# Investigation on the performance of the EPF toolbox models with NGBoost Model for Electricity Price Forecasting

SUPPORTING MATERIAL

Maria Antony

# Introduction

This work deals with the analysis and evaluation of different models available for the prediction of electricity prices. There are many factors influencing the electricity price. Appropriate feature selection is critical in this regard. There exists a free-access library called as EPFtoolbox (Electricity Price Forecasting) which provides two models and five benchmark datasets for electricity price forecasting. As this work discusses the different models for price forecasting, we used the two models of the epf toolbox called LEAR and DNN and a probabilistic model called NGBoost(Natural Gradient Boosting). We have done all the experiments on three datasets provided by the EPFtoolbox library. And all the analyses and results are reported. The results are evaluated using different performance matrices *mean absolute error*(MAE), *symmetric mean absolute percentage error*(sMAPE), and *root mean squared error*(RMSE).

# List of Abbreviations

EPF: Electricity Price Forecasting

MAE: Mean Absolute Error

MSE: Mean Squared Error

RMSE: Root mean squared error

LEAR: Lasso Estimated Auto-Regressive

DNN: Deep Neural Network

SHAP: Shapley additive explanation

EPEX: European Power Exchange

GCA: grey correlation analysis

PCA: principal component analysis

DE: differential evolution algorithm

SVM: Support vector machine

# Repository Link

Please find the repository link for the code on Github.

https://github.com/maria-9625/Electricity-price-forecasting

Read me:

The repository contains the notebook for all the three models worked on three different datasets. Each model is worked for three datasets thereby there will be 9 models.

Installation

1. EPFToolbox

EPFTool box library can be easily installed via pip. First, clone the library and navigate to the folder[8]:

```
git clone https://github.com/jeslago/epftoolbox.git
cd epftoolbox
```

Then, simply install the library using pip:

```
pip install.
```

2. NGBOOST

`via pip`

```
pip install --upgrade ngboost
```

`via conda-forge`

```
conda install -c conda-forge ngboost
```

2.

# Context and Literature Investigation

We have done a critical review of different papers to understand the different methodologies and state-of-the-art electricity price forecasting. Papers on electricity price forecasting based on EPFtoolbox have been already reviewed and reported. Apart from that different papers on various models have been read and found out that NGBoost has a scope in the electricity price forecasting and reviewed certain papers on the same.

## Electricity Price Forecasting with Neural Networks on EPEX Order Books[1]

Machine learning techniques are used in this work to anticipate German energy spot market prices. The projections take into account not just spot market bid and ask order book data, but also fundamental market data such as renewable infeed and predicted total demand. Their findings demonstrate that neural networks can produce competitive order-book-based price estimates. They do not, however, perform considerably better than simpler approaches such as linear regression. Unlike the traditional order-book-based forecasting approach, network architecture optimization necessitates a large amount of statistical research. We also discovered that limiting the number of characteristics increases performance in general. The feed-forward neural network with only 10 features, as selected by the random forest, performs best in terms of RMSE.

## Research ArticleForecast of Short-Term Electricity Price Based on Data Analysis[2]

The framework in this article is used to anticipate power prices in New England. This research provides a large data-based forecasting framework that picks a minimal amount of data to accomplish accurate forecasting while decreasing time costs. The findings indicate that the risk of over-extracting data can be decreased. Furthermore, when compared to DS without data processing, forecasting accuracy is greater and time-cost is lower, with forecasting accuracy increasing from 81.68 percent to 91.44 percent and time-cost decreasing from 35,074 seconds to 1,809 seconds. The framework gives a strategy for projecting power prices that are widely applicable. The model, however, may be used in various disciplines of forecasting. This article introduces the model of power price forecasting, the data processing method, numerical findings, and case study details. The framework contains feature classification based on periodicity and temporal correlation, sample selection based on grey correlation analysis(GCA), feature selection based on GCA, feature extraction based on principal component analysis(PCA), and DE-SVM-based power price forecasting.

## Short-Term Solar Irradiance Forecasting Using Calibrated Probabilistic Models [3]

Since the project is an investigation of the performance of EPF toolbox models and the NGBoost model, it is very important to know about the NGBoost. A paper provides an idea about NGBoost. This work explains that for the forecasting of solar irradiance, the authors have developed different state-of-the-art probabilistic models. To ensure the predictions are well-calibrated, they have investigated post-hoc calibration techniques. The NGBoost model has been trained and evaluated on public data from seven stations in the SURFRAD network. This paper gives a clear picture of the different methodologies that have been used to improve the performance and thus got the result that on hourly-resolution forecasting, NGBoost with CRUDE post-hoc calibration yields equivalent performance to a numerical weather prediction model. The paper also found that At the intra-hourly resolution, NGBoost outperformed each of the baseline models without post-hoc calibration across all of the stations. In addition to this, At the hourly resolution, NGBoost outperformed two NWP model variations across three stations. NGBoost appears to be an excellent baseline for probabilistic solar irradiance forecasting at both intra-hourly and hourly resolutions, based on our findings.

