Universität Rostock

Traditio et Innovatio

Data Science Challenge 2024

Predicting Day-ahead electricity prices

Group Name:	CTRL + C Elites
name:	Nafees Mohammad Adil, MD Humayun Kabir, Syed Hamza Abbas Naqvi, Abdus Samad
DozentIn:	Dr.-Ing. Hannes Grunert, M.Sc
Institut:	University of Rostock
Fakultät:	Computer Science International
Modul:	Project Masters Computer Science and NEIDI
Fachsemester:	Winter 24/25
Abgabedatum:	13th December 2024

Wissenschaftliche Studienarbeit

Universität Rostock

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Abstract

This project focuses on developing a machine-learning model to predict day-ahead electricity prices in the German market. We investigate the impact of various factors, including historical price data, weather conditions, and electricity generation and consumption. A comprehensive data analysis is conducted to identify relevant features and potential correlations. The datasets used also contained missing data which we have successfully processed. The dataset includes electricity generation, consumption and price features from SMARD.DE and we have also included weather features from home.openweathermap.org. After preprocessing of the dataset, we have trained 2 models and compared the accuracy between them and chose the best performing model to predict the hourly electricity price for 18.Feb.2025.

2 Introduction

The energy market is undergoing significant transformation, leading to increased volatility in electricity prices. Accurate day-ahead price forecasting is crucial for energy producers, traders, and consumers. This research aims to develop a robust machine-learning model to predict German electricity prices. By leveraging historical data, weather information, and other relevant factors, we aim to improve the accuracy and reliability of price forecasts, enabling better decision-making and risk management in the energy market.

3 Phase 1: **Gathering Domain Knowledge & Data Sources**

At the beginning of the project, we have read relevant research papers and explored other sources to gain more insights in the German electricity market. Our target was to explore and find out the relationships and factors that effect the price of electricity of Germany. We have visited Stadtwerke Rostock AG in Schmarl Rostock, which is the regional energy service provider for the coastal area, offering services such as electricity, natural gas, and heating. Their headquarters are located at Schmarler Damm 5, 18069 Rostock, in the Schmarl district. We have had a great discussion with the representatives at Stadtwerke Rostock AG where they have answered questions about the German electricity market. We have summarized our findings below:

3.1 Analysis of German Electricity Market

Germany's energy market is a key player in the global shift toward renewable energy. This transition, known as the Energiewende, is about more than just switching to greener energy sources—it's about balancing climate goals with energy security and economic stability. However, moving away from conventional energy sources hasn't been without its challenges. This review examines insights from three studies, referred to as [1], [2], and [3], to explore the market's strengths, gaps, and the strategies needed to navigate this transformation.

3.2 Overview of the German Electricity Market

Germany is heavily focused on integrating renewable energy sources (RES) like wind and solar into its electricity grid. [1] dives deep into this topic, analyzing how renewables have changed

electricity pricing dynamics. Meanwhile, [2] looks at how these changes affect households, municipalities, and traditional energy producers. In contrast,[3] takes a broader view, imagining how Germany's energy market might evolve by 2035, depending on various political, economic, and technological factors. Together, these studies provide a snapshot of a market that is dynamic, forward-thinking, but also facing some significant hurdles.

3.3 Market Challenges and Shortcomings

1. **Managing Renewables Integration**

   The biggest issue with renewables is their variability. Wind doesn't always blow, and the sun doesn't always shine. This makes it harder to keep the grid stable. [1] explains how this intermittency leads to periods of overproduction or shortages, which strain the grid. Adding to this, the "merit order effect"—where renewable energy's near-zero marginal cost lowers market prices—sounds good for consumers but creates financial headaches for conventional power producers.

   [2] also indicates a second issue: as wholesale prices of electricity reduce, retail prices are artificially raised by charges and taxes like the EEG Umlage. It is a constraint on consumers as they actually subsidize the transformation of energy. [3] also adds that geopolitical risks such as dependence on Russian gas simply make it tougher to realize Germany's renewable energy targets.

2. **Forecasting and Price Volatility**

   Predicting renewable energy output isn't easy. [1] shares data showing forecast errors of around 2.2%, making it difficult to match supply with demand. This unpredictability forces energy markets to rely on backup systems, driving up costs and complicating operations. Although [2] doesn't focus on forecasting, it argues for decentralized energy systems, where local production could reduce these issues. Decentralization could help stabilize prices and offer better control at the regional level.

3. **Uneven Impacts on Stakeholders**

   Not everyone feels the effects of the energy transition equally. [2] paints a picture of traditional energy producers struggling to adapt, as renewables take up a larger market share. Municipalities also face challenges—they could become key players in a decentralized system, but they lack the resources and financial incentives to take on such a role. On the other hand, [3] looks at Germany's global energy strategy, warning that a failure to act decisively could prolong dependency on imported energy, further complicating the market's transition.

3.4 The Effect on Market Prices

One area where all three studies align is how renewables influence electricity prices:

- [1]shows that increasing renewable energy tends to push wholesale prices down, thanks to low operational costs.

- However, [2] emphasizes that retail prices don't reflect these savings. Instead, taxes and surcharges drive household costs higher, making the energy transition less consumer-friendly.
- [3] warns that if Germany doesn't address these pricing issues, it risks price volatility and lost investor confidence, which could slow renewable energy adoption.

3.5 Strategies for Improvement

1. **Smarter Grids and Better Forecasting**

   One clear solution is improving grid management. [1] highlights the importance of advanced forecasting tools and demand-side management to handle renewable variability. Smarter grids, coupled with real-time data analysis, could optimize supply and demand balance.

   Decentralized energy systems are another promising option. [2] argues that municipalities could play a larger role in producing and distributing energy locally. This would reduce dependency on the national grid and offer communities more control over their energy needs.

2. **Reforming Policies and Incentives**

   Regulatory reform is key to a smoother transition. [2] suggests that revising the retail pricing structure to reduce household burdens could make the energy shift more equitable. This includes tweaking taxes and surcharges so that costs are distributed more fairly across stakeholders.

   [3] takes a broader approach, proposing policies that encourage long-term investments in renewable energy infrastructure. For example, offering financial incentives for energy storage technologies could help stabilize the grid and make renewables more reliable.

