Predicting aFRR Up-Regulation Energy Prices Using Weather, Demand-Side, and Spot Price Data

Jiahong Chen

C111: Introduction to Machine Learning for Physical Sciences

Dr. Alexander Lozinski

Introduction

The Automatic Frequency Restoration Reserve (aFRR) market is one of the key balancing mechanisms in Europe. The system operators automatically conduct up-regulation or down-regulation to maintain the grid frequency every few seconds. Up-regulation occurs when demand is higher than expected or supply is lower than expected, while down-regulation occurs in the reverse circumstances. Among the key balancing indicators, the aFRR Up-regulation price, referring to the marginal price for upward balancing energy, captures the real-time scarcity of supply. The Up-regulation price responds to unpredictable weather conditions as well as demand-side factors, making it often volatile.

The intention of this project is to model the Finnish aFRR Up-regulation price based on weather factors, such as cloud amount, wind speed, and precipitation, as well as demand-side factors such as electricity consumption and public holiday status. In addition to the weather and demand-side data, the day-ahead spot price is also included for training, acting as a market reference. Understanding how these variables influence real-time balancing prices provides valuable insight into the effectiveness of the balancing mechanism and supports forecasting for stakeholders.

Dataset and Feature selection

The project uses dataset published at https://doi.org/10.5281/zenodo.17494556. This dataset contains hourly data from the Automatic Frequency Restoration Reserve (aFRR) energy market in Finland, covering the period from June 20, 2024, to March 16, 2025 (UTC). This dataset contains 6456 observations with 17 variables as follows:

1. *Datetime*: Timestamp in UTC+2 (Finnish local time).
2. *Up*: aFRR Up-regulation energy price (EUR/MWh).
3. *Down*: aFRR Down-regulation energy price (EUR/MWh).
4. *Sp*: Day-ahead electricity market (spot) price (EUR/MWh).
5. *cloud_amount*: Fractional cloud cover (%) at representative Finnish weather stations.
6. *wind_speed*: Average hourly wind speed (m/s).
7. *precipitation_amount*: Hourly precipitation (mm).
8. *pressure*: Atmospheric pressure (hPa) measured at surface level.
9. *air_temperature*: Air temperature (°C) at surface level, averaged over the hour.
10. *relative_humidity*: Relative humidity (%) of the air.
11. *wind_direction*: Average wind direction (degrees, 0–360°), measured clockwise from north.
12. *Down_Cap*: aFRR Down-regulation capacity market price (EUR/MW).
13. *Up_Cap*: aFRR Up-regulation capacity market price (EUR/MW).
14. *is_public_holiday*: Binary indicator (0 = no, 1 = yes) identifying whether the timestamp falls on a Finnish public holiday.
15. *electricity_consumption*: Total Finnish electricity consumption (MWh) at the system level.
16. *electricity_consumption_Finnish_networks*: Electricity consumption measured specifically within Finnish transmission networks (MWh).
17. *electricity_consumption_forecast*: Day-ahead forecast of total Finnish electricity consumption (MWh).

Since the aim of the project is to examine the effect of weather and demand-side factors, other market indicators such as capacity market price are neglected. In addition, the electricity consumption forecast indicates day-ahead predictions instead of real-time weather conditions and is therefore also neglected. Another variable to be considered is the day-ahead spot price. This factor is included in the prediction, in addition to the weather and demand-side factors, in order to provide a benchmark for prices at the same time of day. The importance of the spot price is further justified in the discussion section. To conclude, the target variable is *Up*, and the predictive features include: *sp, cloud_amount, wind_speed, precipitation_amount, pressure, air_temperature, relative_humidity, wind_direction, is_public_holiday, electricity_consumption, electricity_consumption_Finnish_networks.*

