1 \, \varepsilon_{t-1} + \cdots + \theta_t \, \varepsilon_{t-p} + \varepsilon_t \tag{12}$$

where ϕ, θ and β are the AR, MA and exogenous parameters, x_{t-p} donotes past values, $m_{k,t}$ is the exogenous variables and ε_{t-p} denotes past forecasting errors, ε_t is the current error which is white noise process, $\varepsilon_t \sim \; WN\,(0, \sigma^2)$. Residuals are considered white noise if it is completely random with zero mean, constant variance (homoskedasticity, σ^2) and no autocorrelation to each others (independence over time). White noise is the weaker assumption of Random Walk model which refers there is some pattern in the time series that could be captured by the time series model like ARIMA. The integrated part d is handled by differencing the data before applying the AR and MA components.

While the X (exogenous) part is linear, the overall ARIMAX model can capture non-linear relationships in the data due to the interaction of all components (AR, MA, and I) (Pankratz, 2012).

To select the best ARIMAX model, we often use statistical metrics such as the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). These criteria are used to balance the complexity of the model with the goodness of fit. The AIC is calculated as (Akaike, 1974)-

$$\mathrm{AIC} = -2\ln(\mathrm{L}) + 2\mathrm{k} \tag{13}$$

Where, L is the likelihood of the model and k is the number of parameters in the model. The model with the lowest AIC is typically considered the best.

Similarly, the BIC penalize the complexity of the model more severely (Schwarz, 1978) such as –

$$\mathrm{BIC} = -2\ln(\mathrm{L}) + \mathrm{k}\ln(\mathrm{n}) \tag{14}$$

Where n is the number of observations. Lower BIC values indicate a better model fit while avoiding overfitting.

Auto ARIMA is an automated approach (for statistics software) to identify the optimal parameters for the ARIMA (AutoRegressive Integrated Moving Average) model. Instead of manually selecting the parameters (p,d,q), Auto ARIMA uses statistical techniques and search algorithms to determine the most suitable combination that minimizes forecast error. Statistical software like R or Python are commonly used to get most appropriate parameter for ARIMA model. In Python, the "*pmdarima"* library is commonly used to calculate Auto ARIMA models. This library implements the Auto ARIMA algorithm which automatically determines the parameters by minimizing (penalize) metrics like Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) (Hyndman, Rob J. & Khandakar, 2008).

Once the initial values for p, d, q and x are selected, the ARIMAX model is fit to the data. After estimating ARIMAX model, we should conduct diagonostic test for the model with the residuals (Box, George EP et al., 2015). The popular residual diagnostics approaches are checking ACF of residuals, Ljung-Box test and normality tests. These are conducted to ensure that the residuals resemble white noise which indicates that the estimated ARIMAX is a well-fitted model.

6.3. XGBoost Model

XGBoost (Extreme Gradient Boosting) is an efficient and powerful implementation of gradient-boosted decision trees designed for speed and performance. It was first proposed by Tianqi Chen and Carlos Guestrin in 2011 and has been continuously optimized and improved in the follow-up study of many scientists (Chen, Tianqi & Guestrin, 2016). It combines an ensemble with decision trees to build a strong predictive model by iteratively correcting the errors of the previous models (Friedman, 2001). The model uses techniques like regularization, parallelization, and tree pruning to enhance accuracy and prevent overfitting, making it highly effective for both classification and regression tasks (Chen, T., 2015). Due to its scalability and robustness, XGBoost has been widely adopted in machine learning competitions and real-world applications (Chen, Tianqi & Guestrin, 2016).

Traditional Boosting Tree models rely solely on first-order derivative information when constructing trees. When training the n-th tree, the model adjusts based on the residuals (errors) from the previous $n-1$ trees. However, this sequential dependency on residuals can make distributed training difficult because each tree's training depends on the outcomes of its predecessors (Chen, Tianqi & Guestrin, 2016). XGBoost addresses this issue by performing a second-order expansion on the loss function. This means that XGBoost takes into account both the *first-order (gradient)* and *second-order (Hessian)* information. The incorporation of second-order derivatives allows the model to make more precise updates, leading to better optimization. Additionally, it enables parallel computing, leveraging multithreading to significantly improve training speed. XGBoost automatically distributes tasks across multiple cores in the CPU, which makes the training process faster and more scalable. The XGBoost algorithm is briefly introduced as follows (Chen, Tianqi & Guestrin, 2016; Friedman, 2001) –

6.3.1. Objective Function

The objective function in XGBoost balances model complexity and performance. It consists of two components: **the loss function** L and **the regularization term** Ω.

$$Obj = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (15)$$

Where, $L(y_i, \hat{y}_i)$ is the loss function (expressing as MSE) measuring how well the model predicts the target y_i. $\Omega(f_k)$ is the regularization term to prevent overfitting (Chen, Tianqi & Guestrin, 2016).

6.3.2. Loss Function

For regression problems, the loss function (L) is typically mean squared error (MSE):

$$L(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{16}$$

This measures the squared difference between predicted and actual values (Friedman, 2001).

6.3.3. Regularization Term

The regularization term $\Omega(f_k)$ penalizes model complexity. For decision trees, it includes both the number of leaves and the magnitude of the leaf weights:

$$\Omega(f_k) = \gamma\mathrm{T} + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \tag{17}$$

Where, T is the number of leaves, w_j represents the leaf weights, γ and λ are hyperparameters controlling the regularization strength (Chen, Tianqi & Guestrin, 2016).

6.3.4. Gradient and Hessian Approximation

In XGBoost, second-order Taylor expansion is used to approximate the objective function. Gradients and Hessians are computed to optimize the loss function.

$$g_i = \frac{\partial L(y_i, \hat{y}_i)}{\partial \hat{y}_i}, \qquad h_i = \frac{\partial^2 L(y_i, \hat{y}_i)}{\partial {\hat{y}_i}^2} \tag{18}$$

Where, g_i is the gradient of the loss function and h_i is the second-order derivative (Hessian)

6.3.5. Tree Structure Scoring

The optimal structure of a decision tree is determined by calculating the following score for each split:

$$Gain = \frac{1}{2}\left(\frac{\sum_{i\in I_L} g_i}{\sum_{i\in I_L} h_i + \lambda} + \frac{\sum_{i\in I_R} g_i}{\sum_{i\in I_R} h_i + \lambda} - \frac{\sum_{i\in I} g_i}{\sum_{i\in I} h_i + \lambda}\right) - \gamma \tag{19}$$

Where, I_L and I_R represent the left and right partitions of the split; g_i and h_i are the gradient and Hessian, as previously defined; γ and λ are regularization parameters.

Once the tree is built, the prediction for a new instance is made by summing the contributions of each tree-

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \qquad (20)$$

Where, $f_k(x_i)$ is the prediction from the k-th tree for instance x_i; K is the number of trees in the model (Chen, Tianqi & Guestrin, 2016).

A fundamental structure of XGBoost involves feeding the residuals (errors) from one tree into the subsequent tree.

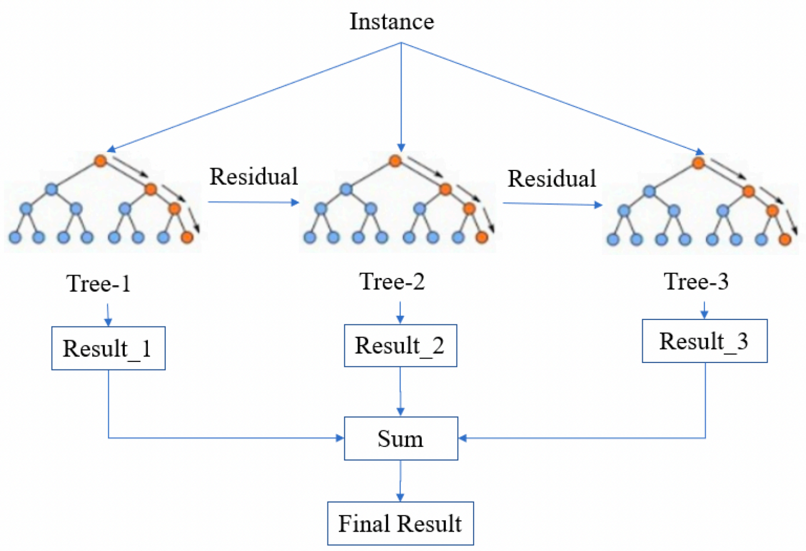


Figure 12: Simplified structure of XGBoost (Wang, W. et al., 2020).

For example, the residuals from tree-1 are passed into tree-2, and the process continues iteratively with each tree working to minimize the residual errors from the previous one. This approach gradually improves the model's performance by reducing prediction errors across successive iterations.

While XGBoost is a machine learning model and it does not have any formal diagnostic tests like ARIMA models. So, Residial analysis and normality test can help to evaluate and interpret XGBoost models.

6.4. Models Validation and Evaluation Approach

This section explains different comparative matrics to evaluate the models.

6.4.1. Diebold-Mariano Test

The Diebold-Mariano test is used to compare the predictive accuracy of two competing models. It assesses whether the difference in forecasting errors between two models is statistically significant (Diebold & Mariano, 2002).

The test statistic is given by:

$$DM = \frac{\bar{d}}{\sqrt{\left[\left(\frac{1}{T}\right)\gamma_0 + \left(\frac{2}{T}\right)\sum_{k=1}^{h-1}\gamma_k\right]}} \tag{21}$$

Where:

$\bar{d}$: Mean of the loss differential, $d_t = g(e_{1,t}) - g(e_{2,t})$, where $g(e)$ is the loss function (e.g., squared error).

T: Number of observations (forecast horizons).

γ_0: Variance of the loss differential d_t

γ_k: Covariance of d_t and d_{t-k}, accounting for autocorrelation in errors.

h: Forecast horizon.

The difference between the forecast errors of two models is expressed as:

Loss differential, $d_t = g(e_{1,t}) - g(e_{2,t})$

Where, $e_{1,t}$ forecast error from model 1 $(y_t - \hat{y}_{1,t})$and $e_{2,t}$ forecast error from model 2 $(y_t - \hat{y}_{2,t})$

Given hypothesis for the test is-

H_0: *Both models have equal predictive accuracy when test statistics is* 0.

H_1: *The predictive accuracy of the two models differs*.

If the DM statistic falls outside the critical values of the normal distribution for a chosen significance level (e.g., α = 0.05), the null hypothesis is rejected, indicating a significant difference in predictive accuracy. Under the null hypothesis, the DM statistic follows a standard normal distribution $DM \sim N(0, 1)$ (Diebold & Mariano, 2002).

6.4.2. Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is a measure of the average magnitude of errors between predicted values and actual observed values. It calculates the average of the absolute differences between the predicted and actual values, ignoring the direction of the error (whether positive or negative). MAE is used to measure the average magnitude of forecast errors. It is easy to interpret because it gives the average amount by which the predictions are off in absolute terms. MAE is particularly useful when the magnitude of the error is more important than the direction of the error. A lower MAE values indicate better model accuracy since the prediction error is smaller on average (Willmott & Matsuura, 2005).

$$MAE = \frac{1}{n}\sum_{t=1}^{n} | x_t - \hat{x_t}^2 | \qquad (22)$$

Where, n is the number of observations, x_t is the actual value at time t, $\hat{x_t}$ is the predicted value at time t and | this operator gives the absolute value of a number.

6.4.3. Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is another common measure of forecast error that penalizes larger errors more than smaller ones. It is commonly used when it is important to heavily penalize large errors. It tends to amplify the influence of outliers (large deviations between the predicted and actual values). As such, it is a useful metric when errors of large magnitude are undesirable. So, RMSE is more sensitive to outliers than MAE because it squares the errors before averaging. For example, if large deviations (outliers) are present, RMSE will increase more than MAE. Likely to MAE, a lower RMSE values suggest better predictive performance(Chai & Draxler, 2014).

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} (x_t - \hat{x_t})^2} \qquad (23)$$

Where, n is the number of observations, x_t is the actual value at time t, $\hat{x_t}$ is the predicted value at time t.

6.4.4. R-squared

R-squared (R^2), also known as the coefficient of determination, measures the proportion of the variance in the dependent variable (the actual values) that is predictable from the independent variable (the model's predictions). It is a relative measure of goodness of fit which indicates how well the model explains the variability of the observed data. A higher R^2 value indicates that the model explains a larger portion of the variance, thus providing a better fit (Wooldridge, 2012).

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(x_t - \widehat{x_t})^2}{\sum_{t=1}^{n}(x_t - \overline{x_t})^2} \tag{24}$$

Where, x_t is the actual value, $\widehat{x_t}$ is the predicted value, $\overline{x_t}$ is the mean of the actual values, and n is the number of observations.

7. Exploratory Data Analysis: DK1

The Exploratory Data Analysis (EDA) focuses on examining the patterns and behaviors of spot electricity prices in dk1 considering hourly, weekly, monthly, and yearly trends. Additionally, the insights into temporal dependencies and seasonal patterns in the data are explained through Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots to support model selection.

7.1. Historical Trends in Elspot (DK1) Price

The hourly and weekly (day-of-the-week) spot price plots reveal significant insights into the short-term fluctuations in electricity prices.