3. **Planning for the Future**

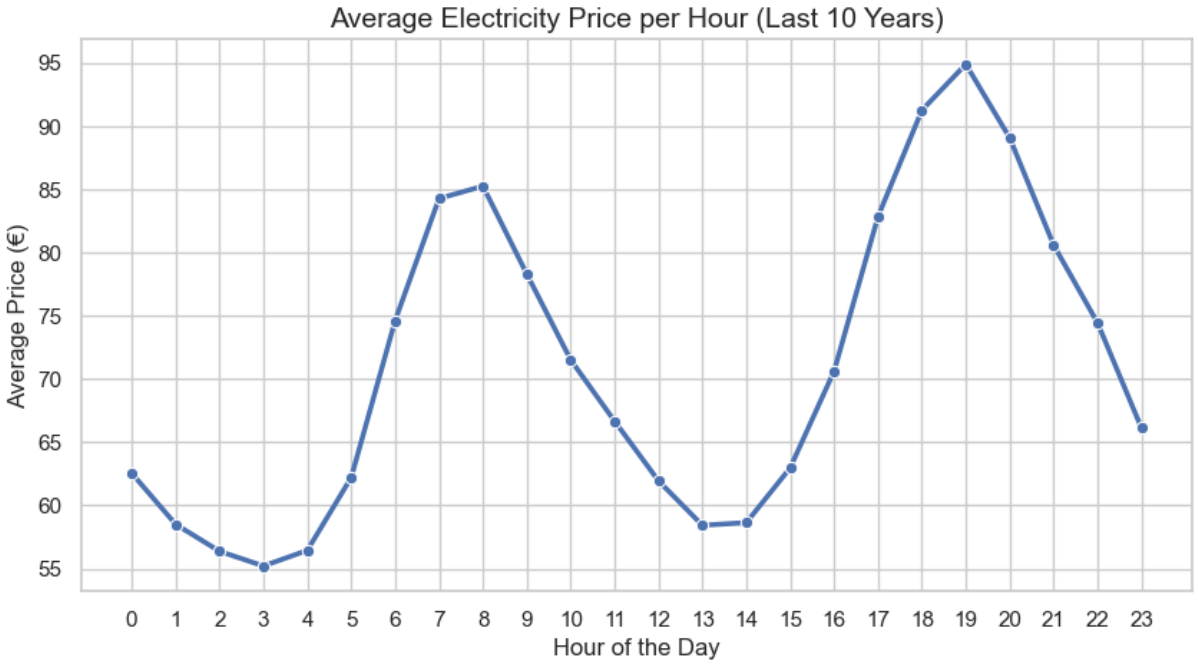
   One of the most forward-thinking strategies comes from [3], which uses scenario planning to prepare for multiple possible futures. By considering scenarios like "Green Growth"(a rapid transition to renewables) or "Missed Opportunity"(slow progress due to weak policies), decision-makers can craft flexible strategies to handle uncertainties.

4 Phase 2: Visual Story Telling

4.0.1 Introduction

The German electricity market operates on a dynamic pricing model influenced by energy supply and demand, renewable energy penetration, and market regulations. This analysis explores the insights derived from five different graphs, covering energy generation trends, electricity prices, and their temporal variations.

4.1 Hourly Average Electricity Price Trends in Germany: A 10-Year Analysis:



This chart is showing average hourly electricity price for the last 10 years in the German electricity market. It gives information regarding the fluctuation of the price of electricity during 24 hours. Following are some of the interesting observations:

Key Insights:

The cheapest hours are early morning hours (2 AM to 5 AM), quite possibly because there is less electricity demand. Prices begin to increase steeply after 6 AM and reach a peak in the 8-9 AM range, which coincides with the morning peak in demand when industries and companies begin to open up for business.

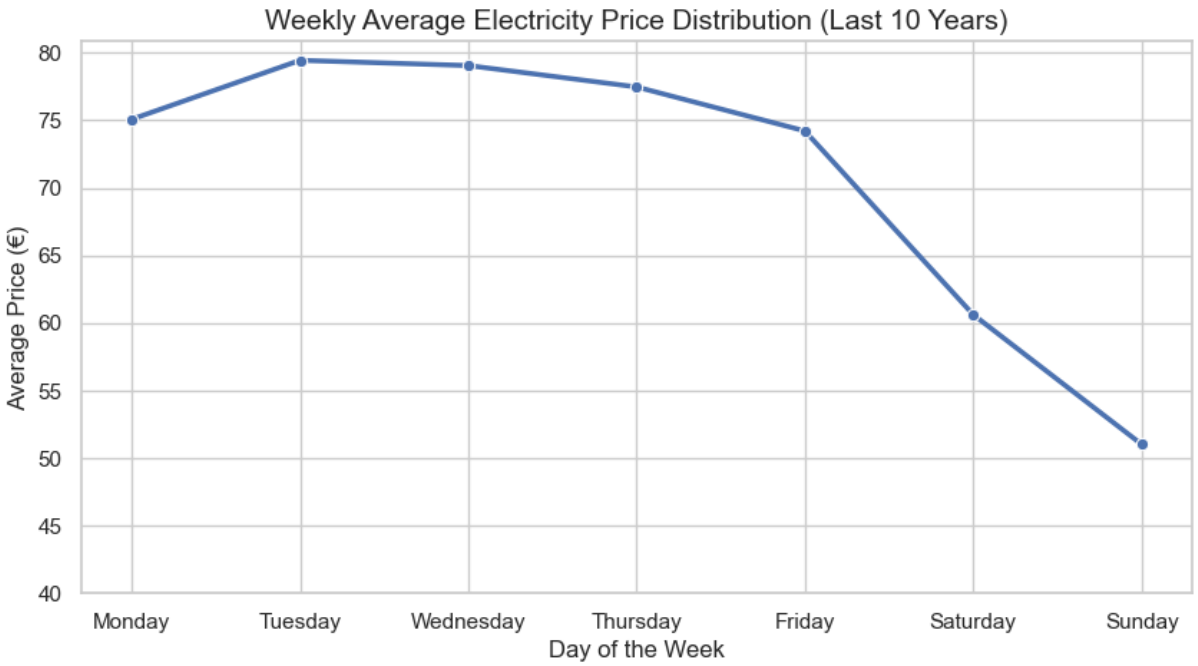
Prices decline from late morning (10 AM) to early afternoon (2 PM), perhaps because of reduced industrial demand and increased solar power generation.

Prices are relatively flat in the early afternoon before suddenly increasing from 4 PM to 7 PM and reaching a peak at around 7-8 PM. This is an evening peak when individuals return home and electricity consumption is at its highest. Prices start decreasing from 8 PM and keep on decreasing slowly until the early morning hours.