Data Preprocessing

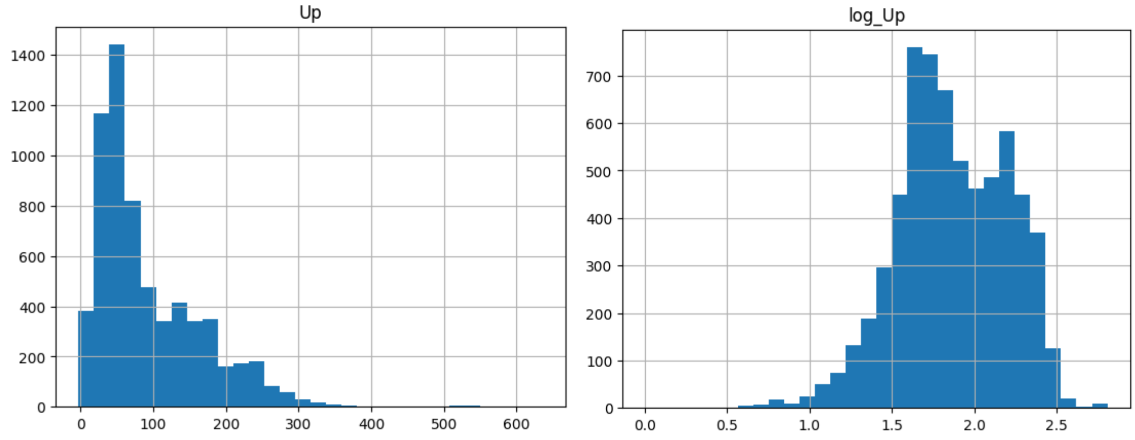
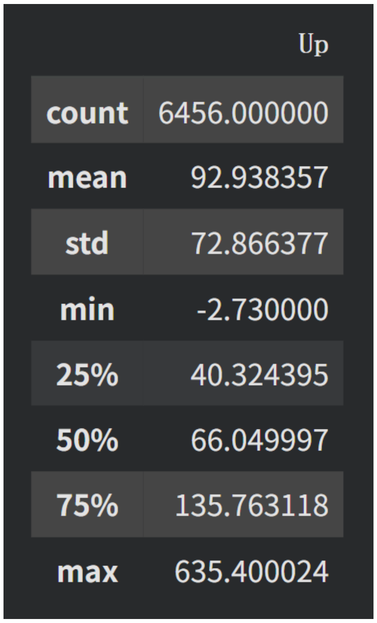


Figure 1: Distribution of aFRR Up-regulation energy price before and after log transformation



	Up
count	6456.000000
mean	92.938357
std	72.866377
min	-2.730000
25%	40.324395
50%	66.049997
75%	135.763118
max	635.400024

Figure 2: Summary statistics of aFRR Up-regulation energy price

Since the dataset has no missing value, it is directly loaded and stored for further processing by pandas library.

For y variable, aFRR Up-regulation energy price, a histogram is plotted to visualize the distribution. From Figure 1, the distribution is highly skewed right with a wide spread of values. Therefore, a log function with base 10 is used to scale the variable. Since the minimum value of Up is around 2.73, a constant of 3.74 is added to the value before taking logarithm to keep the value after transformation positive that is easy for further modelling. The distribution of aFRR Up-regulation energy price after the transformation is shown in the right of Figure 1, which shows the distribution is less skewed after the transformation.

For x variables, it contains both categorical and numerical data. The only categorial data is, *is_public_holiday*. Since the data is binary and without implicit ordering, it has been directly inputted as 0 or 1 for training. For the scaling, StandardScaler() method is used in the pipeline to scale to x variables for training.

Modelling

In this project, the target variable, aFRR Up-regulation energy price, is a numerical variable, so modelling the price can be defined as a regression task. To find a well-fit model, four regression models were evaluated: Ridge Regression, Support Vector Regression (SVR), Random Forest Regression, and a Multi-Layer Perceptron (MLP) neural network. All models are trained through Pipeline in scikit-learn library, combining data scaling and model fitting to ensure fair comparison. The dataset is split into training and test sets in a 70-30 ratio. Following is the explanation of these 4 implemented regression models and the chosen parameters:

1. Ridge Regression

Ridge Regression is a linear model that adds L2 regularization to stabilize coefficient estimates and prevent overfitting. For the alpha that multiplies the L2 term, it has been set in default value of 1.0 to keep the medium regularization strength to prevent overfitting or underfitting.