## An interpretable probabilistic model for short-term solar power forecasting using natural gradient boosting[4]

A more recent article published in 2022 is all about the use of the NGBoost model for short-term solar power forecasting. The majority of Photovoltaic power forecasting methods are based on machine learning algorithms that give no insight or explanation into their estimates (black boxes). As a result, their direct use in contexts where openness is necessary, as well as the trustworthiness of their forecasts, may be called into doubt. In this particular regard, this paper has proposed a two-stage probabilistic forecasting framework so that it may be able to produce highly accurate, reliable, and sharp forecasts and so it would be a solution for the transparency of both the point forecasts and the prediction intervals.

As mentioned about the stages, the first stage carries out the implementation of NGboost, and the second stage deals with the calculation of shapely additive explanation(SHAP) values to give a clear idea about the intentions and the reasons behind the predictions. Experiments using the data from two PV parks in southern Germany have been conducted. The main findings in this paper are as follows: The suggested model did not produce any counter-intuitive or unexpected relationships, according to the findings. This discovery is critical since debugging machine learning models is a difficult undertaking. By choosing and then implementing the most significant available feature, an increase of roughly 6% in RMSE and 10% in CRPS may be obtained. The suggested model may be used by a variety of stakeholders, such as system operators and traders, whose decisions must be transparent and risk-free.

# Software Life Cycle

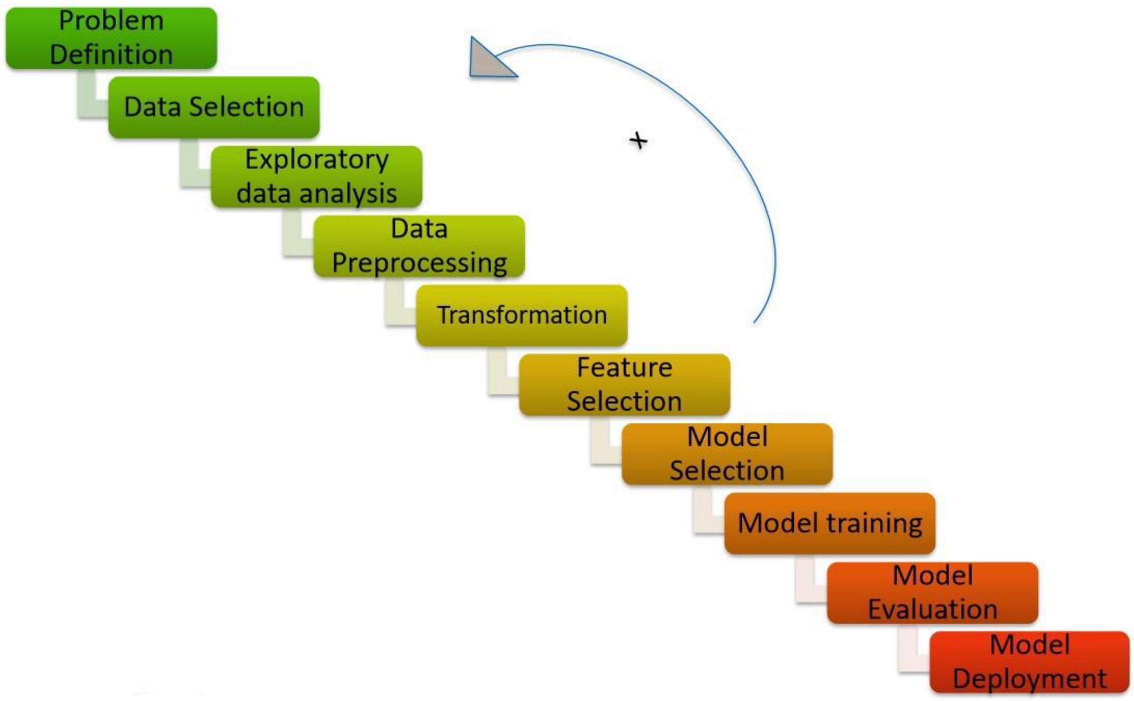


*Fig 1. Machine – Learning life cycle [5]*

1. Problem Definition

   The main objective of this project is to investigate the performance of different models for electricity price forecasting. The models, LEAR, and DNN are compared with a probabilistic model, NGBoost. The performances are evaluated using different evaluation matrices such as MAE, sMAPE, and RMSE.