Market Implications:

Morning and evening peaks show peak hours of demand when electricity is highest priced. The midday decline in electricity prices shows the role of renewable sources of energy, mostly solar energy, that reduces the utilization of expensive fossil-fuel-based electricity. Late evening and morning periods are the cheapest, as one would expect during decreased demand.

4.2 Weekly Average Electricity Price Distribution in Germany Over the Last Decade:



This chart indicates the weekly average price distribution of electricity in the German electricity market for the last 10 years. It indicates the prices of electricity on various days of the week. Some of the findings are as follows:

Key Insights:

Peak Prices during Weekdays (Monday–Thursday):
The prices of electricity are highest from Monday to Thursday, and the highest prices on Tuesday and Wednesday are around 80 €/MWh. This is due to the fact that the trend experiences more industrial and commercial usage of electricity during weekdays that increases the demand and the rate too.

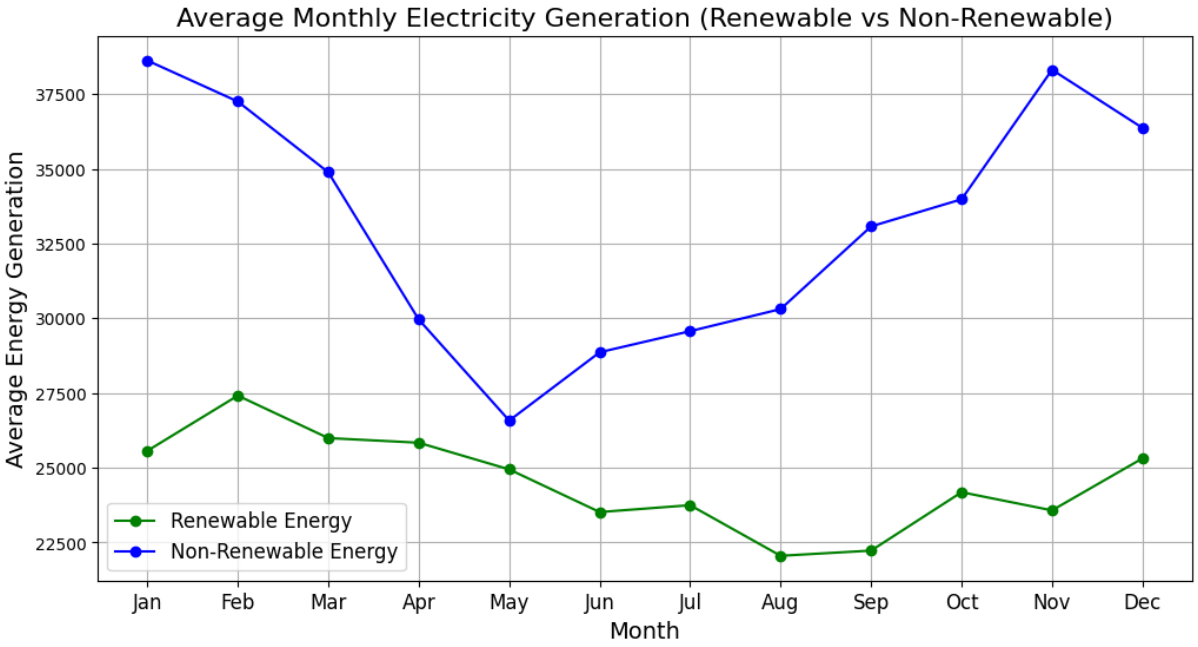
Drop on Friday:
The rate begins decreasing from Friday when the weekend shift comes when various industries reduce their manufacturing activities.

Sharp Decline on Weekends (Saturday Sunday):
Prices of electricity go down significantly on Saturday and reach its lowest point on Sunday (55 €/MWh). This is due to electricity demand being derived from only residential and basic services.

Market Implications:
Higher weekday prices reflect industrial and commercial utilization of energy determines electricity demand. Lower weekend prices reflect less economic activities with the capacity to use more of the renewable sources such as wind and sun, which have a tendency to provide more to

the grid during low-demand days.

4.3 Seasonal Trends in Renewable vs. Non-Renewable Electricity Generation in Germany:



This line graph shows the 10 years average monthly German electricity trend from renewable and non-renewable sources, which indicates some seasonal fluctuation in production.

Non-Renewable Energy (Blue Line):

Highest generation is in winter, with a peak in January (38,000 MWh) and November (39,000 MWh), likely due to higher heating loads and lower generation from renewables. Lowest generation is in summer, with the trough in May (26,000 MWh), when solar and wind contribute more. There is a secondary peak in September, at seasonal peak demand.

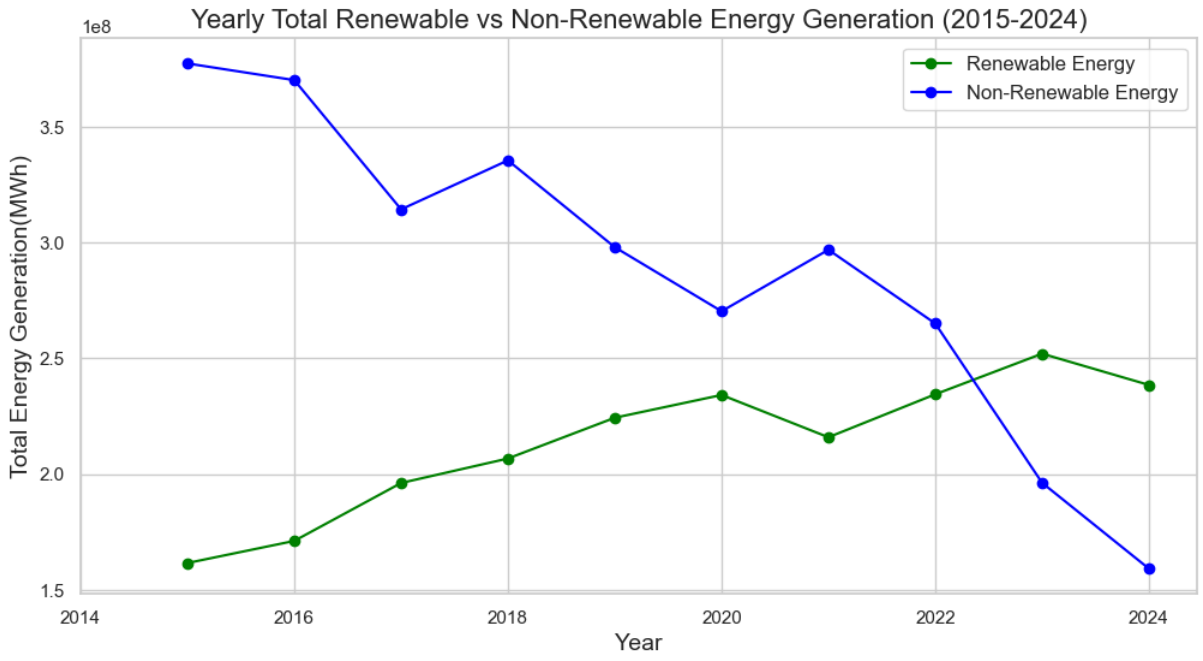
Renewable Energy (Green Line):