2. Support Vector Regression

Support Vector Regression uses kernel functions to model nonlinear relationships by mapping inputs into high-dimensional feature spaces. For SVR model, all parameters are kept default since the default values are tested to give a good prediction. Specifically, C is set to be 1.0 and epsilon is set to be 0.1 as default.

3. Random Forest

Random forest is an ensemble model that builds many decision trees and averages their predictions, allowing it to capture interactions and nonlinear patterns in the data. For random forest model, the random state is set to 42, following the standard practice. Other parameters are kept default. For example, max_depth is not specified to make to the decision tree to expand widely to capture the relatively complex relationship between variables. It is tested that limiting max_depth would reduce model performance.

4. Multi-Layer Perceptron (MLP) Regressor

The MLP Regressor is a neural-network model capable of learning highly nonlinear relationships through multiple hidden layers. In MLP model, hidden_layer_sizes is set to be (100,50) to capture complex relationships. It has been tested that using a default of one layer (100,) or two layers of smaller size would result in poorer performance. More complex layer sizes are not used to save computational cost and time.

Result

Table 1: Root mean square error (RMSE) for 4 models by cross validation method

	Ridge Regression	Support Vector Regression	Random Forest	Multi-Layer Perceptron (MLP) Regressor
RMSE (6 digits)	0.399531	0.215176	0.197393	0.209452