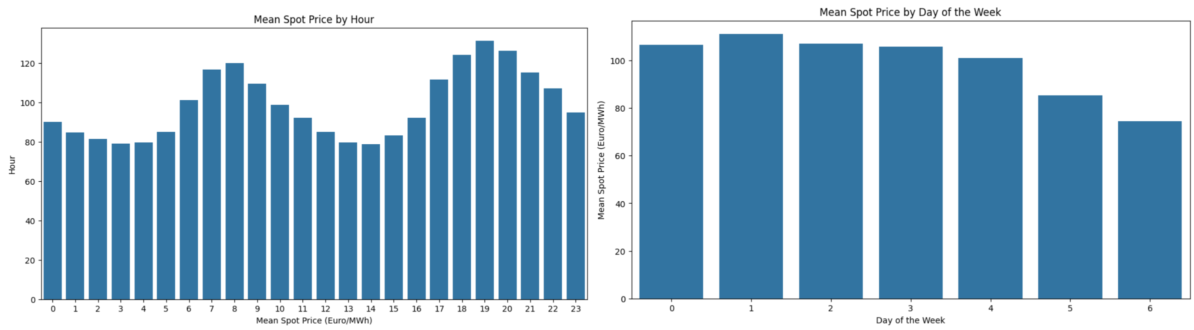


Figure 13: Mean spot prices of DK1 by hourly (13a) and weekly (13b)

The peak hours are generally observed in the early to late afternoon and evening hours (around 7-20) when electricity demand is likely higher due to increased commercial and residential usage. Conversely, prices tend to be lower during the early morning hours (0-6)due to reduced demand. The weekly plot shows mean spot prices across the days of the week where 0 denotes Monday and 6 is for Sunday. We can also find that, weekdays, from Monday to Friday it reveals a relatively consistent price level and weekends has comparatively lower prices. Together, these plots highlight a predictable, cyclical pattern in spot prices driven by variations in daily and weekly electricity demand.

Moreover, the monthly and yearly plots reveal longer-term patterns in spot prices. These plots capture seasonal fluctuations and broader economic influences.

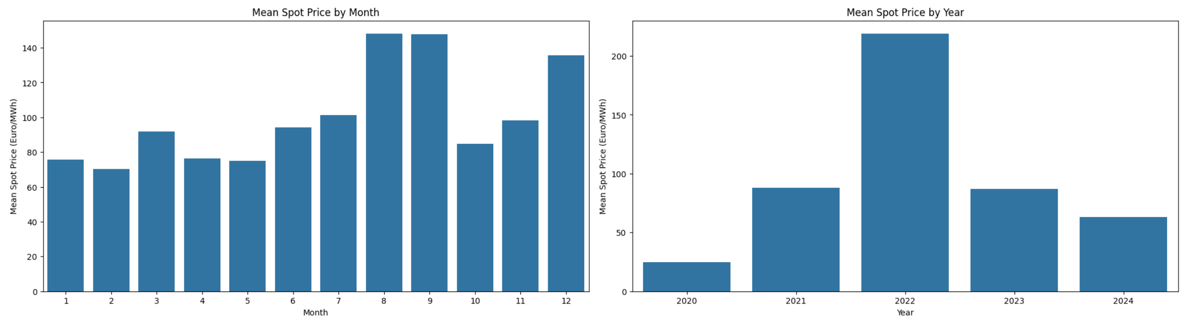


Figure 14: Mean spot prices of DK1 by monthly (14a) and yearly (14b)

The monthly plot shows that spot prices are generally higher in winter months (November-December) and late summer (July-August). Winter peaks likely reflect increased heating needs and summer peaks may be due to higher cooling demands. However, the yearly plot indicates broader market shifts with a significant peak in 2022. This spike can be attributed to external factors such as supply chain disruptions, high fuel prices, and geopolitical events like the Russia-Ukraine war which created a dramatic impact on European energy markets and drove prices to unprecedented levels (EIA, 2022; IEA, 2022). COVID-19-related disruptions could also have impacted supply chains and fuel availability in earlier years which contribute to price volatility.

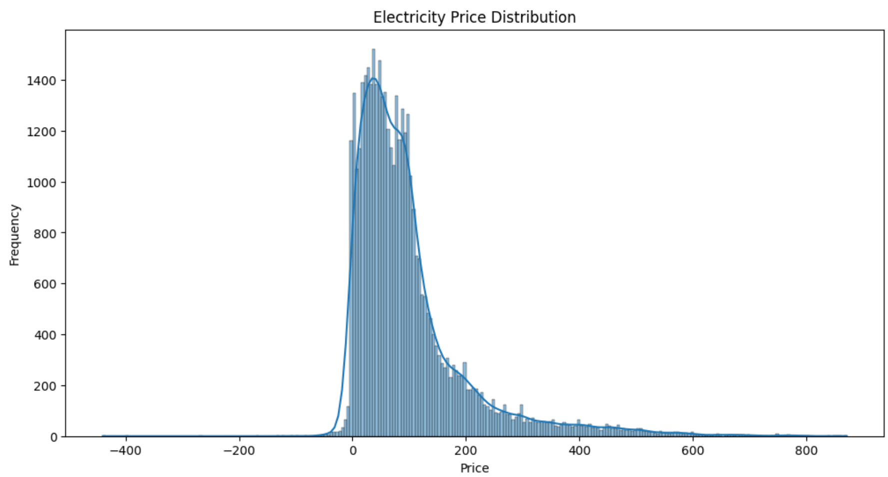


Figure 15: KDE plot for DK1 price distribution

The distribution plot of electricity prices shows the frequency of different price ranges. During periods of unexpectedly high demand or supply shortages, prices can spike sharply, observing long tail. Events like the Russia-Ukraine war, COVID-19, regulatory interventions or pricing caps have led to supply constraints and increased costs, pushing prices to extreme levels (either positive or negative). Such volatility presents challenges for consumers and producer. It necessitates implementing strategies to mitigate risk during price surges such as with long-term contracts, investing in storage solutions or diversifying energy sources.

7.2. Time-based Dependencies Indentification

Another segment of this section is explaining Autocorrelation Function (ACF) Partial Autocorrelation Function (PACF). Both of the plot are assessed to capure time-based dependencies in the data. These plots help to identify critical aspect in appropriate time series models for forecasting. The ACF and PACF plots are presented at lag intervals of 24, 48, and 72.

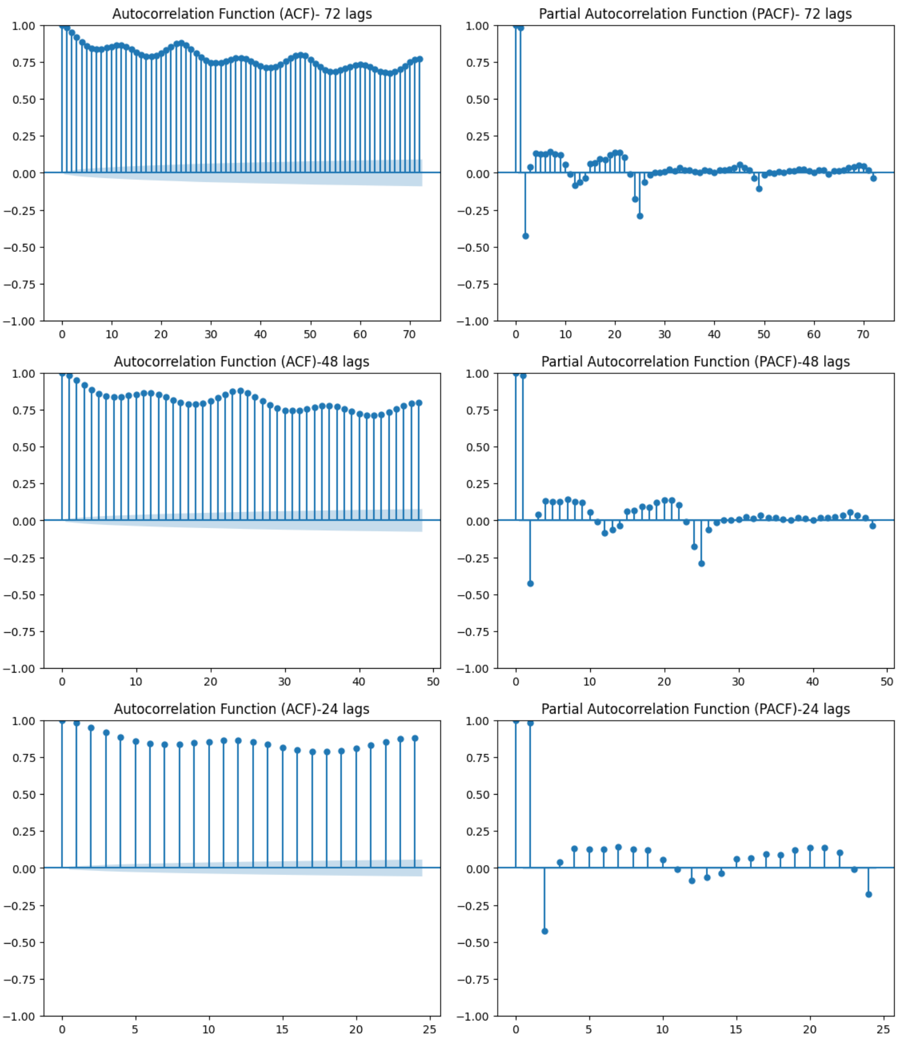


Figure 16: ACF and PACF Plot in different lags- 72, 48 and 24

At the initial stage of time series modeling, particularly when selecting parameters for ARIMA, the autoregressive (AR) and moving average (MA) terms are typically chosen based on insights from the ACF and PACF plots. For all lag levels (24, 48, and 72) in the plots, the ACF exhibits a gradual, systematic decline (geometric decay), indicating a strong autocorrelation in the series that diminishes progressively over time.

Conversely, the PACF shows notable spikes in the first few lags, specifically cut off seen at lag 2, after which it sharply declines. This pattern suggests that only a small number of PACF lags, primarily the first two, capture most of the relationship in the series. Such a rapid decline in PACF after a few significant lags, combined with the geometric decay in ACF, is indicative of an autoregressive (AR) structure, specifically suggesting an AR(2) process for this time series (Saqib Ali, 2019).

While an AR(2) procss appears based on the preliminary findings, ARIMA modeling often requires additional analysis, such as using Auto-ARIMA, to identify optimal parameters that best fit the data and forecast accuracy needs. Auto-ARIMA and other model selection techniques further refine parameter choices by evaluating multiple combinations and selecting those that minimize forecast error (Box, G. E. et al., 2008).

7.3. Volatility and Negative Pricing Impacts

Once, I came across a blog where the author used an analogy to illustrate the potential risks of prolonged negative electricity pricing. The writer compared the situation to a supermarket offering free products indefinitely. From a buyer's perspective, getting free goods might seem ideal, but from the supplier's perspective, continuous giveaways would quickly lead to financial losses, and eventually, one day, there would be no supermarket in the area.

In recent years, electricity pricing in Denmark and across the broader European market has experienced significant volatility along with frequent occurrences of zero or even negative prices.

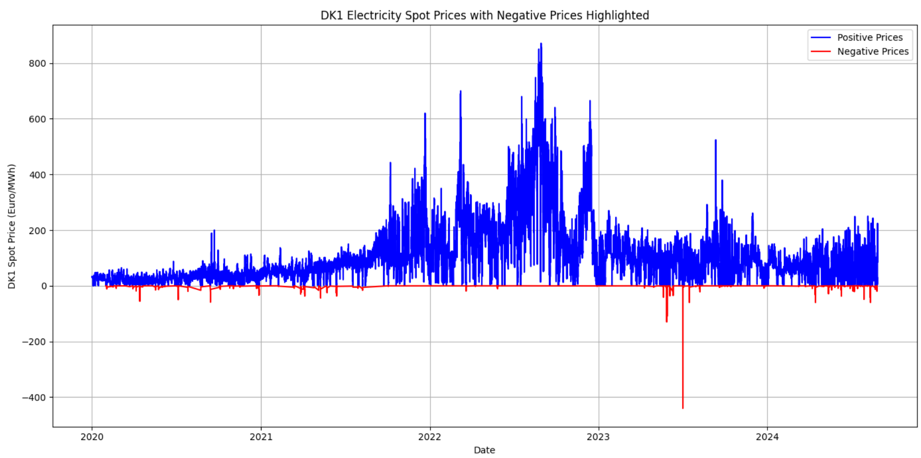


Figure 17: Time series plot with dk1 spot price

As Denmark (especially the DK1 region) and other European countries increase their reliance on these renewable sources, the electricity supply becomes more volatile. Since wind generation is unpredictable and varies with weather conditions, any rapid shift in wind speed can result in sudden price fluctuations. This weather dependency results in abrupt supply changes, which affect market prices if demand cannot adapt quickly (Paraschiv et al., 2014).

The high volatility in electricity spot prices, especially around 2022 and early 2023, can be attributed to a combination of market disruptions, geopolitical events, supply chain issues, and shifts in energy demand and supply patterns.