It is the highest in February (27,500 MWh), perhaps due to maximum wind power generation. The generation drops from March with minimum in August (22,000 MWh), perhaps because of lesser wind power and absence of hydro generation. The growth is moderate in September to December (25,500 MWh in December), perhaps because of greater wind generation.

Market Implications:

Germany is more fossil fuel dependent during winter because demand is greater and availability of renewables is lesser. Renewable energy is highest at the end of winter and spring but lowest during summer, so solar power cannot replace seasonality of loss of wind and hydro assets.

4.4 Germany's Shift to Renewable Energy: Yearly Electricity Production (2015-2024)



This line chart represents Germany's total year-to-year production of electricity from non-renewable and renewable sources of energy between the years 2015 and 2024, reflecting a change in the energy patterns of the nation.

Non-Renewable Energy (Blue Line):

Non-renewable generation of power declined steadily year on year. It was quite high between 2015 and 2017, with a decrease. There is a steep fall-off from the year 2018 onwards and the greatest post-2021 decline, a record low for the year 2024. This is consistent with Germany dropping coal and nuclear power generation to favor green energy.

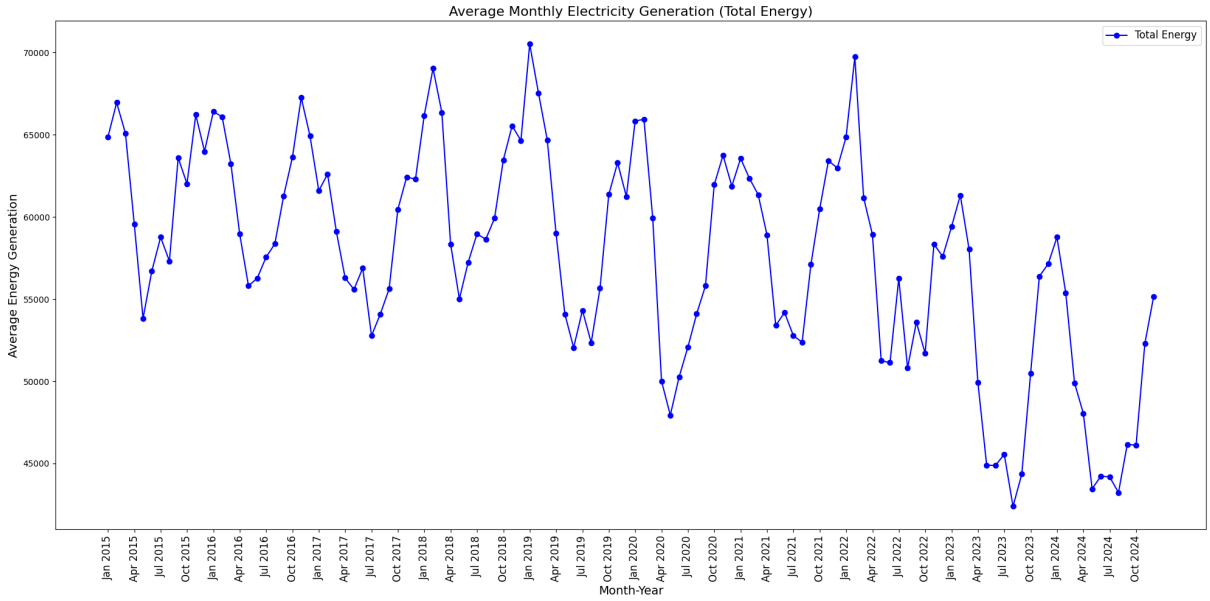
Renewable Energy (Green Line):

Renewable energy generation has been steadily increasing since 2015. There is a sharp rising trend up to 2020 and slight deviations afterward. Renewables beat non-renewables in 2023, a significant breakthrough for Germany's clean energy transition. The modest dip in 2024 may be due to seasonal variations or short-term deficit in supply.

Market Implications:

Germany is de-fossilizing, and that has been indicated by non-fossil fuel production declines. Transition to renewables has been realized but intermittently supplied output is a pointer to the need for grid stabilization as well as storage infrastructure development. The data indicates Germany's successful transformation to energy with it becoming an energy leader with clean energy as it emphasizes that there is the necessity for the continued innovations within energy infrastructure.

4.5 Seasonal Variations and Trends in Germany's Total Power Generation Over 10 Years:



This line chart shows German average monthly power generation for 10 years, trend and seasonality in total energy generation.

Key Insights:

Seasonal Pattern:

December to February months witness peak electricity generation. This can be because there are greater energy requirements for warming up.
Production reduces considerably during summer seasons (May-August), and possibly because of lower heating requirements and peak requirements for clean sources of energy like sun and wind power.

Repeating Peaks and troughs:

There is a pattern in the graph, where electricity generation rises towards the end of the year and drops in the middle.
Sharp reductions in certain years, particularly 2020 and 2023, result from economic slumps, policy shocks, or shifts in energy sources.

Greater Volatility:

Volatility is reported to have increased in more recent times due to variation in the supply of energy. Because of an increase in the portion of renewable power, which relies on climatic conditions, or volatility of market dynamics.

Market Implications:

More winter electric generation implies depending more on fossil energy and other energy re-

sources that are non-renewable during rare shortages of renewables.
Lower summer output may result in lower reliance on conventional energy sources supplemented by sun and wind.
This result again points towards the seasonally fluctuating and dynamic nature of the German power market and reflects the exploitation of renewables as well as smart energy planning.