2. Data selection

   As the project mainly focuses on an open-access library called as EPFToolbox, the library itself provides 5 different open benchmark datasets. Those data are from various sources.

| Market | Period |
|---|---|
| Nord pool | 01.01.2013 – 24.12.2018 |
| PJM | 01.01.2013 – 24.12.2018 |
| EPEX-France | 09.01.2011 – 31.12.2016 |
| EPEX-Belgium | 09.01.2011 – 31.12.2016 |
| EPEX-Germany | 09.01.2012 – 31.12.2017 |

*Fig 2. List of datasets available[6]*

- **Nord pool:** The Nord pool day-ahead electricity market, is one of the largest European power markets.
- **PJM:** The zonal prices of the COMED area in the *Pennsylvania-New Jersey-Maryland* (PJM) market.
- **EPEX-FR:** The French day-ahead electricity market.
- **EPEX-BE:** The Belgian day-ahead electricity market.
- **EPEX-DE**: The German day-ahead electricity market.

Each market contains 6 years of data as mentioned in figure 2[6].

Each dataset comprises historical prices and two relevant exogenous inputs based on day-ahead forecasts of price drivers. The day--ahead forecast representing other exogenous inputs are market dependent[6]:

- **Nord pool:** System load + Wind power generation.
- **PJM:** System load + Zonal load in the COMED area.
- **EPEX-FR:** System load + Generation in France
- **EPEX-BE:** System load in France + Generation in France
- **EPEX-DE**: Zonal load in the TSO Amprion zone + Wind power generation

| Date | Prices | Generation forecast | System load forecast |
|---|---|---|---|
| 2011-01-09 00:00:00 | 32.54 | 63065 | 63000 |
| 2011-01-09 01:00:00 | 21.55 | 62715 | 58800 |
| 2011-01-09 02:00:00 | 15.71 | 61952 | 58500 |
| 2011-01-09 03:00:00 | 10.58 | 59262 | 54300 |
| 2011-01-09 04:00:00 | 10.32 | 56883 | 51900 |
| 2011-01-09 05:00:00 | 10.33 | 56332 | 50900 |
| 2011-01-09 06:00:00 | 9.22 | 55096 | 50100 |
| 2011-01-09 07:00:00 | 10 | 55507 | 51000 |
| 2011-01-09 08:00:00 | 10.19 | 58763 | 53200 |
| 2011-01-09 09:00:00 | 30 | 60500 | 54800 |

*Fig 3. Sample dataset preview[6]*

Each dataset contains 24hours of data for each day.

## 3. Exploratory data analysis(EDA)

It is an important stage where we can analyze the data in the initial stage and provides some inferences from the data. Using summary statistics and graphical representations, the data is investigated and essential insights and qualities of the data are provided [5].

```
df_train
```

| Date | Price | Exogenous 1 | Exogenous 2 |
|---|---|---|---|
| **2011-01-09 00:00:00** | 32.542 | 63065.0 | 63000.0 |
| **2011-01-09 01:00:00** | 21.549 | 62715.0 | 58800.0 |
| **2011-01-09 02:00:00** | 15.711 | 61952.0 | 58500.0 |
| **2011-01-09 03:00:00** | 10.583 | 59262.0 | 54300.0 |
| **2011-01-09 04:00:00** | 10.324 | 56883.0 | 51900.0 |
| **...** | ... | ... | ... |
| **2016-01-02 19:00:00** | 37.360 | 65413.0 | 64725.0 |
| **2016-01-02 20:00:00** | 32.840 | 63183.0 | 61618.0 |
| **2016-01-02 21:00:00** | 26.150 | 60221.0 | 58354.0 |
| **2016-01-02 22:00:00** | 30.620 | 60328.0 | 55770.0 |
| **2016-01-02 23:00:00** | 36.190 | 61835.0 | 59887.0 |

43680 rows × 3 columns

*Fig 4. A sample from the dataset[7]*

The dataset contains 43680 rows and 3 columns. Thus we used 3 datasets PJM, NP, and FR for the electricity price forecasting. And we have saved each training and test data from the library and saved it as a CSV file and used it for other models.