In 2022, electricity prices surged across Europe primarily due to the Russian invasion of Ukraine, which disrupted natural gas supplies. Russia reduced or cut off gas exports to many European countries leading to a shortage. The reduction of gas supply forced power producers to rely on more costly alternatives causing prices to spike. The heavy dependence on natural gas made European markets, including Denmark's DK1, vulnerable to these price swings (Liu et al., 2023).

Negative pricing has also become increasingly common the recent years. Negative electricity prices happen when suppliers need to pay consumers to use electricity, usually because there is an oversupply. This often occurs in markets with high renewable generation, like Denmark's DK1, where wind power can continue producing even during low demand times (e.g., at night). The extreme negative price volatility seen in mid-2023, with prices dropping below -€400, is a significant and alarming event in Denmark's energy history. This unprecedented drop is a stark indicator of ongoing challenges in managing supply and demand balance, especially in a grid with high renewable penetration and limited storage capacity. Limited storage capacity and restricted grid connections with neighboring countries prevent effective balancing of supply and demand, leading to negative prices. This situation is further complicated by grid bottlenecks with boarder countries which restrict cross-border electricity flows and contribute to price imbalances (Biber et al., 2022; Valitov, 2019).

This persistent volatility phenomenon can discourage investment in renewable and traditional energy sources due to unpredictable returns, potentially slowing down the energy transition (Schöniger & Morawetz, 2022). High and volatile prices can also impact industrial competitiveness, create potential financial and operational stresses and lead to increased costs for consumers (Prokhorov & Dreisbach, 2022). If the volatility remains unchecked, it could hinder both energy security and the reliability of the electricity market.

8. Empirical Alanysis

This section presents the empirical analysis conducted to evaluate the model ARIMAX and XGBoost.

8.1. Pre-Model Estimation Outcomes

Before building the models, it is essential to check the data's properties and ensure it meets the requirements for modeling. This section outlines the preparatory analysis before setting up the final model.

8.1.1. Missing Data Examination

The dataset used in this study is thoroughly examined for missing values to ensure data completeness and reliability for modeling. The Python method *df.isnull().sum()* is employed to compute the count of missing values for each variable in the dataset. Upon execution, the result indicated that there were no missing values in any column.

8.1.2. ADF and LB Test on Time Series

The following table presents the statistical outcomes of the Augmented Dickey-Fuller (ADF) test and the Ljung-Box test applied to the DK1 electricity spot price series. These tests were conducted to determine its suitability for time series modeling.

Test	Statistic	P-Value	Conclusion
ADF Test	-8.1332	1.076×10^{-12}	The series is stationary (reject null hypothesis).
LB Test	714461.87	0.0	The series exhibits significant autocorrelation (reject null hypothesis).

Table 2: Test Outcomes ADF and LB Test

The Augmented Dickey-Fuller (ADF) test results indicate a strong rejection of the null hypothesis of non-stationarity, with an ADF statistic of -8.1332 and a p-value close to zero. Stationarity is essential for reliable time series modeling, as it ensures that the series has consistent statistical properties over time. The Ljung-Box test further highlights significant autocorrelation within the series, with a test statistic of

714461.87 and a p-value of 0.0. This result implies that the DK1 series exhibits a pattern of autocorrelation. Together, these results underline the need for advanced modeling approaches to effectively capture the underlying pattern in the time series.

8.1.3. Training and Testing Split

The table 3 summarizes the range and dimensions of the in-sample and out-of-sample datasets used in the analysis. The dataset contains total 40926 observation for each features (15), among them 90% is used for training and remaining 10% is used as testing sample.

Data Set	Date Range	Shape
Training Data	2020-01-01 00:00:00 to 2024-03-03 14:00:00	(36639, 16)
Testing Data	2024-03-03 15:00:00 to 2024-08-20 06:00:00	(4071, 16)

Table 3: Training-testing split

The training data allows the model to learn the underlying patterns and relationships between the target variable and the other features whereas valuate the model's predictive performance on unseen data.

8.2. ARIMAX Model

The ARIMAX model is configured using the optimal parameters suggested by the lowest Akaike Information Criterion (AIC) value. After determining the best parameters, the model is set up to estimate the effects of various exogenous variables on the target variable. The outcomes include the estimated coefficients, statistical significance of variables, and diagnostic tests assessing the stationarity and normality of residuals. Each aspect of the empirical outcomes is discussed in the following subsections.

8.2.1. Auto-ARIMA for Parameter Selection

To select the optimal parameters for an ARIMA model, Auto ARIMA relies on the Akaike Information Criterion (AIC) to evaluate and compare models. A lower AIC indicates a better-fitting model. Different ARIMA configurations were tested where each associated with a unique AIC value.

ARIMA Parameters	AIC Value
(0,1,0)(0,0,0)[0]	358502.073
(1,1,0)(0,0,0)[0]	350812.154
(0,1,1)(0,0,0)[0]	351673.877
(0,1,0)(0,0,0)[0]	358500.073
(1,1,2)(0,0,0)[0]	349879.694
(2,1,1)(0,0,0)[0]	346785.680
(3,1,2)(0,0,0)[0]	346207.208
(1,1,3)(0,0,0)[0]	350385.853
(3,1,1)(0,0,0)[0]	346281.378
(3,1,4)(0,0,0)[0]	346204.411
(4,1,3)(0,0,0)[0]	346197.794
(3,1,3)(0,0,0)[0]	346198.648
(4,1,2)(0,0,0)[0]	346258.270
(5,1,3)(0,0,0)[0]	**346197.787**
(5,1,2)(0,0,0)[0]	346227.160

Table 4: AIC estimation by auto.arima model

The model with parameters (p,d,q) = (5,1,3) has the lowest AIC score (346197.794). However, models with slightly higher AIC values (like (3,1,3) and (4,1,3) can also fit the data resonably well but could be marginally less optimal than (5,1,3).

Based on the optimal parameters (5,1,3), we will set up the ARIMAX model where 5,1,3 denotes as autoregressive order of 5, integration (differencing) order of 1, and moving average order of 3. Here, 5 indicates the number of lagged values used to predict the current value (AR terms), 1 represents the degree of differencing applied to make the series stationary, and 3 specifies the number of past error terms included in the model (MA terms). Exogenous features (production data) will also be now considered to set up the ARIMAX model.

8.2.2. Interpretation of ARIMAX Estimation Results

The model's performance and reliability were evaluated using several statistical indicators, including Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Log-Likelihood, alongside diagnostic tests like the Ljung-Box (LB) test and Jarque-Bera (JB) test. We can explain the model outcomes highlighting the statistical significance, normality, and stability of the model step by step.

As we know that lowest AIC and BIC value provides optimal fit for the model, we find 305475 and 305580 as the lowest value for this ARIMAX model. AIC penalizes overly complex models whereas BIC incorporates model complexity but applies a stronger penalty for additional parameters. These parameter has more implication when comparing more than one model.