4.6 Dataset Overview

The dataset used for this analysis spans from March 11, 2024, to March 17, 2024. It contains hourly data on electricity generation, consumption, and the day-ahead price from the website smard.de, these datasets contain relevant features such as grid load, day of the week, and time-based variables. This dataset was used to predict the day-ahead electricity price for each hour within the range.

4.6.1 Feature Selection

The features selected for the model include both energy generation sources and time-related factors. These features were chosen based on their potential impact on electricity pricing. Below is the list of features:

5 Phase 3: Data Cleaning Exploratory Data Analysis

5.0.1 The Dataset

After we have gathered ample knowledge of the German electricity market, we have decided to to use historical data of ten years ranging from 2015 to 2024 from SMARD.de and home.openweathermap.org to train our model. The datasets were in the form of hourly frequency and Our target was to find relevant features that could be used to train our model. After complete analysis of the datasets in the websites, we have discovered that the “Actual Electricity Generation” and “Actual Electricity Consumption” datasets would be the right choice from SMARD.DE.

Actual Electricity Generation: The actual generation corresponds to the net generation. The net generation is the electricity fed in minus the electricity required by the generation units themselves. The actual generation only includes electricity fed into the general supply network.

This dataset included electricity generation data in the following energy sources as individual columns in .xslx format and these were also used as features :

1. Nuclear
2. Biomass
3. Hydropower
4. Wind offshore
5. Wind onshore
6. Photovoltaics

7. Lignite
8. Hard coal
9. Natural gas
10. Hydro pumped storage
11. Other conventional energy sources

Actual Consumption:

The actual consumption on the SMARD website corresponds to the total load (including network losses without stored energy). This is the net generation minus export transmission capacity, plus import transmission capacity and minus the feed-in capacity from hydro-pumped storage power stations. The net generation does not include the electricity required by conventional power plants themselves, electricity fed in within industrial networks and closed distribution networks, or electricity fed into the Deutsche Bahn network. The net generation may include a forecast component if no data on actual generation are available.

The features used in actual consumption are:

1. Total grid load
2. Residual load
3. Hydro pumped storage

Weather Data

To enhance the accuracy of our predictions, we have incorporated weather data, as mentioned earlier. Weather conditions significantly influence electricity prices, making it essential to include these features for precise hourly price forecasting. We sourced high-quality hourly weather data from OpenWeather, which we seamlessly integrated with the datasets from SMARD. The weather features include:

1. Temperature
2. dew point
3. feels like
4. temp min
5. temp max
6. Pressure
7. Humidity
8. Windspeed
9. Wind deg
10. Wind gust
11. Snow
12. Clouds

5.0.2 Problems faced and Solutions implemented during data processing.

5.0.3 Downloading the Dataset in the Correct Format

One of the initial challenges we faced was downloading the dataset in the correct format. Initially, we obtained the dataset in .CSV format from SMARD.DE. However, the model could not be trained because the data structure was not readable by the model. The issue stemmed from formatting inconsistencies and incorrect data parsing. The problem was resolved by converting the dataset into an .XLSX file format, ensuring proper data structure and readability.

5.0.4 Merging Data Files into One

For the model to be trained effectively, we needed to merge multiple data files into a single dataset, incorporating all relevant features along with the target variable. During this process, we discovered that the rows from different files were not aligning correctly. Specifically, timestamps in different datasets did not match, leading to missing or misaligned data points. To resolve this, we implemented a structured approach:

- **Standardizing Timestamps** – We ensured that all datasets followed a consistent date and time format.
- **Matching Records** – Each record was aligned based on the exact date and hour across all datasets.
- **Handling Missing Values** – Any gaps in the data were either interpolated or removed based on relevance to maintain data integrity.

5.0.5 Handling missing values

Some columns in the dataset had a significant amount of missing data. For example, the hourly electricity price, which we used as the target variable for training our model, was missing from January 1, 2015, 12:00 AM, to September 30, 2018, 11:00 PM. To address this issue, we searched for alternative data sources to fill in the missing values. Additionally, there were numerous missing values in the electricity generation and consumption columns. In the dataset, these missing values were represented by the ßtring, which made direct imputation difficult since no reliable external source was available to fill them. To handle this, we: Replaced invalid placeholders (") with NaN to ensure proper numerical processing. Used forward fill (ffill) and backward fill (bfill) in Pandas to propagate known values and fill the missing data effectively. This approach ensured data consistency while maintaining the integrity of historical trends.

5.0.6 Removing unnecessary columns

The dataset contained many columns that were not used as features or targets, such as electricity prices from other European countries. While some tests showed promising results when incorporating electricity prices from countries Germany trades with, we found that this approach led to overfitting. Implementing these additional features effectively would require more time and further experimentation, which could potentially improve the model's performance. Some

columns in the weather datasets were also omitted as it contained a lot of unnecessary columns that weren't as useful in training our model.

6 Phase 4: Predictive Modeling And Model Performance

The dataset used for training the model spans from January 1, 2015, to December 7, 2024. It contains hourly data on electricity generation, consumption, and day-ahead price from the website smard.de and weather data from home.openweathermap.org. These datasets contain relevant features such as grid load, day of the week, and time-based variables required to train the model. Our model uses the dataset to train itself, then predicts the price of the electricity of any date we wish. In order for our model to predict the price of a particular date, we have to provide the model with all the feature data of the target day we want to predict the price of electricity. We can collect all this data from SMARD.de in the forecasted electricity generation and consumption tab, and weather data from openweather.org. We have to input all those feature data to a .xslx file named "Manual_Features.xslx". It contains data for electricity generation, consumption, and weather for the day we want to predict prices. Currently, this file must be prepared manually with 24 rows (one for each hour of the day).

6.1 Predictive Model Comparison

At the start of the predictive modelling stage, we looked into linear regression model because of its simplicity and ease of use, but unfortunately it couldn't handle the complex data relationships, imputing empty data cells in the dataset and the redundant data, therefore we moved to 2 other models for this project which are "Random Forest" and "XGBoost". Initially we stuck with Random Forest but after training and tuning the hyperparameters it was not giving good enough accuracies therefore we switched to the XgBoost model.