## 4. Data Pre-processing

Once the datasets are imported, then we have to process the data for training and testing. Data pre-processing is crucial as the erroneous data may lead to wrong outputs. Cleaning up the data takes the longest, but it's necessary for deleting inaccurate data and filling in gaps. Important tasks include:

- Removing unnecessary data and outliers
- Handling missing data
- Conforming data to a standardized pattern.
- Masking private or sensitive data entries.

```
df_train.isnull().sum()
```

```
Price          0
Exogenous 1    0
Exogenous 2    0
dtype: int64
```

*Fig 5. Checking for null values[7]*

Since the data is readily available and is provided by the library, all the datasets are ready to use. Figure 5 shows that there are no null values or outliers. All the datasets are ready for further steps.

## 5. Transformation

During this process, we'll see if we can extract some relevant insights from the current columns and then change them in ways that will be beneficial for modelling [5].

| | Real price | DNN 1 | LEAR 56 |
|---|---|---|---|
| **27/12/2016 00:00** | 24.08 | 24.530144 | 25.1290 |
| **27/12/2016 01:00** | 22.52 | 23.115845 | 24.2083 |
| **27/12/2016 02:00** | 20.13 | 22.563509 | 23.9289 |
| **27/12/2016 03:00** | 19.86 | 22.047380 | 23.4943 |
| **27/12/2016 04:00** | 20.09 | 23.055725 | 23.5104 |
| **...** | ... | ... | ... |
| **26/01/2017 19:00** | 30.66 | 30.362480 | 30.3091 |
| **26/01/2017 20:00** | 29.85 | 29.554022 | 30.2290 |
| **26/01/2017 21:00** | 29.67 | 29.148659 | 29.6552 |
| **26/01/2017 22:00** | 29.05 | 28.654337 | 29.2686 |
| **26/01/2017 23:00** | 28.66 | 28.070572 | 28.9244 |

744 rows × 3 columns

**NP_LEAR_MAE_MAPE**

```
MAE_NP = pd.DataFrame(columns=['MAE','sMAPE(%)'], index=NP.index)
for date in NP.index:
    MAE_PJM.loc[date]['RMSE'] = RMSE(pd.Series(NP.loc[date]['LEAR 56']), pd.Series(NP.loc[date]['Real price']))
    MAE_NP.loc[date]['MAE'] = MAE(pd.Series(NP.loc[date]['LEAR 56']), pd.Series(NP.loc[date]['Real price']))
    MAE_NP.loc[date]['sMAPE(%)'] = sMAPE(pd.Series(NP.loc[date]['LEAR 56']), pd.Series(NP.loc[date]['Real price'])) * 100
```

*Fig 6. Transforming the data for evaluation purposes[7]*

Here the main dataset has been changed in a particular format to evaluate the models. Figure 6 gives us an idea about the same. Here the dataset of Nordpool has been changed in such a way that, the results of the DNN model and LEAR models provided by the epf toolbox are concatenated along with the price to find MAE, sMAPE, and RMSE. The transformations have been done for all the other datasets as well to evaluate and compare the models used for electricity price forecasting.

## 6. Feature selection

Feature selection deals with the fetching of the important and relevant features for forecasting. Since the data is readily available in this project, train and test data have been fetched directly from the library and saved to a CSV file.
In turn, these CSV files have been used for all the other models which are included in the forecasting.

## 7. Model selection

We have used 3 different models. Since this project is all based on a comparison of different datasets for electricity price forecasting, we have used 2 models provided by the epf library, and another model used is a probabilistic model called as NGBoost. Since the data acceptable for LEAR and DNN models is 1 year of data(365 days), both the models have been tested for 365 days. NGBoost model also has been trained for 365 days. The documentation of the library says that, for further research, it is more likely to use 30 days of data. In this regard, NGBoost has trained for all the 3 different datasets for 30 days and analyzed them.

## 8. Model Training

As mentioned above, the whole 3 datasets have been trained and it took around 8 to 10 hours for training and test each dataset. Along with the forecasted values, it is trained in such a way that it returns mean and standard deviation as well.

## 9. Model Evaluation

All three models have been used to forecast the price of electricity. The main objective of this work is to compare the performance of the three different models that have been used. All the models are evaluated by comparing all the essential performance matrices such as MAE, sMAPE, and RMSE and recorded.

## 10. Model deployment

We have developed and created a model in a training environment. All the notebooks associated with the project are deployed as a GitHub repository[7].

# RESULTS

Apart from all the results mentioned in the report, we do have some more results which is an extension of existing results.