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                    dk1   No. Observations:                36639
Model:               SARIMAX(5, 1, 3)   Log Likelihood             -152713.724
Date:                Fri, 15 Nov 2024   AIC                         305475.447
Time:                        15:21:57   BIC                         305679.656
Sample:                             0   HQIC                        305540.353
                              - 36639
Covariance Type:                  opg
===============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------------
LocalPowerMWhDK1               0.0536      0.002     21.799      0.000       0.049       0.058
LocalPowerSelfConMWhDK1       -0.0726      0.026     -2.818      0.005      -0.123      -0.022
CentralPowerMWhDK1             0.0084      0.001      9.160      0.000       0.007       0.010
CommercialPowerMWhDK1         -0.1506      0.019     -8.022      0.000      -0.187      -0.114
HydroPowerMWhDK1               2.2526      1.057      2.131      0.033       0.181       4.324
OffshoreWindGe100MW_MWhDK1    -0.0075      0.001     -6.338      0.000      -0.010      -0.005
OffshoreWindLt100MW_MWhDK1    -0.0126      0.005     -2.638      0.008      -0.022      -0.003
OnshoreWindGe50kW_MWhDK1      -0.0118      0.001    -15.564      0.000      -0.013      -0.010
OnshoreWindLt50kW_MWhDK1      -0.1502      0.053     -2.825      0.005      -0.254      -0.046
SolarPowerGe10Lt40kW_MWhDK1    1.3187      0.242      5.459      0.000       0.845       1.792
SolarPowerGe40kW_MWhDK1       -0.0207      0.003     -8.087      0.000      -0.026      -0.016
SolarPowerLt10kW_MWhDK1       -0.3856      0.019    -20.102      0.000      -0.423      -0.348
SolarPowerSelfConMWhDK1       -0.0615      0.009     -6.811      0.000      -0.079      -0.044
PowerToHeatMWhDK1             -0.0662      0.002    -36.760      0.000      -0.070      -0.063
GrossConsumptionMWhDK1         0.0465      0.001     53.927      0.000       0.045       0.048
ar.L1                          0.3225      0.092      3.497      0.000       0.142       0.503
ar.L2                          0.0931      0.105      0.888      0.375      -0.112       0.299
ar.L3                          0.3443      0.085      4.070      0.000       0.179       0.510
ar.L4                         -0.2013      0.024     -8.543      0.000      -0.248      -0.155
ar.L5                         -0.0726      0.007    -10.000      0.000      -0.087      -0.058
ma.L1                         -0.2997      0.093     -3.220      0.001      -0.482      -0.117
ma.L2                         -0.3121      0.104     -3.012      0.003      -0.515      -0.109
ma.L3                         -0.5941      0.088     -6.746      0.000      -0.767      -0.421
sigma2                       214.5655      1.144    187.563      0.000     212.323     216.808
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):            935258.55
Prob(Q):                              0.93   Prob(JB):                         0.00
Heteroskedasticity (H):               4.18   Skew:                             0.44
Prob(H) (two-sided):                  0.00   Kurtosis:                        27.74
===================================================================================
```

Figure 18: ARIMAX Outcome

The log-likelihood measures how well the model's parameters explain the observed data. The log-likelihood is often negative for most time series and regression models, especially when dealing with a large number of data points. We find a negative log-likelihood value (-152713) which refer the relative magnitude of this value with other models.

The coefficient of each exogenous variable (LocalPowerMWhDK1, CentralPowerMWhDK1, etc.) indicates its estimated impact on the target variable (dk1) when holding other factors. The p-values associated with these coefficients reveal if the effect is statistically significant. In contrast, variables with higher p-values (like ar.L2) is statistically insignificant at for lag 2.

The magnitude of each coefficient shows the estimated change in dk1 prices associated with a one-unit change in the respective variable. Notably, HydroPowerMWhDK1 has a large positive coefficient, suggesting it may have a significant upward influence on prices, while SolarPowerLt10kW_MWhDK1 has a strong negative effect, indicating a downward pressure on prices when solar production is high. The value of sigma2 (sigma squared, σ^2) represents the estimated variance of the residuals (errors) in the model.

The Ljung-Box test with a p-value of 0.93 suggests that residuals are not significantly autocorrelated, implying that the model effectively captures most of the temporal dependencies. LB test also implies that errors of the model meet the partial assumption of white noise (WN) process.

Jarque-Bera (JB) test assesses the normality of residuals. The very low p-value indicates that the residuals deviate significantly from a normal distribution. This is further supported by the high skewness (0.44) and kurtosis (27.74) values which also suggest that residuals have a non-symmetric distribution with extreme outliers. Heteroskedasticity refers non-constant variance whereas homoskedasticity indicates constant variance of dataset or residuals. Here, low p-value (two-sided = 0.00) suggests that residual variance may not be constant over time which implies heteroskedasticity and disregard white noise process. This could affect the model’s ability to accurately capture fluctuations, especially during volatile periods.

8.3. XGBoost Model

As a powerful supervised machine learning algorithm, XGBoost model is designed for both regression and classification tasks. The model's parameters are optimized through cross-validation to identify the optimal set for the final model. Cross-validation has almost similar perspective to the parameter selection process in Auto ARIMA (selecting optimal p,d,q) where the algorithm iteratively analyzes data patterns to suggest the best configuration. After determining the parameter, we replaced the final model with Python.

8.3.1. Cross-Validation for Optimal Parameters

The grid search approach is used to evaluate multiple combinations of parameters. Each combination is evaluated using a validation set and the configuration with the best performance metric RMSE.

Parameter	Description	Values	Selected Optimal Parameter
max_depth	Maximum depth of trees	[3, 6, 9]	6
eta	Learning rate	[0.01, 0.1, 0.3]	0.1
min_child_weight	Minimum child weight to allow a split	[1, 3, 5]	5
subsample	Fraction of samples used per tree	[0.8, 1]	1
colsample_bytree	Fraction of features used per tree	[0.6, 0.8, 1]	0.6
gamma	Minimum loss reduction for a split	[0, 1, 5]	5
n_estimators	Number of boosting rounds	[50, 100, 200]	200
reg_alpha	L1 regularization (promotes sparsity)	[0, 0.1, 1]	0.0
reg_lambda	L2 regularization (penalizes large leaf weights)	[1, 5, 10]	1

Table 5: Cross-Validation parameter grid and selected optimal parameters

This set of optimal parameters is used in the final XGBoost model to predict electricity prices. These parameters reflect a balance between complexity (e.g., tree depth, regularization) and generalization (e.g., subsampling and column sampling), ensuring the model can perform well on both training and unseen test data. This tuning process mirrors the approach used in statistical models like Auto ARIMA, which iteratively tests configurations to identify the best fit for the data.

8.3.2. Optimal Parameter Selection

The table 6 below illustrates how the parameters of the XGBoost model identified through cross-validation align with the mathematical equations and their corresponding Python code implementation.

Equation Term	Code for Parameter	Description
$\boldsymbol{L(y_i, \hat{y}_i)}$	objective='reg:squarederror'	The loss function, L quantifies the error between actual values and predictions. For regression, squared error is used: $(y_i, \hat{y}_i)^2$
Learning rate (eta, $\boldsymbol{\eta}$)	eta=0.1	Controls the contribution of each tree to the final prediction. Smaller eta slows learning, thus requiring more trees to converge but it reduces overfitting.
Gamma, $\boldsymbol{\gamma}$	gamma=5	Minimum loss reduction required to split a node. Controls (regularize) tree complexity

Number of leaves ($\boldsymbol{T}$)	max_depth=6	Controls the maximum depth of the tree, i.e., the number of splits.
Regularization (L2, lambda, $\boldsymbol{\lambda}$)	Lambda =1	L2 regularization parameter penalizes large leaf weights, reducing overfitting. Lambda regularization term applies a penalty to the model.
Regularization (**L1**, alpha, $\boldsymbol{\alpha}$)	Alpha = 0	L1 regularization parameter promotes sparsity in leaf weights, encouraging simpler models. Default is alpha = 0.
Number of tree leaves (T)	n_estimators=200	Total number of boosting iterations (trees leaves) used. Each tree corrects the residual error of previous trees.
Minimum Weight $\boldsymbol{Weight_{min}}$	min_child_weight=5	Specifies the minimum sum of instance weights needed in a leaf node. Prevents overfitting by avoiding overly specific leaf splits.

Table 6: Aligning equation terms with XGBoost code parameters

Source: Own illustration, inspiration from (Wang, Y. & Ni, 2019)

8.3.3. Model Boosting Rounds

XGBoost works by sequentially adding decision trees to the model, with each tree attempting to correct the errors made by the previous ones. This process is known as boosting which helps to minimize the error step by step and improve the accuracy for precise prediction. The plot provides a clear representation of how the model's performance improves as the number of boosting iterations increases. Along with improving the accuracy of the model, each round aims at reducing the residual errors from previous rounds. The y-axis represents the RMSE value while the x-axis represents the boosting iterations or the number of trees added to the model.

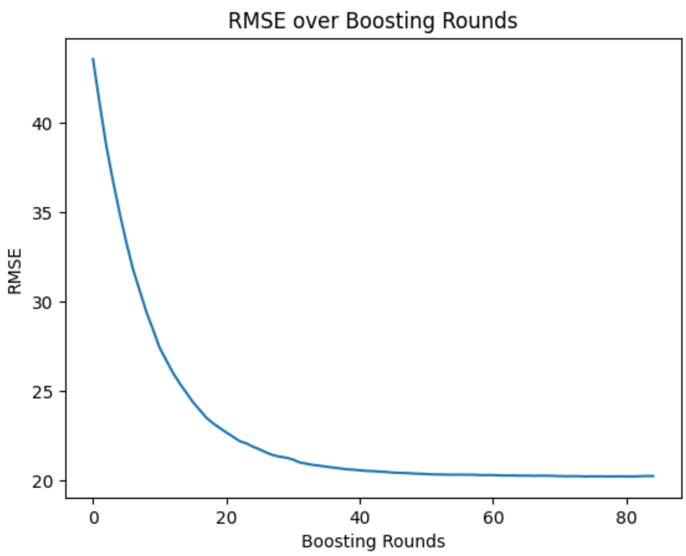


Figure 19: RMSE over Boosting Round

At the beginning of the training process, the RMSE starts at a relatively high value (above 40). This reflects that, initially, the model performs less accurate because it has very few trees or weak learners. However, as more trees are added (as indicated by increasing boosting rounds on the x-axis), the RMSE decreases sharply. This steep decline indicates that the model is quickly learning patterns in the data and substantially improving its predictive accuracy over the first few iterations. After around 40 boosting rounds, the RMSE curve begins to flatten out. This indicates that the model's performance is no longer improving at the same rate. By this point, most of the easily learnable patterns in the data have been captured by the model. Beyond this stage, adding more trees results could only consider as marginal improvements in the RMSE. The model begins to approach its minimum RMSE value slightly above 30 in this case. We it could be interpreted that the model has reached a point where adding further trees does not substantially reduce error. So, according to the RMSE plot, any further addition of trees may not provide significant benefits and could lead to overfitting if unchecked.

8.4. Model Comparison: ARIMAX and XGBoost

This section provides a comparative analysis of the models based on evaluation metrics and highlights their respective strengths and weaknesses in forecasting accuracy and adaptability.

8.4.1. KDE and Residual Distribution

The histograms and Kernel Density Estimates (KDE) for both the ARIMAX and XGBoost residuals reveal key differences in the distribution of errors.

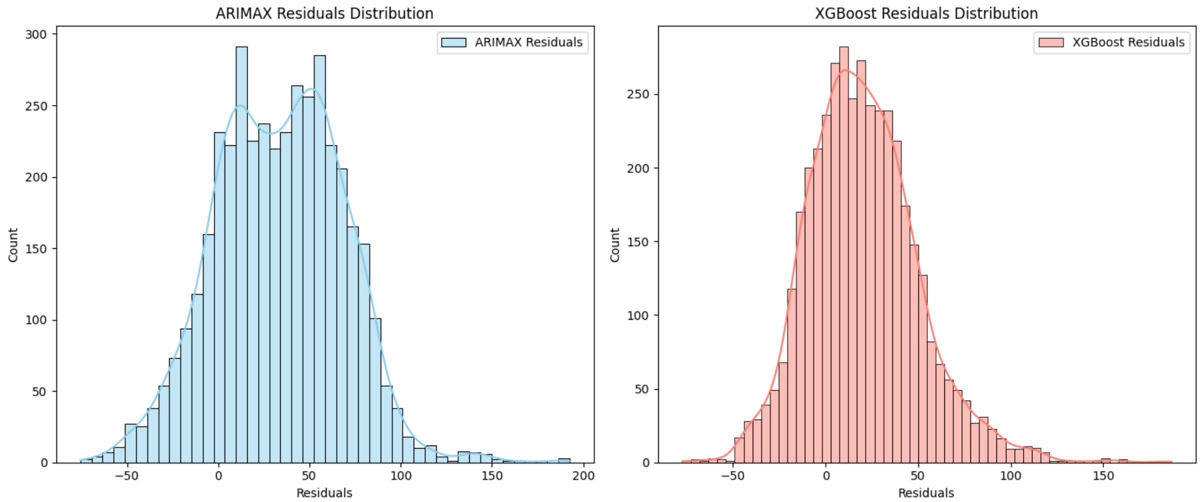


Figure 20: Distribution and Shape of KDE

The residual distribution comparison between two models reveals distinct differences in their predictive accuracy. ARIMAX residuals display a wider spread, indicating that the model often makes larger errors. On the other hand, XGBoost residuals are more compact, suggesting more accurate and consistent predictions. The ARIMAX distribution has dual peaks which could signal systematic errors or patterns that the model fails to address whereas XGBoost exhibits a single sharp peak around zero, showing that its predictions are closer to the actual values more frequently.

The tails of the ARIMAX residuals are heavier which reflects higher occurrences of extreme errors in both overestimation and underestimation. In contrast, XGBoost has lighter tails which suggests that it can handle unusual values better and makes fewer extreme errors. The bell-shaped, smooth curve of XGBoost residuals further supports its ability to capture the underlying patterns in the data with greater reliability. Overall, XGBoost demonstrates superior performance, producing residuals that are less biased and more tightly centered around zero.

8.4.2. Model Performance

This comparison between ARIMAX and XGBoost models is evaluated based on their in-sample (training) and out-of-sample (testing) performance metrics:

Metrics	ARIMAX (Training)	ARIMAX (Testing)	XGBoost (Training)	XGBoost (Testing)
MAE	8.81	39.66	21.43	27.07
MSE	244.45	2380.49	1188.73	1276.70
RMSE	15.63	48.79	34.97	35.73
R-squared	0.979		0.897	

Table 7: Model evaluation matrics (MAE, MSE, RMSE, R-squared)

ARIMAX demonstrates exceptional performance on the training data as reflected by its significantly lower error metrics (MAE: 8.81, MSE: 244.45, RMSE: 15.63) and an R-squared value of 0.979. These evaluation matrics indicate that nearly all the variance in the target variable is explained. In contrast, XGBoost's training performance, this model still perform exceptional (MAE: 21.43, MSE: 1188.73, RMSE: 34.48, R^2 **:** 0.897), however, is less precise than ARIMAX. It concludes that ARIMAX has strong ability to fit the training data patterns. However, when evaluating testing performance, we can see a significant contrast. ARIMAX performs poorly on unseen data, as evidenced by much higher error metrics (MAE: 39.66, MSE: 2380.49, RMSE: 48.79). In contrast, XGBoost exhibits much better capabilities on test dataset, with lower error metrics (MAE: 27.07, MSE: 1276.70, RMSE: 35.73) indicating that it retains a good predictive power on unseen data.

When we visualize the residual plots on the testing data, they reveal critical. While the R-squared value for ARIMAX is higher, this alone cannot be used as a definitive indicator of superior model performance. The R-squared value is primarily a measure of how much variance in the dependent variable is explained by the model. However, in the context of forecasting, especially in volatile and complex datasets such as electricity prices, relying solely on R-squared can be misleading.