- Random Forest: Random Forest model is a collection of decision trees that work together to make predictions. (https://www.geeksforgeeks.org/)

  To predict the electricity prices, at the beginning we used the Random Forest model. It was chosen for its simplicity, interpretability, and ease of use. But the prediction result generated from Random Forest was not reasonable and accurate. Also, there were many challenges to process large amounts of numerical non-linear data. Model performance is evaluated using:

  - Mean Absolute Error (MAE) : Measures average prediction error.
  - Mean Squared Error (MSE) : Measures average squared error.
  - Root Mean Squared Error (RMSE) : Penalizes larger errors more heavily.
  - R-squared (R^2) : Indicates the proportion of variance explained by the model.

- XGBRegressor: XGBRegressor is a gradient-boosting algorithm that builds decision trees sequentially, with each tree correcting the errors of the previous ones. XGBRegressor is perfect for higher predictive accuracy from complex, non-linear relationship datasets. After

running this model, we found that XGBRegressor provides more accurate results than Random Forest. Also, this model is perfect for numerical data process, performance, control overfitting, handling missing data, etc.

Metric	Random Forest	XGB Regressor
MAE	65.6	16.88
MSE	7191.87	642.51
RMSE	84.80	25.34
R^2	-1.19	0.708

Tabelle 1: Performance comparison of XGB Regressor and Random Forest

6.2 Code Implementation

We have created an instruction file that shows a step by step process on how to set up the environment and the libraries needed to run our code. Mentioned below are some of the explanations of the major part of our code.

6.2.1 Importing Dataset

The first step of our program is to import the dataset into our code base:

Listing 1: Feature Importance in XGBoost

```
weather_data = pd.read_excel(data_dir + 'Weather_Data_2015_2024-to-dec-7--11PM.xlsx')
generation_data = pd.read_excel(data_dir + 'Actual-Generation-2015-2024-to-dec-7--11PM.xlsx')
consumption_data = pd.read_excel(data_dir + 'Actual-Consumption-2015-2024-to-dec-7--11PM.xlsx')
price_data = pd.read_excel(data_dir + 'Price_2015_2024-to-dec-7--11_ADDED-MISSING-PRICES-PM.xlsx')
```

after importing the datasets, we have done some pre-processing such as converting to a date/time format where the model could understand the timestamps.

6.2.2 Merging Data

After data pre-processing, we have combined all feature data into a central file. Before that, we are processing the data again to ensure that the final data is completely perfect for training. In order to merge all the data-files into one, we needed to use the 'Start Date' column as a common reference column, which was available in all the data files. We have also replaced any invalid strings in the columns with NaN and forward fill.

Listing 2: Merge datasets based on timestamp

```
# Merge datasets based on timestamp
```

```
merged_data = generation_data.merge(price_data, on='Start date', how='inner')
merged_data = merged_data.merge(consumption_data, on='Start date', how='inner')
merged_data = merged_data.merge(weather_data, left_on='Start date', right_on='Start date', how='left')

# Handle missing data
merged_data.dropna(axis=1, how='all', inplace=True)
merged_data.fillna(method='ffill', inplace=True)

# Replace invalid strings with NaN and forward fill
merged_data.replace('-', np.nan, inplace=True)
merged_data.fillna(method='ffill', inplace=True)

# Save merged data
merged_file_path = '/home/Desktop/DS/cleaned_merged_data.xlsx'
merged_data.to_excel(merged_file_path, index=False)
```

6.2.3 Feature Engineering

Along with our static features which we got from the datasets, we have also engineered some extra features to help the model get higher accuracy. the features includes Lag features and rolling averages for price and temperature for the last 24, 48 and 72 hours.

Listing 3: Handle price lag features dynamically based on available data

```
# Handle price lag features dynamically based on available data
last_known_prices = merged_data['Germany/Luxembourg [ /MWh]'].iloc[-72:]
num_predictions = len(manual_features_df)

if len(last_known_prices) >= 24:
    manual_features_df['price_lag_24'] = (last_known_prices.iloc[-24:].values.tolist() * ((num_predictions + 23) // 24))[:num_predictions]
else:
    manual_features_df['price_lag_24'] = last_known_prices.values.tolist() * ((num_predictions + len(last_known_prices) - 1) // len(last_known_prices))
    manual_features_df['price_lag_24'] = manual_features_df['price_lag_24'][:num_predictions]

if len(last_known_prices) >= 48:
```

```
    manual_features_df['price_lag_48'] = (last_known_prices.iloc
        [-48:-24].values.tolist() * ((num_predictions + 23) // 24))[:
        num_predictions]
else:
    manual_features_df['price_lag_48'] = (last_known_prices.iloc[-min
        (48, len(last_known_prices)):].values.tolist() * ((
        num_predictions + 23) // min(48, len(last_known_prices))))[:
        num_predictions]

if len(last_known_prices) >= 72:
    manual_features_df['price_lag_72'] = (last_known_prices.iloc
        [-72:-48].values.tolist() * ((num_predictions + 23) // 24))[:
        num_predictions]
else:
    manual_features_df['price_lag_72'] = (last_known_prices.iloc[-min
        (72, len(last_known_prices)):].values.tolist() * ((
        num_predictions + 23) // min(72, len(last_known_prices))))[:
        num_predictions]


# Calculate rolling temperature features
# Here we use historical averages for simplicity
if 'temp' in merged_data.columns:
    # Ensure temp column exists in merged_data
    historical_temperatures = merged_data['temp'].resample('D').mean()
        .iloc[-72:]
    if 'temp' in manual_features_df.columns:
        manual_features_df['temp'] = historical_temperatures.iloc[-1].
            repeat(len(manual_features_df))

    for window in [24, 48, 72]:
        if 'temp' in manual_features_df.columns:
            manual_features_df[f'temp_roll_{window}'] =
                manual_features_df['temp'].rolling(window=window,
                min_periods=1).mean()
else:
    print("Temperature data ('temp') not found in merged_data.
        Skipping temperature-related features.")
```

Along with the Lagging features, we have also implemented hour of the day, day of the week and month and also if it was a weekend or not in that day. As we discovered that the price of the electricity varies significantly depending on the day of the week, these engineered features played a big role in increasing the accuracy of the model.

Listing 4: Adding time-based features

```
# Here we are adding time-based features
manual_features_df['hour'] = manual_features_df['Start_date'].dt.hour
manual_features_df['day_of_week'] = manual_features_df['Start_date'].
    dt.dayofweek
manual_features_df['month'] = manual_features_df['Start_date'].dt.
    month

# Add weekend feature (1 for weekend, 0 for weekday)
manual_features_df['is_weekend'] = manual_features_df['day_of_week'].
    apply(lambda x: 1 if x >= 5 else 0)
```

6.2.4 Training the model

The model is configured with hyperparameters tuned to balance complexity and performance, including 300 estimators, a learning rate of 0.05, and a maximum tree depth of 6. Categorical and datetime features are preprocessed for compatibility, with datetime columns converted to Unix timestamps and categorical columns encoded as numeric labels. The model is trained using the training dataset (X_train, y_train) while monitoring performance on the test set (X_test, y_test) through early stopping. A try-except block ensures robustness against potential errors during training.