## 1. PJM Dataset

The major findings are as follows:

1. Along with the MAE, sMAPE, and RMSE, we have found the sharpness score and plotted the average sharpness score and average RMSE for each day of the week as follows for NGBoost

| | Sharpness Score | RMSE |
|---|---|---|
| **Day** | | |
| **Friday** | 7.746192 | 0.284519 |
| **Monday** | 7.446509 | 0.182714 |
| **Saturday** | 5.063388 | 0.225623 |
| **Sunday** | 7.366110 | 0.319092 |
| **Thursday** | 11.446203 | 0.130283 |
| **Tuesday** | 9.045150 | 0.199045 |
| **Wednesday** | 7.386486 | 0.196958 |

*Fig 7. Average sharpness and average RMSE of the PJM dataset[7].*

2. The above data in figure 7 has been plotted as a line graph.

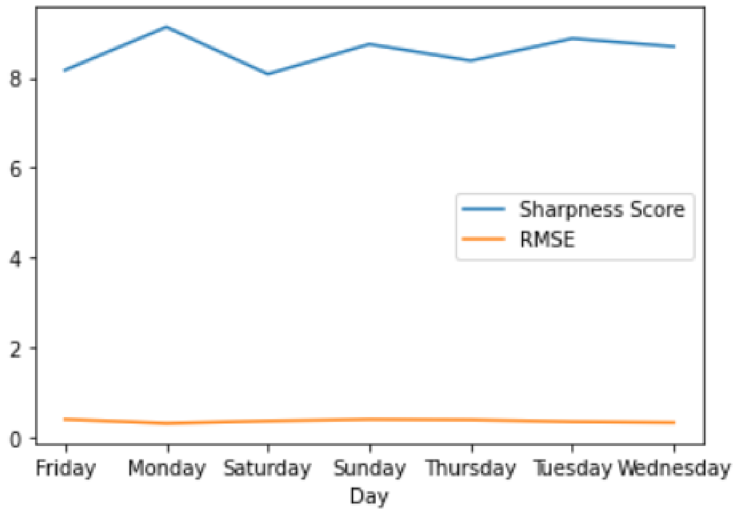


*Fig 8. Average sharpness and average RMSE of PJM dataset in a line graph[7]*

3. The above data in figure 7 has been plotted as a bar graph.

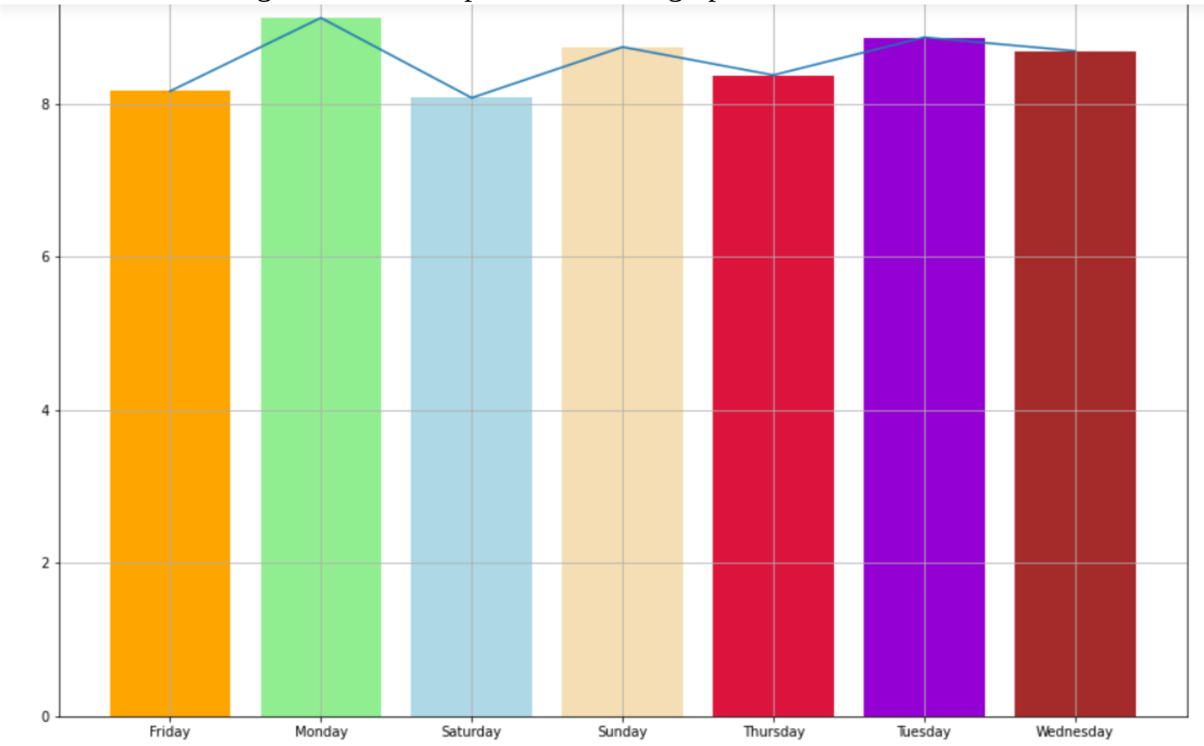


*Fig 9. Average sharpness on NP dataset for NGBOOST in a bar graph[7]*

All the above findings result in an average sharpness score of 8.5792 and an average RMSE of 0.3711 for NGBOOST