The residuals for the ARIMAX model (Figure 21) demonstrate high volatility and significant deviations from zero.

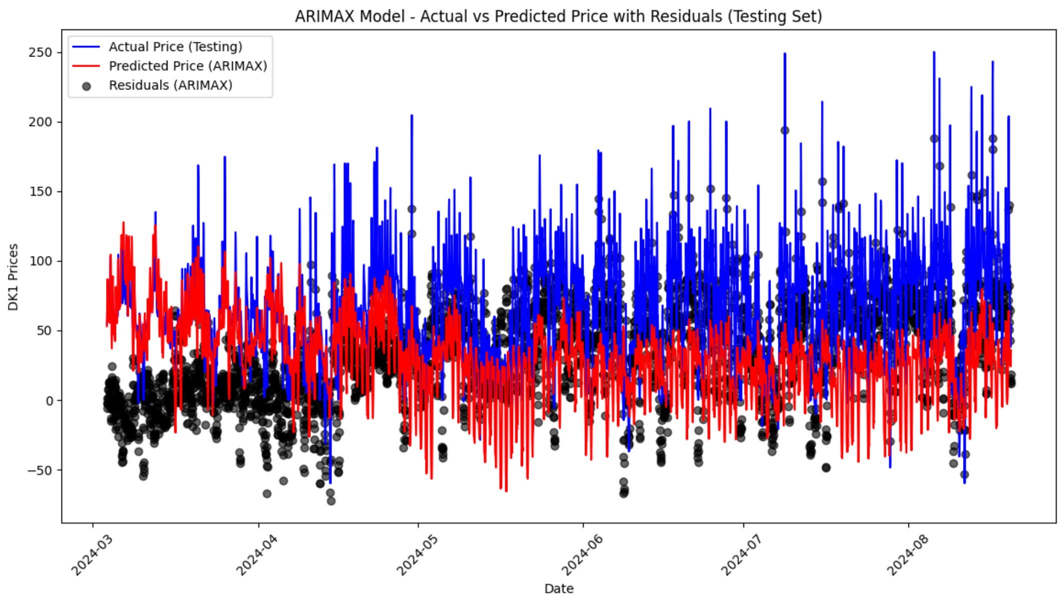


Figure 21: Residuals concentration on test data (ARIMAX Model)

This indicates that the ARIMAX model struggles to capture the underlying patterns in the testing data effectively. The broader spread and higher variability in the residuals imply that ARIMAX's predictions are frequently far from the actual values, which results in larger forecast errors. Although ARIMAX achieves excellent performance on the training dataset, its effectiveness diminishes significantly on the testing dataset which indicates a tendency to overfit.

In contrast, the residuals for XGBoost (Figure 22) are much more concentrated around zero. This refers that the model's predictions are closer to the actual values.

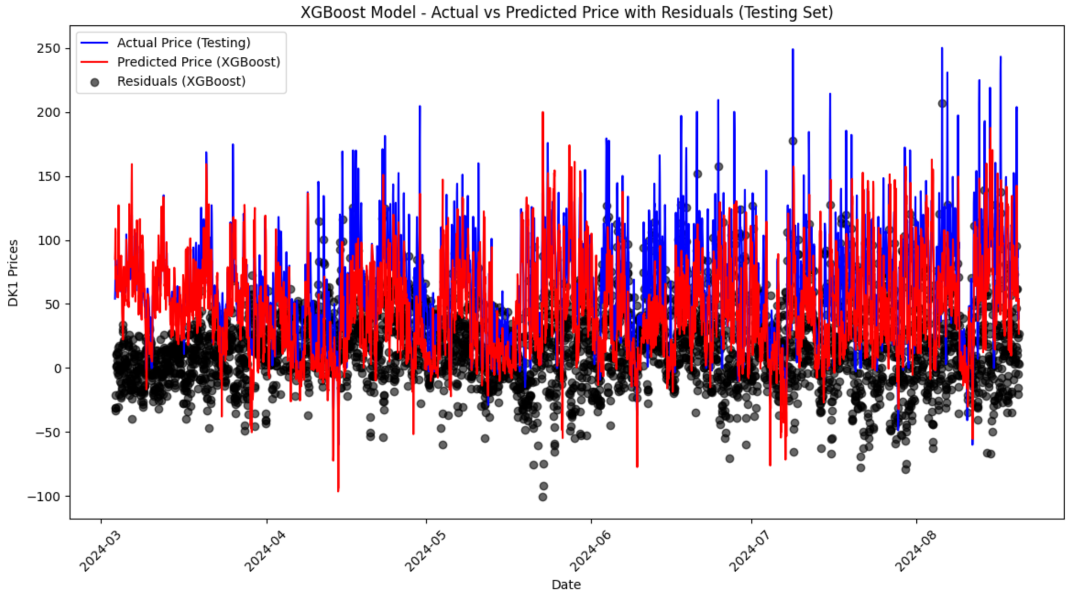


Figure 22: Residuals concentration on test data (XGBoost Model)

The lower variability and reduced volatility in the XGBoost residuals suggest the effective ability to capture the complex, nonlinear relationships in the data and performs well in predicting unseen data.

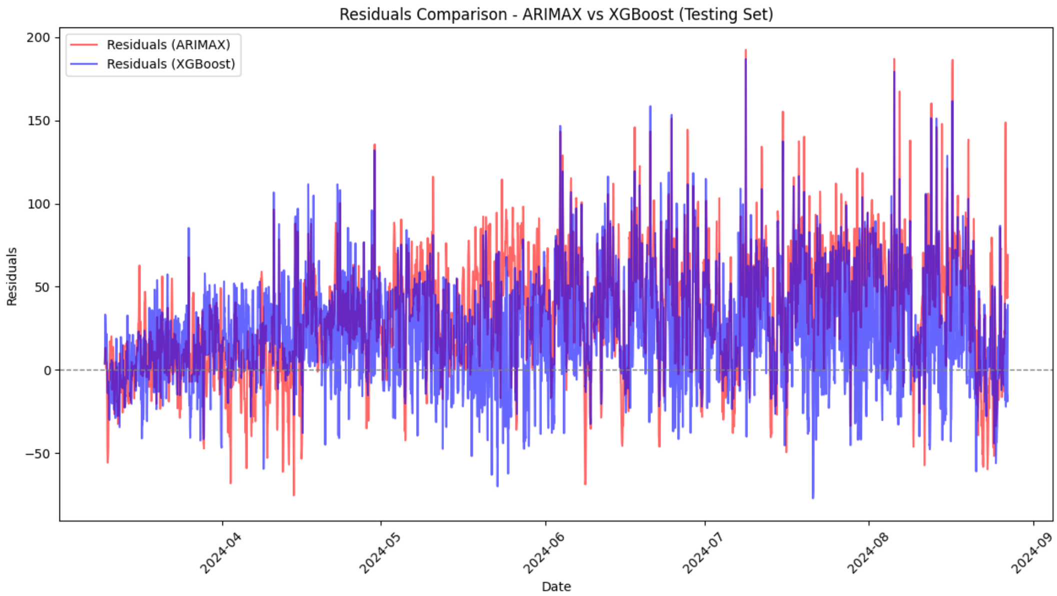


Figure 23: Residuals overlaping each other on both models- ARIMAX & XGBoost

However, the residual overlap (Figure 23) plot provides a clear visualization of the comparative performance. The residuals for XGBoost are tightly clustered around zero, whereas ARIMAX residuals exhibit greater fluctuations and a broader spread. This difference highlights XGBoost's superior ability to align its predictions with actual data points in the testing set.

The root cause of this difference could lie in the nature of the two models. ARIMAX is highly dependent on the autoregressive and moving average components to capture temporal patterns, which often results in overfitting, especially when the dataset includes volatility or noise. Conversely, XGBoost, as a gradient-boosting framework, is inherently designed with regularization mechanisms (e.g., controlling tree depth, learning rate, and subsampling) that prevent overfitting and promote better generalization.

8.4.3. Time-Based Errors comparison

The time-based error analysis conducts to illustrate different time-dependent patterns in electricity prices. This section vizualizes the behavior of errors hourly and daily basis.

Hourly Mean Absolute Error Analysis

The hourly MAE plots indicate that both models display a distinctive pattern across different times of the day with varying levels of accuracy.

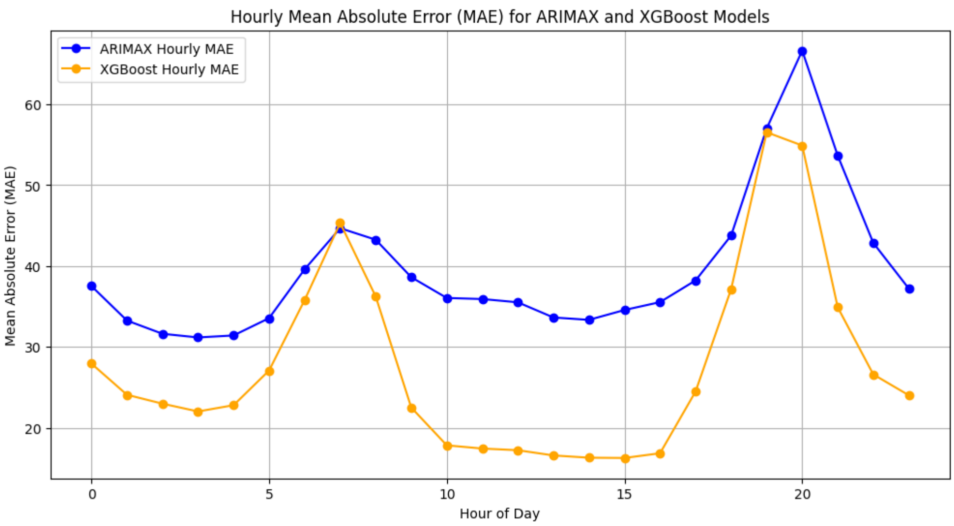


Figure 24: Hourly Mean Absolute Error of AIMAX and XGBoost

The ARIMAX model shows consistently higher errors during peak hours, particularly around 7 p.m. (19.00) where the MAE reaches above 60. This suggests that ARIMAX struggles to accurately predict prices during peak demand times possibly due to the increased volatility and complex price behavior during these hours. XGBoost, on the other hand, exhibits a relatively smoother performance curve with lower MAE values, particularly during peak and off-peak hours especially from the morning to afternoon. However, XGBoost also spikes at peak hours (around 7 p.m.), though it manages to maintain a lower error than ARIMAX.

Daily Mean Absolute Error Analysis

The daily MAE analysis highlights a clear discrepancy between ARIMAX and XGBoost performance across different days of the week.

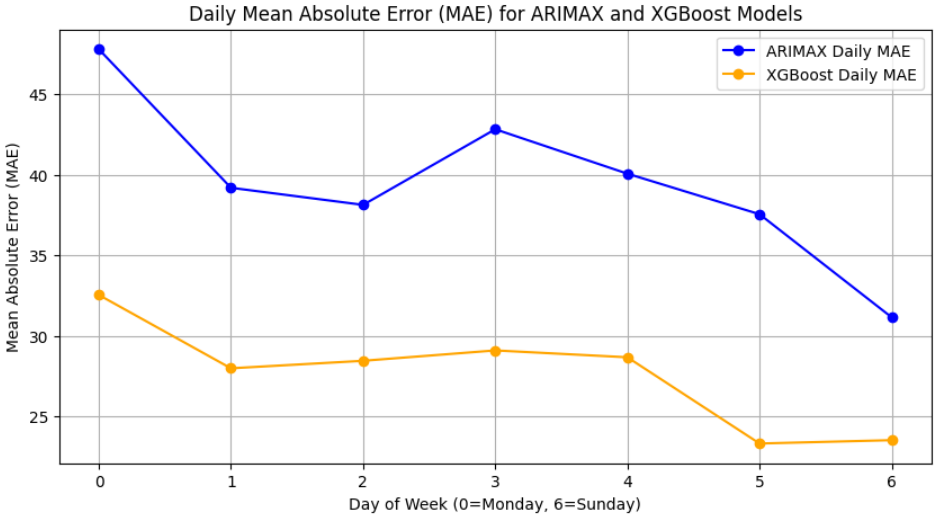


Figure 25: Daily Mean Absolute Error of AIMAX and XGBoost

ARIMAX consistently shows higher MAE values across all days where the highest errors occurring on Mondays. This suggests a possible lag in the model's ability to adjust after the weekend when the market might experience different dynamics. XGBoost displays comparatively lower MAE values and a more stable error distribution across the week. This indicates that it can adapt better to weekly price trends and manages to capture the fluctuations more consistently.

Hourly and Daily RMSE Analysis

The RMSE values are more accepted due to the squaring of errors. Compared to the MAE plot, we can see the trend is similar. The RMSE plot similarly highlights peak hours and the larger errors on Monday as the most challenging period for both models.

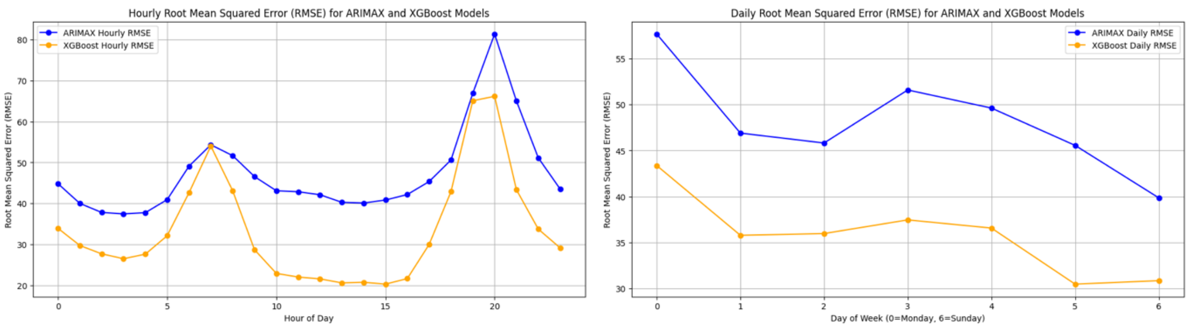


Figure 26: Hourly (26a) and Daily (26b) Root Mean Squared Error of AIMAX and XGBoost

The RMSE analysis, along with the MAE, suggests that XGBoost is more resilient in capturing hourly and daily patterns with lower error magnitude across both metrics. While both models face difficulties during peak demand hours and start-of-week days. XGBoost's lower RMSE during these periods indicates that it captures the underlying price variations with greater precision and stability than ARIMAX. This advantage of XGBoost makes it a preferable model for applications requiring accuracy during high-volatility periods.

8.4.4. t-test and Diebold-Mariano Test on Error

The Diebold-Mariano (DM) evaluate whether the forecast accuracy of two models is significantly different.

Diebold-Mariano Test	Between ARIMAX and XGBoost	Result Comparison
Test Statistic	18.28	Statistically significant difference ($p < 0.05$)
P-Value (DM Test)	3.393e-73	Very low p-value; significant difference in model accuracy

Table 8: Diebold-Mariano Test on two groups (ARIMAX and XGBoost Error)

The test compares the forecasting accuracy of two models- ARIMAX and XGBoost with test statistic 18.28 and a p-value of 3.393e-73 which is significantly below the typical threshold of 0.05. This indicates that there is a statistically significant difference in the forecasting performance of the two models.

Since we find consistently lower error metrics for XGBoost, mentioned in the previous sections, the DM test statistic validates that XGBoost outperforms ARIMAX in forecasting accuracy

To sum up, this statistical comparison highlights the robustness of XGBoost for Dk1 price forecasting applications.

8.4.5. Forecasting Performance Over Different Horizons

In this section, we draw an empirical evaluation of the forecasting performance of ARIMAX and XGBoost models over different horizons: 1-day, 2-day, and 7-day forecasts. The comparative analysis is conducted using key metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

Forecast Horizon	Model	Mean Absolute Error (MAE)	Mean Squared Error (MSE)	Root Mean Squared Error (RMSE)
1-Day	ARIMAX	40.35	2181.23	46.70
	XGBoost	22.69	1045.90	32.34
2-Day	ARIMAX	29.60	1317.28	36.29
	XGBoost	20.47	803.43	28.34
7-Day	ARIMAX	33.14	1677.47	40.96
	XGBoost	17.95	610.72	24.71

Table 9: MAE, MSE, RMSE on out-of-sample for 1-day 1, 2-day and 7-day

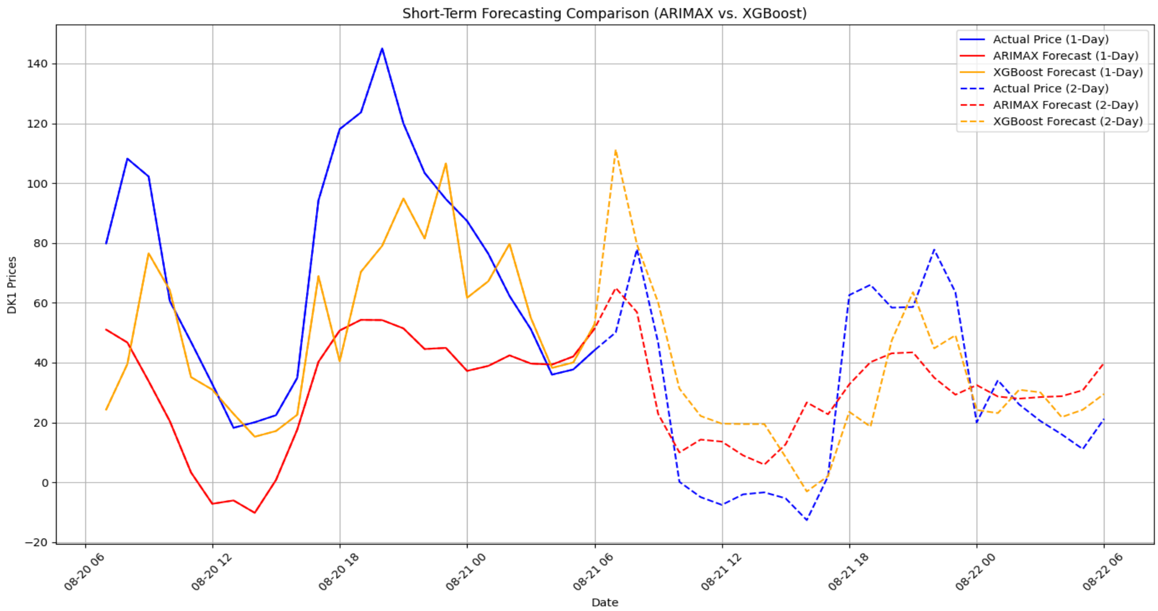


Figure 27: Short-term Forecasting comparizon on unseen sample (1-day and 2-day)

(Forecasting range: August 20, 5 am to August 22, 6am 2024)

For 1-day ahead forecasting, XGBoost outperforms ARIMAX in terms of accuracy and error metrics. XGBoost achieves a much lower MAE of 22.69 compared to ARIMAX's 40.35. Similarly, XGBoost has lower MSE (1045.90) and RMSE (32.34) compared to ARIMAX's MSE (2181.23) and RMSE (46.70). These results indicate that XGBoost has a better short-term predictive ability, providing more accurate and stable forecasts for the immediate next day.

In the 2-day forecast, XGBoost again demonstrates superior performance over ARIMAX. The MAE for XGBoost decreases to 20.47 compared to ARIMAX's 29.60 and the RMSE for XGBoost is (28.34) is significantly lower than ARIMAX (36.29). ARIMAX, while improving slightly from the 1-day performance, however, it still struggles with higher error levels.

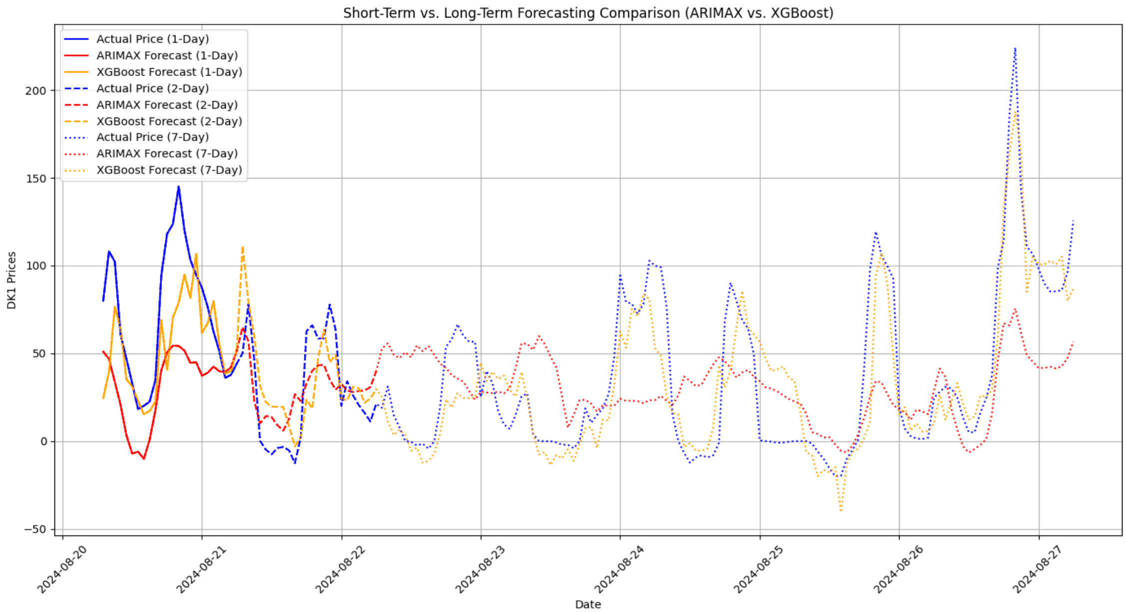


Figure 28: Long-term Forecasting comparizon on unseen sample (7-day)
(Forecasting range: August 20 to August 27, 2024)

To assess the long term model performance, 7-day forecast clearly exhibits XGBoost's dominance in long-term forecasting. XGBoost achieves the lowest MAE (17.95), MSE (610.72), and RMSE (24.71) across all horizons. In contrast, ARIMAX's error metrics (MAE: 33.14, MSE: 1677.47, RMSE: 40.96) show that the model increasingly struggles as the forecast horizon extends. This reflects ARIMAX's inherent limitation in handling long-term dependencies, probably, due to its reliance on autoregressive and moving average components. Conversely, better performance of XGBoost can be attributed to its ability to model complex nonlinear relationships and incorporate regularization mechanisms and thus making it more robust over extended periods.

The visualizations clearly show that XGBoost aligns more closely with the actual values for all forecast horizons. However, ARIMAX forecasts tend to deviate significantly, especially in long-term predictions. XGBoost provides smoother and more consistent predictions, reducing the likelihood of extreme deviations. So, the accompanying visualizations and the tabulated metrics provide an in-depth understanding of how each model performs as the forecasting horizon lengthens.

Overall, XGBoost emerges as the superior model across all forecast horizons. While ARIMAX demonstrates strong in-sample performance, its poor generalization on out-of-sample data results in higher forecast errors, particularly over extended periods. XGBoost's consistent ability to maintain lower error metrics and better alignment with actual data across short-term and long-term forecasts underscores its robustness and adaptability. Therefore, XGBoost is recommended for both immediate and longer-term electricity price forecasting in this study.

9. Discussion

This section provides the broader implications of the key findings, exploring the energy market challenges and suggestions for future research to enhance forecasting methodologies in complex energy systems.

9.1. Volatile Trends in DK1 Spot Prices and Impact on Modeling

The electricity spot price for Denmark's DK1 region is highly susceptible to fluctuations driven by supply-demand imbalances, renewable energy penetration, and external geopolitical factors. As illustrated in time serie EDA analysis, DK1 spot prices exhibit periods of extreme volatility, with significant spikes and frequent dips into negative prices. This variability poses unique challenges for predictive modeling, as the data does not adhere to a consistent trend or seasonality pattern. Spot price volatility, especially with sudden shifts, often requires models that can dynamically adjust to changing patterns. This volatility has a direct impact on forecasting model performance. Traditional time series models, such as ARIMAX, struggle to handle this high level of fluctuation due to their reliance on historical trends and inherent assumptions of data stationarity and linearity. Machine learning models, particularly XGBoost, have shown superior performance with volatile datasets like DK1 spot prices.

9.2. Negative Prices: Challenges for Market Participants

The occurrence of negative electricity prices, as observed in the DK1 market, poses significant challenges for market participants. While negative prices may appear advantageous to consumers in the short term, they present a "curse" to market participants, especially traditional generators who struggle to maintain profitability. Over time, persistent negative prices can undermine investment in both renewable and conventional energy sources, deterring the expansion of a robust energy infrastructure that balances reliability with environmental goals. For market participants, negative pricing disrupts traditional revenue models, compelling them to seek alternative approaches to manage risks, such as forward market participation or hedging strategies.

9.3. Recent Hardships in the Energy Market

The European energy market has faced unprecedented challenges in recent years, largely stemming from a series of global and regional disruptions. One primary factor has been the geopolitical tension surrounding Russia's invasion of Ukraine, which led to a reduction in Russian natural gas exports to Europe. Additionally, Germany's decision to phase out (gradually closing) nuclear power plants has increased reliance on fossil fuels like natural gas and coal to meet its energy needs. This dependence on fossil fuels makes the energy supply more vulnerable to price spikes and fluctuations in international energy markets. Moreover, China has shifted partly to LNG based power production to reduce reliance on coal, which is more polluting. So, a significant portion of the world's LNG supply now goes to China, leaving less for other regions like Europe. This increased demand from Asia intensifies the competition for LNG supplies in Europe and making the European market more volatile. However, high inflation rates and rising interest rates also add to the financial burdens on energy companies, making it more expensive to invest in renewable energy projects or maintain existing infrastructure.

9.4. Key Findings

The comparative analysis in this study reveals that while the ARIMAX model excelled in in-sample accuracy with significantly lower training error metrics, it struggled with out-of-sample forecasting. This performance indicates overfitting where the model captures the training data patterns but fails to adapt to unseen scenarios, such as volatility or noise. Conversely, the XGBoost model demonstrated consistent and robust performance across training and testing datasets. XGBoost showcased its ability to generalize better to new data regardless of short-term and long-term forecating proving its adaptability to varying temporal patterns. The findings conclude that XGBoost, with optimal parameter tuning are comparatively better suited for dynamic and volatile markets like energy forecasting of DK1 zone.

9.5. Limitations

This study acknowledges certain limitations that provide opportunities for future improvement and exploration.

Both ARIMAX and XGBoost models rely on historical data to make predictions. However, electricity prices are influenced by sudden external factors such as geopolitical events, extreme weather conditions, or unexpected policy changes, which are not always reflected in historical datasets. These unpredictable factors may lead to deviations in model predictions.

The study focuses on the DK1 region in the Danish energy market. While the findings offer valuable insights into price forecasting for renewable energy-dominated markets, they may not be directly applicable to other regions with different energy mixes, regulatory frameworks, or market dynamics.

While XGBoost outperformed ARIMAX across both short-term and long-term horizons, the models' performance was only evaluated up to a 7-day horizon. Extending the analysis to longer forecasting periods could offer further insights into their robustness and applicability.