Listing 5: Training XGBoost model with tuned hyperparameters

```
# Training  XGBoost model with tuned hyperparameters
xgb_model = XGBRegressor(
    objective='reg:squarederror',
    n_estimators=500,  # Reduced to prevent overfitting
    learning_rate=0.05,
    max_depth=6,  # Lowered to reduce complexity
    colsample_bytree=0.8,
    subsample=0.8,
    random_state=42
)

# Drop invalid columns or convert them to numeric
for col in ['Start_date', 'End_date_x', 'End_date_y', 'End_date', '
    dt_iso']:
    if col in X_train.columns:
        X_train[col] = pd.to_datetime(X_train[col], errors='coerce').
            astype(np.int64) // 10**9
        X_test[col] = pd.to_datetime(X_test[col],
        errors='coerce').astype(np.int64) // 10**9

for col in ['weather_main', 'weather_description', 'weather_icon']:
```

```
    if col in X_train.columns:
        X_train[col] = X_train[col].astype('category').cat.codes
        X_test[col] = X_test[col].astype('category').cat.codes

# Handle sklearn_tags error
try:
    xgb_model.fit(
        X_train, y_train,
        eval_set=[(X_test, y_test)],
        verbose=10
    )
except AttributeError as e:
    if '__sklearn_tags__' in str(e):
        print("Ignoring sklearn_tags error and proceeding with
            training.")
    else:
        raise e
```

6.3 Feature Importance

Feature	Importance
price_lag_24	0.581467362335562
price_lag_48	0.1198538766818046
Start_date	0.0481573538272171
Residual_load_MWh	0.04627955332984146
price_lag_72	0.04204835928990794
Lignite_MWh	0.02148945362357597
Hard_coal_MWh	0.022458044100407386
Wind_onshore_MWh	0.02058048970667057
Other_renewable_MWh	0.011054596345891537
Fossil_gas_MWh	0.009047382463955742

Tabelle 2: Top 10 Feature Importances

The above feature importance graph from our XGBoost model highlights the key factors influencing electricity price predictions. The most dominant predictors are lagged electricity prices, particularly the prices from the previous 24 and 48 hours (price_lag_24 and price_lag_48). This suggests that there are considerable time-dependent factors in power prices, with historical price trends having a big influence on current pricing. Apart from lagged prices, the residual load (Residual_load_MWh_) emerges as a crucial factor, suggesting that net electricity demand (after accounting for renewable energy contributions) plays a major role in determining price fluctuations. Among the different power generation sources, lignite, hard coal, and fossil gas-based electricity generation show relatively high importance. This likely reflects the impact of conventional power plant operations, fuel costs, and grid balancing mechanisms on electricity pricing. Interestingly, renewable energy sources such as onshore and offshore wind, solar photovoltaics, and hydropower, whose collective contribution is lower. Furthermore, weather conditions such as temperature, wind speed, humidity, pressure, and snowfall have very low significance. While weather trends influence electricity demand and renewable generation, their explicit contribution to price forecasting appears to be limited in the model now. These insights suggest potential areas for model improvement. Given the dominance of lagged price features, incorporating more advanced time-series transformations, such as additional rolling averages or seasonal decomposition, could enhance predictive accuracy. Additionally, further analysis of weather feature transformations (e.g., interactions between temperature and demand) might uncover hidden relationships that improve forecast performance. The relatively low impact of renewable energy variables could also indicate an investigation into grid balancing mechanisms and market regulations, which might be smoothing out their effect on prices.

6.4 Prediction

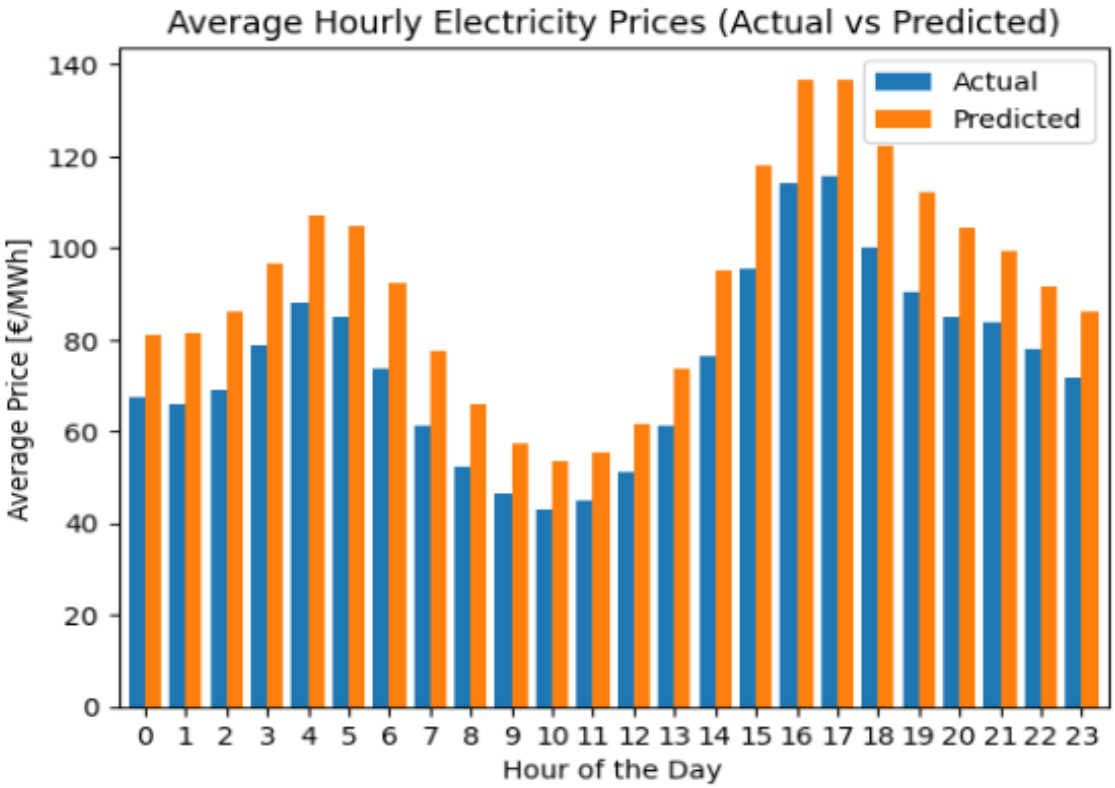
Prediction of the hourly price of electricity of the target date, which in our case is the 18th of February, 2025, We have to populate all the features used to train the model with the target date. We then have to load the data and run the prediction cell in the Jypyter notebook.