Average RMSE OF LEAR: 3.0575

Average RMSE of DNN: 2.9298

## 2. NP dataset

1. Similar to PJM, we have found MAE, sMAPE, and RMSE for the Nordpool dataset, we have found the sharpness score and plotted the average sharpness score and average RMSE for each day of the week as follows for NGBoost.

| | Sharpness Score | RMSE |
|---|---|---|
| **Day** | | |
| **Friday** | 3.428528 | 0.275153 |
| **Monday** | 3.632868 | 0.194087 |
| **Saturday** | 3.245672 | 0.203942 |
| **Sunday** | 2.826510 | 0.167532 |
| **Thursday** | 3.862138 | 0.178480 |
| **Tuesday** | 3.699055 | 0.271424 |
| **Wednesday** | 4.158732 | 0.212043 |

*Fig 10. Average sharpness and average RMSE of NP dataset[7].*

2. The above data is shown in a line graph as follows:

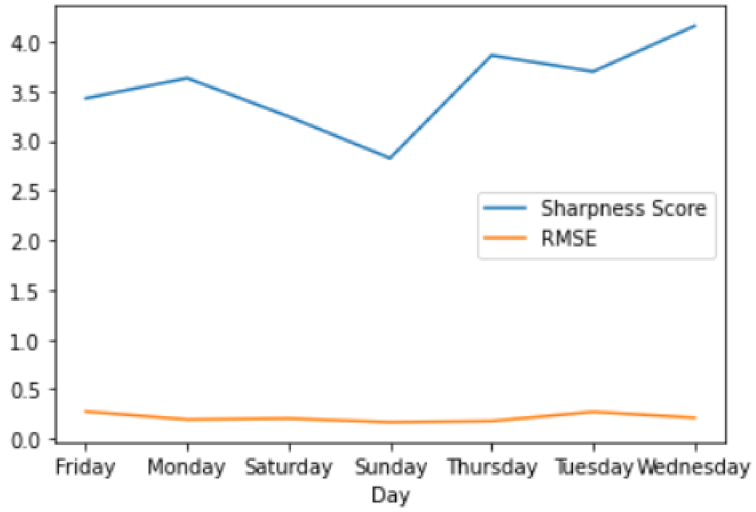


*Fig 11. Average sharpness and average RMSE of NP dataset in a line graph[7]*

3. The sharpness score of the NP dataset has been plotted as a bar graph.

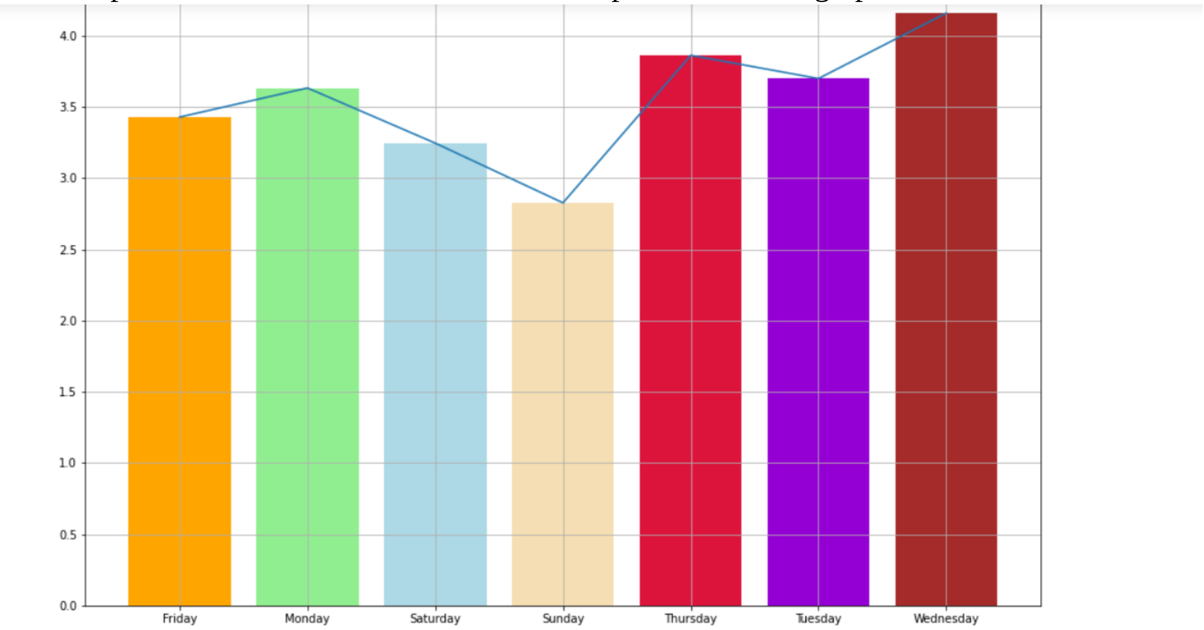


*Fig 12. Average sharpness on NP dataset for NGBOOST in a bar graph[7]*

Average sharpness score: 3.5504

Average RMSE of NGBOOST: 0.2085

Average RMSE of LEAR: 3.1847

Average RMSE of DNN: 2.9032

## 3. FR dataset

1. Similar to PJM and NP, we have found MAE, sMAPE, and RMSE for the French dataset, we have found the sharpness score and plotted the average sharpness score and average RMSE for each day of the week as follows for NGBoost

| Day | Sharpness Score | RMSE |
|---|---|---|
| Friday | 8.166684 | 0.409145 |
| Monday | 9.124862 | 0.322097 |
| Saturday | 8.079418 | 0.372410 |
| Sunday | 8.744356 | 0.407894 |
| Thursday | 8.376660 | 0.398017 |
| Tuesday | 8.870085 | 0.356222 |
| Wednesday | 8.692618 | 0.338847 |