The study primarily relied on standard evaluation metrics such as MAE, MSE, RMSE, and R-squared for assessing model performance. While these metrics are widely accepted, they may not capture all aspects of model utility, particularly under extreme price scenarios.

9.6. Future Research Directions

For hourly energy price prediction, the latest research explores several modern financial, economic, and machine learning approaches. Some Deep Learning appaches e.g. LSTM is widely using for better modelling with hourly energy prices. Due to its ability to capture time dependencies, LSTM has been widely used in energy price prediction. A recent study by (Li & Becker, 2021; Zhou et al., 2019) demonstrated that LSTM effectively models non-linear temporal relationships in electricity prices. Another variation of LSTM, Gated Recurrent Unit (GRU) has been shown to perform similarly but with a simpler architecture. (Gao et al., 2019) tested GRU for energy prices, finding it effective in managing longer time-series dependencies while reducing computational costs. Moreover, a time series model, Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models, have also been adapted for energy prices because of its ability to capture volatility clustering in financial time series. Furthermore, incorporating additional external features, such as real-time weather data or policy shifts, could further enhance model accuracy.

10. Conclusion

Price forecasting accuracy has direct and far-reaching implications for both strategic market participation and risk management in the energy sector. With reliable spot price predictions, market participants are better equipped to make informed bidding decisions, optimize trading strategies and manage risks in both the spot and forward markets. Accurate forecasting thus not only provides financial and operational benefits for individual participants but also fosters a more stable and efficient market environment supporting the broader objectives of sustainable energy investment and market reliability. In the dynamic nature of the energy sector, with increasing integration of renewable sources, continued advancements in forecasting techniques are crucial for enabling a resilient and adaptable electricity market.

This research explored the comparative performance of ARIMAX and XGBoost models in forecasting electricity prices with a particular focus on the unique characteristics of the DK1 market. By incorporating exogenous variables, ARIMAX attempts to enhance prediction accuracy leveraging its strength in capturing autoregressive and moving average patterns. However, the results indicate that ARIMAX struggles to perform on unseen data, particularly in volatile market conditions. While ARIMAX is effective in capturing historical trends and seasonality, its limitations are found with the complex and nonlinear dynamics of modern energy markets.
Conversely, XGBoost, a machine learning model, consistently outperformed ARIMAX in both short-term and long-term forecasting horizons. Its ability to handle nonlinear relationships, integrate exogenous features and prevent overfitting through regularization techniques proved it as a well-performed model in the volatile market. The model demonstrated superior capability, lower forecasting errors and better alignment with real-world electricity price fluctuations. The empirical findings highlight the growing importance of machine learning approaches in energy price forecasting, especially in markets characterized by high volatility and dynamic trends like DK1.

This thesis also examined the forecasting performance across different time horizons. While ARIMAX maintained competitive accuracy for very short-term horizons, its predictive power diminished significantly for longer-term forecasts. XGBoost, on the other hand, exhibited robust performance across all forecasting horizons, making it a more reliable option for market participants aiming to make informed decisions in both day-ahead and long-term trading strategies. From a methodological perspective, this research underscores the necessity of rigorous model evaluation including statistical tests such as the Diebold-Mariano test and comprehensive error analysis to ensure robust conclusions.

Ultimately, this study provides a valuable contribution to the field of energy forecasting by demonstrating the comparative strengths and limitations of ARIMAX and XGBoost. While ARIMAX serves as a solid foundation for understanding time-series dynamics, machine learning models like XGBoost offer a promising capability for capturing the complexities of modern energy systems. These findings pave the way for further research into hybrid models and the integration of alternative machine learning approaches to improve forecasting accuracy.

11. Reference

ACER. (2023). *Policy Paper on The Further Development of the EU Electricity Forward Market.* (). https://www.acer.europa.eu/sites/default/files/documents/Position%20Papers/Electricity_Forward_Market_PolicyPaper.pdf

ACER. (2024, *"Agency for the Cooperation of Energy Regulators: Market Monitoring."* . Retrieved 11 October, 2024, from www.acer.europa.eu

Aduda, J., Weke, P., Ngare, P., & Mwaniki, J. (2016). Financial time series modelling of trends and patterns in the energy markets. *Journal of Mathematical Finance, 6*(2), 324–337.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*(6), 716–723.

Austnes, P. F., Riemer-Sørensen, S., Bordvik, D. A., & Andresen, C. A. (2024). Balancing the Norwegian regulated power market anno 2016 to 2022. *Energy Strategy Reviews, 52*, 101331. 10.1016/j.esr.2024.101331

Benth, F. E., Benth, J. S., & Koekebakker, S. (2008). *Stochastic modelling of electricity and related markets*. World Scientific.

Biber, A., Felder, M., Wieland, C., & Spliethoff, H. (2022). Negative price spiral caused by renewables? Electricity price prediction on the German market for 2030. *The Electricity Journal, 35*(8), 107188. 10.1016/j.tej.2022.107188

Bordignon, S., Bunn, D. W., Lisi, F., & Nan, F. (2013). Combining day-ahead forecasts for British electricity prices. *Energy Economics, 35*, 88–103.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2008). Time series analysis: forecasting and control John Wiley & Sons. *Hoboken, NJ,*

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific Model Development, 7*(3), 1247–1250.

Chen, T. (2015). Xgboost: extreme gradient boosting. *R Package Version 0.4-2, 1*(4)

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. Paper presented at the *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining,* 785–794.

Contreras, J., Espinola, R., Nogales, F. J., & Conejo, A. J. (2003). ARIMA models to predict next-day electricity prices. *IEEE Transactions on Power Systems, 18*(3), 1014–1020.

Cryer, J. D., & Kellet, N. (1991). *Time series analysis*. Springer.

D O'Gorman, K., & MacIntosh, R. (2015). *Research methods for business and management: A guide to writing your dissertation*. Goodfellow Publishers Ltd.

Danish Energy Agency. (2021). *Energy Statistics 2021.* (). https://ens.dk/sites/ens.dk/files/Statistik/energy_statistics_2021.pdf

Danish Energy Agency. (2022, Retrieved 20 September, 2024, from www.ens.dk

Danish Ministry of Climate, Energy and Utilities. (2021)-Energy Islands: A New Frontier in Offshore Wind. Retrieved 17 September, 2024, from https://stateofgreen.com/en/

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association, 74*(366a), 427–431.

Diebold, F. X., & Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & Economic Statistics, 20*(1), 134–144.

Dubois, A., & Gadde, L. (2002). Systematic combining: an abductive approach to case research. *Journal of Business Research, 55*(7), 553–560.

EIA. (2022). *Short-Term Energy Outlook.* ().

Energinet. (2022, *Electricity Market and Grid Operations in Denmark.* Retrieved 5 October, 2024, from https://en.energinet.dk/

EPEX : European Power Exchange. "Day-Ahead and Intraday Markets". Retrieved 8 September, 2024, from www.epexspot.com

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics, ,* 1189–1232.

Fu, Y., Li, Z., Zhang, H., & Xu, P. (2015). Using Support Vector Machine to Predict Next Day Electricity Load of Public Buildings with Sub-metering Devices. *Procedia Engineering, 121*, 1016–1022. 10.1016/j.proeng.2015.09.097

Gao, X., Li, X., Zhao, B., Ji, W., Jing, X., & He, Y. (2019). Short-term electricity load forecasting model based on EMD-GRU with feature selection. *Energies, 12*(6), 1140.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer.

Hernández Rodríguez, M., Baca Ruiz, L. G., Criado Ramón, D., & Pegalajar Jiménez, M. d. C. (2023). Artificial Intelligence-Based Prediction of Spanish Energy Pricing and Its Impact on Electric Consumption. *Machine Learning and Knowledge Extraction, 5*(2), 431–447.

Hyndman, R. J. (2018a). *Forecasting: principles and practice*. OTexts.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software, 27*, 1–22.

IEA. (2022). *World Energy Outlook 2022.* ().

Jarque, C. M., & Bera, A. K. (1987). A test for normality of observations and regression residuals. *International Statistical Review/Revue Internationale De Statistique, ,* 163–172.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems, 30*

Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance, 34*(11), 2767–2787.

Kiesel, R., & Kusterman, M. (2016). Structural models for coupled electricity markets. *Journal of Commodity Markets, 3*(1), 16–38.

Lago, J., De Ridder, F., & De Schutter, B. (2018). Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Applied Energy, 221*, 386–405.

Lago, J., Marcjasz, G., De Schutter, B., & Weron, R. (2021). Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy, 293*, 116983.

Li, W., & Becker, D. M. (2021). Day-ahead electricity price prediction applying hybrid models of LSTM-based deep learning methods and feature selection algorithms under consideration of market coupling. *Energy, 237*, 121543.

Liu, F., Su, C. W., Qin, M., & Lobonţ, O. (2023). Winner or loser? The bidirectional impact between geopolitical risk and energy transition from the renewable energy perspective. *Energy, 283*, 129174.

Ljung, G. M., & Box, G. E. (1978). On a measure of lack of fit in time series models. *Biometrika, 65*(2), 297–303.

Lucic, M., & Xydis, G. (2023). Performance of the autoregressive integrated moving average model with exogenous variables statistical model on the intraday market for the Denmark-West bidding area. *Energy & Environment,* , 0958305X231199154. 10.1177/0958305X231199154

Munoz, M. A., Morales, J. M., & Pineda, S. (2020). Feature-driven improvement of renewable energy forecasting and trading. *IEEE Transactions on Power Systems, 35*(5), 3753–3763.

Nagelkerke, N. J. (1991). A note on a general definition of the coefficient of determination. *Biometrika, 78*(3), 691–692.

Nord Pool: The Power Market Explained. Retrieved 10 October, 2024, from https://www.nordpoolgroup.com/en/trading/Day-ahead-trading/Price-calculation/

Pankratz, A. (2012). *Forecasting with dynamic regression models*. John Wiley & Sons.

Paraschiv, F., Erni, D., & Pietsch, R. (2014). The impact of renewable energies on EEX day-ahead electricity prices. *Energy Policy, 73*, 196–210.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems, 31*

Prokhorov, O., & Dreisbach, D. (2022). The impact of renewables on the incidents of negative prices in the energy spot markets. *Energy Policy, 167*, 113073. 10.1016/j.enpol.2022.113073

Rosellón, J., & Kristiansen, T. (2013). Financial Transmission Rights. *Analysis Experiences and Prospects,*

Saqib Ali. (2019). *Reading the ACF and PACF Plots - The Missing Manual / Cheatsheet*

Saunders, M. (2009). Research methods for business students. *Person Education Limited,* Schöniger, F., & Morawetz, U. B. (2022). What comes down must go up: Why fluctuating renewable energy does not necessarily increase electricity spot price variance in Europe. *Energy Economics, 111*, 106069. 10.1016/j.eneco.2022.106069

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics,* , 461–464.

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika, 52*(3-4), 591–611.

Shumway, R. H., Stoffer, D. S., & Stoffer, D. S. (2000). *Time series analysis and its applications*. Springer.

Spodniak, P., Ollikka, K., & Honkapuro, S. (2021). The impact of wind power and electricity demand on the relevance of different short-term electricity markets: The Nordic case. *Applied Energy, 283*, 116063.

Tsay, R. S. (2005a). *Analysis of financial time series*. John wiley & sons.

Tsay, R. S. (2005b). *Analysis of financial time series*. John wiley & sons.

Valitov, N. (2019). Risk premia in the German day-ahead electricity market revisited: The impact of negative prices. *Energy Economics, 82*, 70–77. 10.1016/j.eneco.2018.01.020

Wang, J., Zong, Y., You, S., & Træholt, C. (2017). A review of Danish integrated multi-energy system flexibility options for high wind power penetration. *Clean Energy, 1*(1), 23–35.

Wang, W., Chakraborty, G., & Chakraborty, B. (2020). Predicting the risk of chronic kidney disease (ckd) using machine learning algorithm. *Applied Sciences, 11*(1), 202.

Wang, Y., & Ni, X. S. (2019). A XGBoost risk model via feature selection and Bayesian hyper-parameter optimization. *arXiv Preprint arXiv:1901.08433,*

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research, 30*(1), 79–82.

Winther, C. D. (2022). *Visual Guide to the Power Grid: Inside the Greatest Machine in the World*. Visual Power Grid Company.

Wooldridge, J. M. (2012). Introductory econometrics: a modern approach (upper level economics titles). *Southwestern College Publishing, Nashville, T ATN, 41*, 673–690.

Xu, Y., Zheng, S., Zhu, Q., Wong, K., Wang, X., & Lin, Q. (2024). A complementary fused method using GRU and XGBoost models for long-term solar energy hourly forecasting. *Expert Systems with Applications, 254*, 124286. 10.1016/j.eswa.2024.124286

Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies, 58*, 308–324.

Zhou, S., Zhou, L., Mao, M., Tai, H., & Wan, Y. (2019). An optimized heterogeneous structure LSTM network for electricity price forecasting. *Ieee Access, 7*, 108161–108173.

12. Python Codes as Appendix

!pip install pandas statsmodels scikit-learn matplotlib openpyxl pmdarima xgboost scipy

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import datetime as dt
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from statsmodels.tsa.stattools import adfuller
from statsmodels.stats.diagnostic import acorr_ljungbox
from datetime import datetime, date, timedelta
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
import xgboost as xgb
```