Listing 6: Load manually created features Excel file

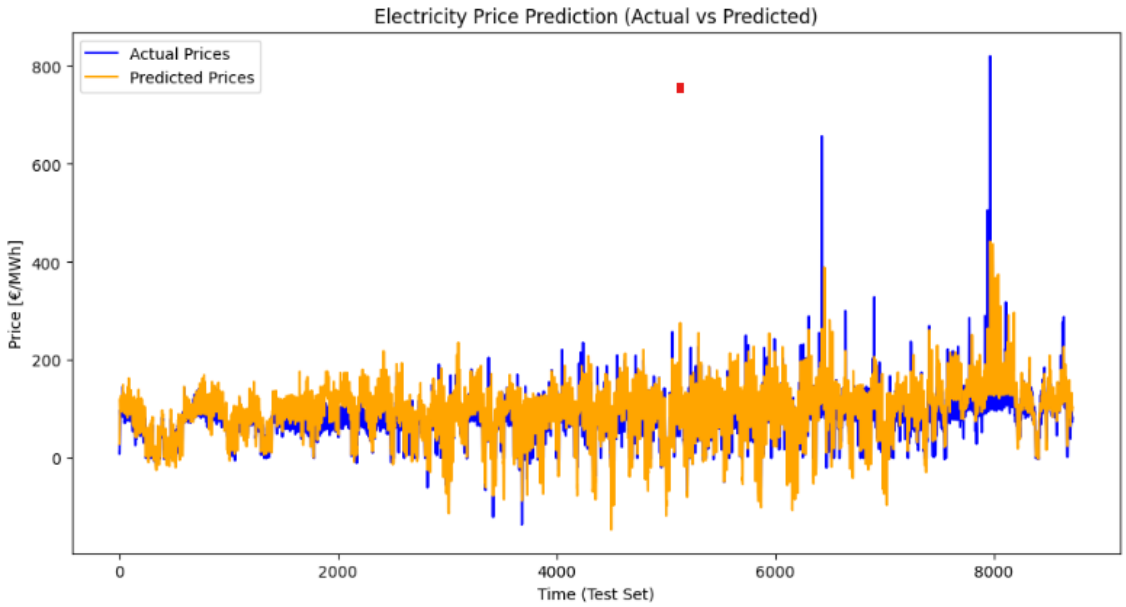
```
# Load manually created features excel file
manual_features_path = '/home/samad/Desktop/DS/Resources/
    Manual_Features.xlsx'
manual_features_df = pd.read_excel(manual_features_path)
```

6.5 Results

The trained model predicts day-ahead hourly electricity prices for a specified date (e.g., February 1, 2025). Predictions are stored in a CSV file and displayed in a tabular format. As for example, here we have attached the comparison of our predicted prices vs actual prices for February 1, 2025



Below are some of the important results visualized. In both the graphs we compared actual prices vs predicted prices and we can see that our model is doing good enough to match the actual price.



As we can see from the diagrams above, the model is close to the actual prices, and it is able to pick up the spikes properly. Although there are definitely more scope to make the accuracy greater by tuning the hyperparameters and adding more features.

7 Challenges Faced And Possible Improvements

During this project, we faced some challenges and how we handled them. Some of them are provided below:

7.1 Challenges faced

- Mean/Median Imputation: Replace missing values with the mean or median of the non-missing values in that column.
- Predictive Modeling: Our model is trained from 1st Jan 2015 till 7 Dec 2024 but in order to predict the price for a particular date (e.g. 01 Feb 2025), we have to manually populate the Excel file with the values of relevant features used during the training. We can get this data from the same sources we used during training. We are working to automate this stage by using an API.

7.2 Possible Improvements

While the current model performs well with an R^2 score of 0.666, there are lots of room for improvements, which we couldn't implement due to time and complexity constraints. Some of the possible ways in which we can potentially improve the performance are given below:

- **Interaction Features:** Create feature interactions between variables such as: Temperature × Demand (colder temperatures usually increase electricity demand), Wind Speed × Wind Generation (higher wind speeds correlate with more wind energy production) and Solar Radiation × Solar Generation

- **Hyperparameter Tuning:** Fine-tuning the hyperparameters of the selected model could enhance performance. We did this manually till we achieved the best possible results. But this could also be done using techniques such as Grid Search or Random Search.
- **Automating Input Feature For Price Prediction:** For the features, right now we are manually providing input to the model. This process can be automated using an API to fetch the required features of electricity and weather data to predict electricity prices.

7.3 Summary

In summary, the XG Boost Regression model performed well on the dataset from January 1st, 2015, to December 7th, 2024, with an R^2 score of 0.7 and an RMSE of 25.34. There is room for further improvement, particularly through better feature engineering, exploring more complex models, and leveraging time series-specific techniques.

8 Conclusion

We have developed a flexible machine -learning model capable of predicting hourly electricity prices for any given date, provided that the necessary feature data for that date is available. To achieve this, we trained the model using the latest data sourced from SMARD.de and OpenWeather.org.

This approach offers a significant advantage: it allows us to forecast electricity prices for any target date, rather than being restricted to specific predefined periods. However, it also comes with certain limitations. The primary drawback is that the feature data must be manually input, making the process more labor-intensive. Additionally, the model cannot generate predictions if the required feature data is unavailable.

Literatur

[1] Hamid Aghaie. Statistical analysis of the german electricity market in presence of renewables. In *2017 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, pages 1–5, 2017.

[2] Dominique Farag and Leeor Groen. Recent developments and future trends in germany's electricity market: an assessment of recent market developments on electricity prices and market stakeholders. Master's thesis, 2016.

[3] Dogan Keles, Dominik Möst, and Wolf Fichtner. The development of the german energy market until 2030—a critical survey of selected scenarios. *Energy Policy*, 39(2):812–825, 2011.