*Fig 13. Average sharpness and average RMSE of FR dataset[7].*

2. The above data is depicted in a line graph as follows:

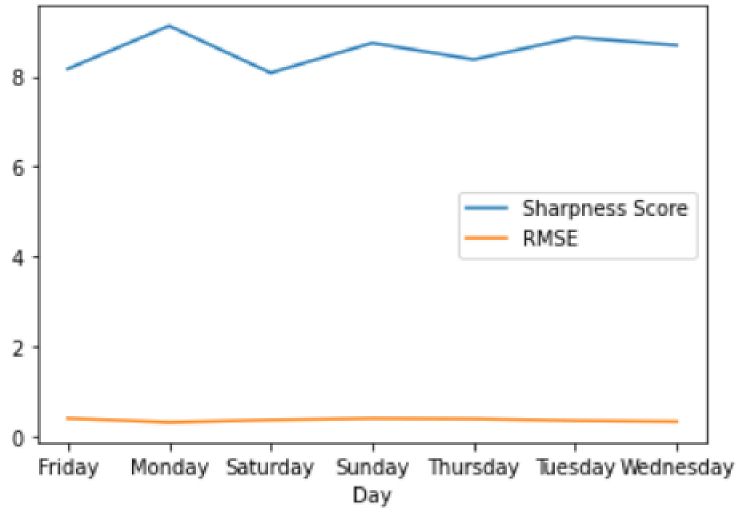


*Fig 14. Average sharpness and average RMSE of FR dataset in a line graph[7]*

3. The sharpness score of the FR dataset has been plotted as a bar graph.

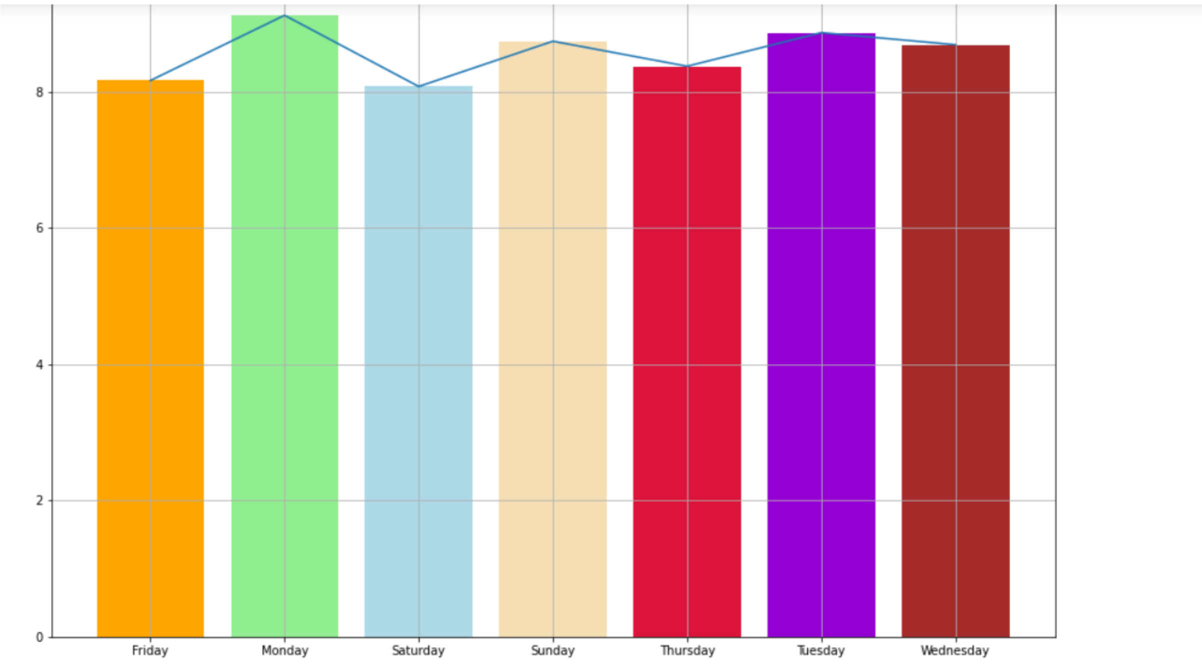


*Fig 15. Average sharpness on FR dataset for NGBOOST in a bar graph[7]*

Average sharpness score: 8.5792

Average RMSE of NGBOOST: 0.3711

Average RMSE of LEAR: 6.1625

Average RMSE of DNN: 5.4525

# Conclusions and further Improvements :

After training, testing, and evaluating, the analysis and evaluation say that out of all the three models DNN is the best one for electricity price forecasting. Unfortunately, NGBOOST is the worst when compared to other models. This is clear from the table mentioned in the report. Since the NGBoost depends on different hyperparameters, the correction proportion of the hyper parameter values might be a reason for its worst behaviour. This can be taken for further research. Another scope of research is nothing but finding the sharpness score for the models provided by the EPFtoolbox. So that, the average sharpness score for each particular day in a week can be plotted for better understanding.

# Tools Used:

The codes have been written in the jupyter notebook.

Editing and writing of the report and supporting material have been done in overleaf and Microsoft word.

The code has been published in Github repository

## Verification and Validation

Verification of each and everything has happened twice every week. Each time the work has been observed by the supervisor and got the feedback which helped us to ensure that we are on the track. The unknown topics are all cleared by the supervisor to ensure that the progress is hassle-free. We have gone through different papers on electricity price forecasting to know the state-of-the-art algorithm and got to know about the EPFtoolbox library and related models. Since there were no papers found on NGBoost for electricity price forecasting, it has a good scope for research. Based on that, we have gone through the papers related to NGBoost. Since it is a new attempt, proper verification via regular meetings was there to keep track of the progress.