Data Processing

```
# Load the dataset from the Excel file
file_path = 'dk1_20_24.xlsx'
df = pd.read_excel(file_path)
```

```
missing_values = df.isnull().sum()
print(missing_values)
df.info()
```

```
#   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   date                          40926 non-null   datetime64[ns]
 1   dk1                           40926 non-null   float64
 2   LocalPowerMWhDK1              40926 non-null   float64
 3   LocalPowerSelfConMWhDK1       40926 non-null   float64
 4   CentralPowerMWhDK1            40926 non-null   float64
 5   CommercialPowerMWhDK1         40926 non-null   float64
 6   HydroPowerMWhDK1              40926 non-null   float64
 7   OffshoreWindGe100MW_MWhDK1    40926 non-null   float64
 8   OffshoreWindLt100MW_MWhDK1    40926 non-null   float64
 9   OnshoreWindGe50kW_MWhDK1      40926 non-null   float64
 10  OnshoreWindLt50kW_MWhDK1      40926 non-null   float64
 11  SolarPowerGe10Lt40kW_MWhDK1   40926 non-null   float64
 12  SolarPowerGe40kW_MWhDK1       40926 non-null   float64
 13  SolarPowerLt10kW_MWhDK1       40926 non-null   float64
 14  SolarPowerSelfConMWhDK1       40926 non-null   float64
 15  PowerToHeatMWhDK1             40926 non-null   float64
 16  GrossConsumptionMWhDK1        40926 non-null   float64
```

```
# Convert the 'date' column to datetime format
df['date'] = pd.to_datetime(df['date'], format='%d/%m/%Y %H.%M.%S')
```

```
# Define the training and testing split point (80% for training, 20%
for testing)
split_ratio = 0.90
split_point = int(len(in_sample) * split_ratio)
```

```
Training Data Range: 2020-01-01 00:00:00 to 2024-03-03 14:00:00
Training Data Shape: (36639, 17)
Testing Data Range: 2024-03-03 15:00:00 to 2024-08-20 06:00:00
Testing Data Shape: (4071, 17)
```

```
# Define the target variable (y) and the exogenous variables (X) for
training and testing sets
y_train = train_data['dk1']
X_train = train_data[
    ['LocalPowerMWhDK1', 'LocalPowerSelfConMWhDK1',
'CentralPowerMWhDK1',
     'CommercialPowerMWhDK1', 'HydroPowerMWhDK1',
'OffshoreWindGe100MW_MWhDK1',
     'OffshoreWindLt100MW_MWhDK1', 'OnshoreWindGe50kW_MWhDK1',
'OnshoreWindLt50kW_MWhDK1',
     'SolarPowerGe10Lt40kW_MWhDK1', 'SolarPowerGe40kW_MWhDK1',
'SolarPowerLt10kW_MWhDK1',
     'SolarPowerSelfConMWhDK1', 'PowerToHeatMWhDK1',
     'GrossConsumptionMWhDK1']
]

y_test = test_data['dk1']
X_test = test_data[
    ['LocalPowerMWhDK1', 'LocalPowerSelfConMWhDK1',
'CentralPowerMWhDK1',
     'CommercialPowerMWhDK1', 'HydroPowerMWhDK1',
'OffshoreWindGe100MW_MWhDK1',
     'OffshoreWindLt100MW_MWhDK1', 'OnshoreWindGe50kW_MWhDK1',
'OnshoreWindLt50kW_MWhDK1',
     'SolarPowerGe10Lt40kW_MWhDK1', 'SolarPowerGe40kW_MWhDK1',
'SolarPowerLt10kW_MWhDK1',
     'SolarPowerSelfConMWhDK1', 'PowerToHeatMWhDK1',
     'GrossConsumptionMWhDK1']
]
```

```
# Example time series data (replace 'time_series_data' with your data)
result = adfuller(df['dk1'])
```

```
# Print ADF test results
print("ADF Statistic:", result[0])
print("p-value:", result[1])
print("Critical Values:", result[4])

# Interpretation
if result[1] <= 0.05:
    print("The series is stationary (reject null hypothesis).")
else:
    print("The series is non-stationary (fail to reject null
hypothesis).")



dk1_series = df['dk1']

# Perform the Ljung-Box test on the dk1 series
# Specify the lags you want to test (e.g., 10 lags)
lb_test = acorr_ljungbox(dk1_series, lags=[24], return_df=True)

# Print the Ljung-Box test results
print("Ljung-Box Test Results:")
print(lb_test)

# Interpretation
if all(lb_test['lb_pvalue'] > 0.05):
    print("The dk1 series are independently distributed (fail to reject
null hypothesis).")
else:
    print("The dk1 series exhibit autocorrelation (reject null
hypothesis).")
```