Validation was completed to ensure that all the functional and ethical requirements are met. And also validation has happened to intend to check the accuracy and performance using different matrices.

## Challenges faced

1. The LEAR model accepts only a year of data as an input, so it took around 8 to 10 hours for each dataset.
2. NGBOOST doesn't accept the date as it is. So we have changed into the desired format using the lambda function.
3. Since the distribution of mean and standard deviation in EPFtoolbox is not defined, we were unable to find the sharpness score.

# References


1. Schnürch, S., & Wagner, A. (2020). Electricity Price Forecasting with Neural Networks on EPEX Order Books. *Applied Mathematical Finance*, *27*(3), 189–206. https://doi.org/10.1080/1350486X.2020.1805337
2. Wu, S., He, L., Zhang, Z., & Du, Y. (2021). *Forecast of Short-Term Electricity Price Based on Data Analysis*. https://doi.org/10.1155/2021/6637183
3. Zelikman, E., Zhou, S., Irvin, J., Raterink, C., Sheng, H., Avati, A., Kelly, J., Rajagopal, R., Ng, A. Y., & Gagne, D. (2020). *Short-Term Solar Irradiance Forecasting Using Calibrated Probabilistic Models*. https://arxiv.org/abs/2010.04715
4. Mitrentsis, G., & Lens, H. (2022). An interpretable probabilistic model for short-term solar power forecasting using natural gradient boosting. *Applied Energy*, *309*, 118473. https://doi.org/10.1016/J.APENERGY.2021.118473
5. https://medium.com/analytics-vidhya/machine-learning-development-life-cycle-dfe88c44222e
6. https://epftoolbox.readthedocs.io/en/latest/modules/data_extract.html
7. https://github.com/maria-9625/Electricity-price-forecasting
8. https://epftoolbox.readthedocs.io/en/latest/modules/started.html