ARIMAX

```
from pmdarima import auto_arima

auto_arima_model = auto_arima(df['dk1'], exogenous=features,
                              seasonal=True, h=24, trace=True)

print(auto_arima_model.summary())
```

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=346207.171, Time=45.46 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=358502.073, Time=1.49 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=350812.154, Time=2.24 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=351673.877, Time=5.85 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=358500.073, Time=0.55 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=349879.694, Time=34.03 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=346785.680, Time=36.21 sec
 ARIMA(3,1,2)(0,0,0)[0] intercept   : AIC=346207.208, Time=110.02 sec
 ARIMA(2,1,3)(0,0,0)[0] intercept   : AIC=inf, Time=106.66 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=350714.757, Time=7.61 sec
```

```
 ARIMA(1,1,3)(0,0,0)[0] intercept   : AIC=350385.853, Time=45.46 sec
 ARIMA(3,1,1)(0,0,0)[0] intercept   : AIC=346281.378, Time=36.94 sec
 ARIMA(3,1,3)(0,0,0)[0] intercept   : AIC=346200.646, Time=77.30 sec
 ARIMA(4,1,3)(0,0,0)[0] intercept   : AIC=346199.791, Time=105.93 sec
 ARIMA(4,1,2)(0,0,0)[0] intercept   : AIC=346260.268, Time=72.59 sec
 ARIMA(5,1,3)(0,0,0)[0] intercept   : AIC=346203.994, Time=132.78 sec
 ARIMA(4,1,4)(0,0,0)[0] intercept   : AIC=inf, Time=145.12 sec
 ARIMA(3,1,4)(0,0,0)[0] intercept   : AIC=346204.411, Time=84.84 sec
 ARIMA(5,1,2)(0,0,0)[0] intercept   : AIC=346229.158, Time=75.07 sec
 ARIMA(5,1,4)(0,0,0)[0] intercept   : AIC=inf, Time=161.44 sec
 ARIMA(4,1,3)(0,0,0)[0]             : AIC=346197.794, Time=50.97 sec
 ARIMA(3,1,3)(0,0,0)[0]             : AIC=346198.648, Time=34.39 sec
 ARIMA(4,1,2)(0,0,0)[0]             : AIC=346258.270, Time=32.08 sec
 ARIMA(5,1,3)(0,0,0)[0]             : AIC=346197.787, Time=57.14 sec
 ARIMA(5,1,2)(0,0,0)[0]             : AIC=346227.160, Time=28.01 sec
 ARIMA(5,1,4)(0,0,0)[0]             : AIC=inf, Time=69.94 sec
 ARIMA(4,1,4)(0,0,0)[0]             : AIC=inf, Time=61.55 sec
```

```
Best model:  ARIMA(5,1,3)(0,0,0)[0]
Total fit time: 1621.738 seconds
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                40926
Model:               SARIMAX(5, 1, 3)   Log Likelihood             -173089.894
Date:                Fri, 15 Nov 2024   AIC                         346197.787
Time:                        16:01:43   BIC                         346275.363
Sample:                             0   HQIC                        346222.316
                              - 40926
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.9620      0.125      7.679      0.000       0.716       1.207
ar.L2          0.0666      0.216      0.309      0.757      -0.356       0.489
ar.L3         -0.3301      0.136     -2.428      0.015      -0.597      -0.064
ar.L4          0.0129      0.024      0.534      0.593      -0.034       0.060
ar.L5         -0.0279      0.009     -3.247      0.001      -0.045      -0.011
ma.L1         -0.6238      0.125     -4.978      0.000      -0.869      -0.378
ma.L2         -0.3292      0.176     -1.875      0.061      -0.673       0.015
ma.L3          0.1273      0.075      1.690      0.091      -0.020       0.275
sigma2       277.3572      0.606    457.841      0.000     276.170     278.545
===================================================================================
Ljung-Box (L1) (Q):                   0.73   Jarque-Bera (JB):            825003.49
Prob(Q):                              0.39   Prob(JB):                         0.00
Heteroskedasticity (H):               5.61   Skew:                             0.29
Prob(H) (two-sided):                  0.00   Kurtosis:                        24.99
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients
(complex-step).
```

```
# Define the SARIMAX model for ARIMAX
model_arimax = SARIMAX(y_train,
                exog=X_train,
                order=(5, 1, 3),  # Start with a basic order, but this
can be fine-tuned
                seasonal_order=(0, 0, 0, 0),  # No seasonality
specified here
                enforce_stationarity=False,
                enforce_invertibility=False)

# Fit the model
arimax_model = model_arimax.fit(disp=False)
print(arimax_model.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                    dk1   No. Observations:                36639
Model:               SARIMAX(5, 1, 3)   Log Likelihood             -152713.724
Date:                Sun, 17 Nov 2024   AIC                         305475.447
Time:                        11:07:15   BIC                         305679.656
Sample:                             0   HQIC                        305540.353
                              - 36639
Covariance Type:                  opg
===============================================================================================
                                  coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------------
LocalPowerMWhDK1                0.0536      0.002     21.799      0.000       0.049       0.058
LocalPowerSelfConMWhDK1        -0.0726      0.026     -2.818      0.005      -0.123      -0.022
CentralPowerMWhDK1              0.0084      0.001      9.160      0.000       0.007       0.010
CommercialPowerMWhDK1          -0.1506      0.019     -8.022      0.000      -0.187      -0.114
HydroPowerMWhDK1                2.2526      1.057      2.131      0.033       0.181       4.324
OffshoreWindGe100MW_MWhDK1     -0.0075      0.001     -6.338      0.000      -0.010      -0.005
OffshoreWindLt100MW_MWhDK1     -0.0126      0.005     -2.638      0.008      -0.022      -0.003
OnshoreWindGe50kW_MWhDK1       -0.0118      0.001    -15.564      0.000      -0.013      -0.010
OnshoreWindLt50kW_MWhDK1       -0.1502      0.053     -2.825      0.005      -0.254      -0.046
SolarPowerGe10Lt40kW_MWhDK1     1.3187      0.242      5.459      0.000       0.845       1.792
SolarPowerGe40kW_MWhDK1        -0.0207      0.003     -8.087      0.000      -0.026      -0.016
SolarPowerLt10kW_MWhDK1        -0.3856      0.019    -20.102      0.000      -0.423      -0.348
SolarPowerSelfConMWhDK1        -0.0615      0.009     -6.811      0.000      -0.079      -0.044
PowerToHeatMWhDK1              -0.0662      0.002    -36.760      0.000      -0.070      -0.063
GrossConsumptionMWhDK1          0.0465      0.001     53.927      0.000       0.045       0.048
ar.L1                           0.3225      0.092      3.497      0.000       0.142       0.503
ar.L2                           0.0931      0.105      0.888      0.375      -0.112       0.299
ar.L3                           0.3443      0.085      4.070      0.000       0.179       0.510
ar.L4                          -0.2013      0.024     -8.543      0.000      -0.248      -0.155
ar.L5                          -0.0726      0.007    -10.000      0.000      -0.087      -0.058
ma.L1                          -0.2997      0.093     -3.220      0.001      -0.482      -0.117
ma.L2                          -0.3121      0.104     -3.012      0.003      -0.515      -0.109
ma.L3                          -0.5941      0.088     -6.746      0.000      -0.767      -0.421
sigma2                        214.5655      1.144    187.563      0.000     212.323     216.808
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):            935258.55
Prob(Q):                              0.93   Prob(JB):                         0.00
Heteroskedasticity (H):               4.18   Skew:                             0.44
Prob(H) (two-sided):                  0.00   Kurtosis:                        27.74
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

```
# Define a function to calculate performance metrics
def evaluate_performance(y_true, y_pred):
    mae = mean_absolute_error(y_true, y_pred)
    mse = mean_squared_error(y_true, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_true, y_pred)
    return mae, mse, rmse, r2
# Training performance
mae_train, mse_train, rmse_train, r2_train =
evaluate_performance(y_train, y_train_pred)
print("Training - MAE:", mae_train, "MSE:", mse_train, "RMSE:",
rmse_train, "R2:", r2_train)

# Testing performance
mae_test, mse_test, rmse_test, r2_test = evaluate_performance(y_test,
y_test_pred)
print("Testing - MAE:", mae_test, "MSE:", mse_test, "RMSE:", rmse_test,
"R2:", r2_test)
```

XGBoost

```
import xgboost as xgb
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error

# Create DMatrix for cross-validation
#X_dmatrix = xgb.DMatrix(X_train, label=y_train)

# Parameter grid for tuning
#param_grid = {
    'max_depth': [3, 6, 9],           # Tree depth
    'eta': [0.01, 0.1, 0.3],          # Learning rate
    'min_child_weight': [1, 3, 5],    # Minimum child weight
    'subsample': [0.8, 1],            # Subsampling ratio
    'colsample_bytree': [0.6, 0.8, 1], # Fraction of features to use
    'gamma': [0, 1, 5],               # Minimum loss reduction
    'n_estimators': [50, 100, 200]    # Number of boosting rounds
}

# Perform Grid Search with 5-fold CV
#grid_cv = GridSearchCV(
    estimator=xgb.XGBRegressor(objective='reg:squarederror'),
    param_grid=param_grid,
    scoring='neg_mean_squared_error',
    cv=5,
    verbose=1,
    n_jobs=-1
)

# Fit Grid Search to find best parameters
#grid_cv.fit(X_train, y_train)

# Output the best parameters
#best_params = grid_cv.best_params_
#print("Optimal Parameters:", best_params)



# Import necessary modules
import xgboost as xgb
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score

# Create the final XGBoost model using optimal parameters
model_xgb = xgb.XGBRegressor(
    objective='reg:squarederror',
    colsample_bytree=0.6,
    eta=0.1,
```

```
    gamma=5,
    max_depth=6,
    min_child_weight=5,
    n_estimators=200,
    subsample=1
)


# Train the model
model_xgb.fit(X_train, y_train)
# Predict on training data
y_train_pred_xgb = model_xgb.predict(X_train)

# Predict on testing data
y_test_pred_xgb = model_xgb.predict(X_test)


# Training performance
train_mae = mean_absolute_error(y_train, y_train_pred_xgb)
train_mse = mean_squared_error(y_train, y_train_pred_xgb)
train_rmse = train_mse ** 0.5
train_r2 = r2_score(y_train, y_train_pred_xgb)

print("Training Performance:")
print(f"MAE: {train_mae}, MSE: {train_mse}, RMSE: {train_rmse}, R²:
{train_r2}")

# Testing performance
test_mae = mean_absolute_error(y_test, y_test_pred_xgb)
test_mse = mean_squared_error(y_test, y_test_pred_xgb)
test_rmse = test_mse ** 0.5
test_r2 = r2_score(y_test, y_test_pred_xgb)

print("Testing Performance:")
print(f"MAE: {test_mae}, MSE: {test_mse}, RMSE: {test_rmse}, R²:
{test_r2}")
```

Defining and ploting Residuals

```
# For ARIMAX model
residuals_arimax = y_test - y_test_pred

# For XGBoost model
residuals_xgb = y_test - y_test_pred_xgb
```

- END-