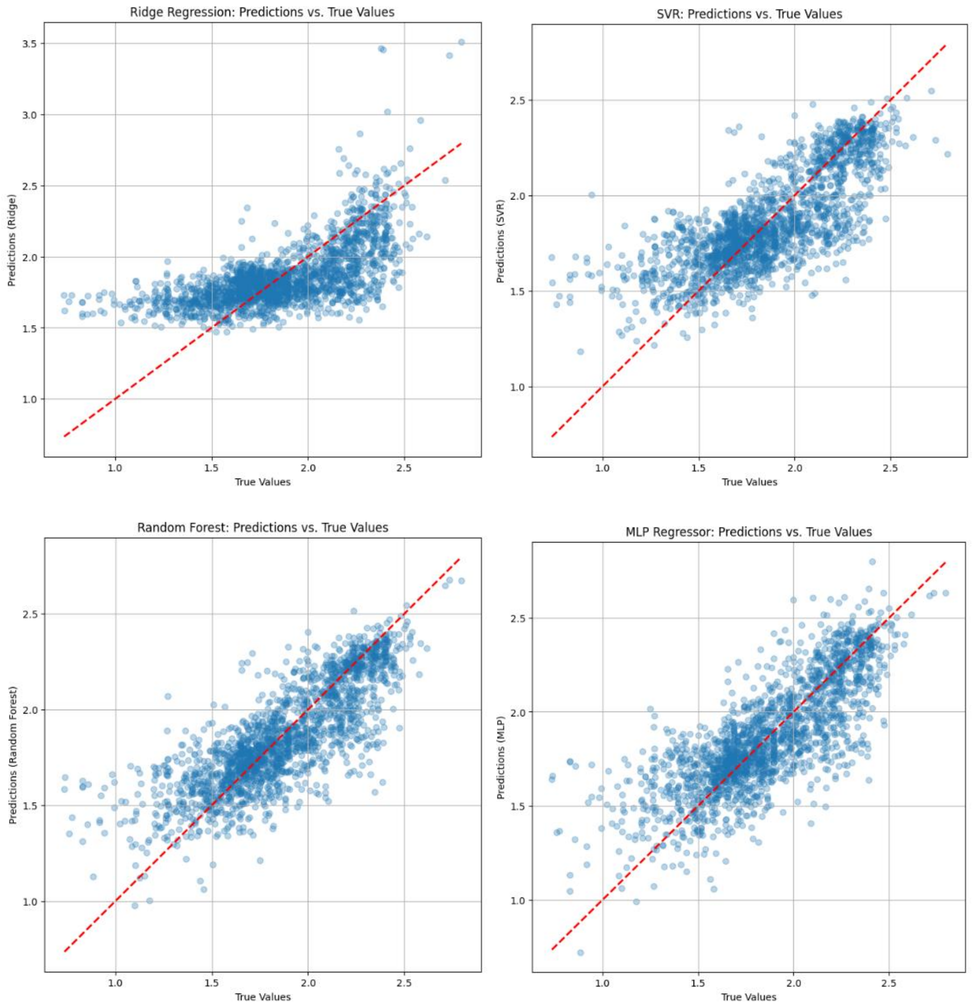


Figure 3: Prediction value against true value (with logarithm transformation) scatterplots for 4 models

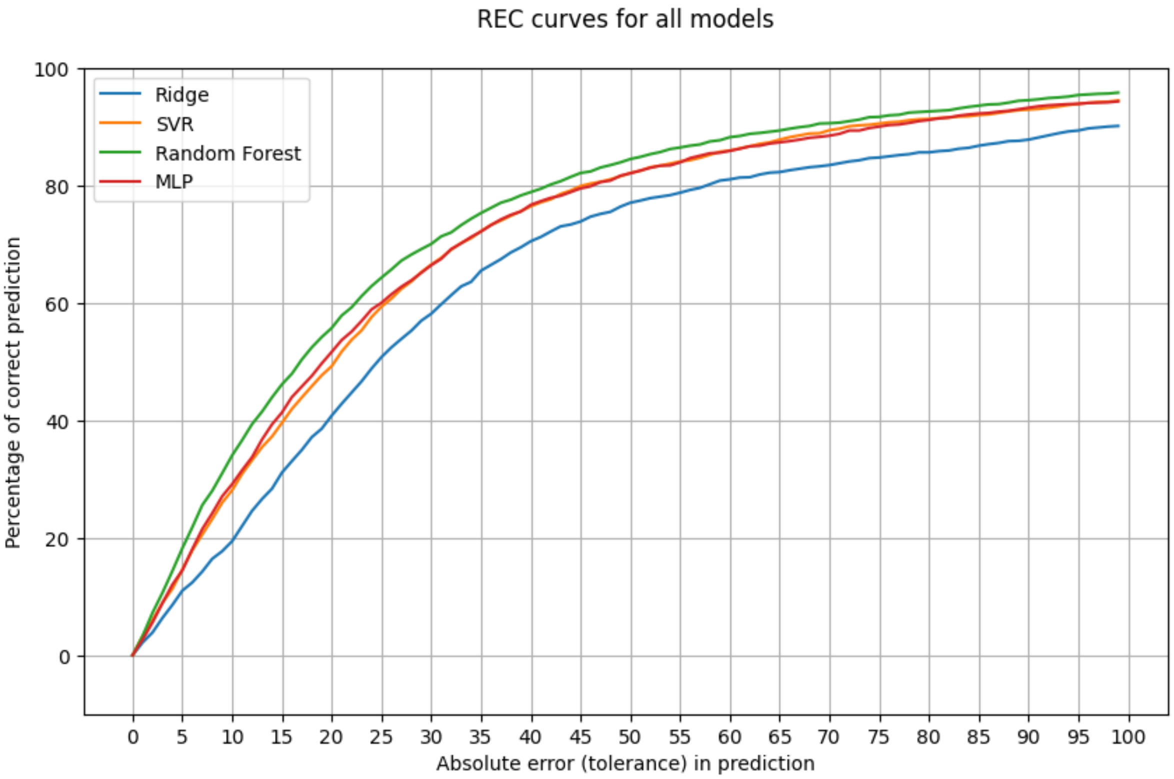


Figure 4: Regression error characteristic (REC) curves comparison for 4 models

Discussion

To assess model performance, Root Mean Square Error (RMSE) is calculated for comparison. In order to reduce the variance caused by random selection, K folds cross validation method is used. As mentioned in the preprocessing part, dataset is first split into 70% training data and 30% test data. During the K fold process, the 70% training data is further divided into 5 folds for cross-validation. In each cross-validation round, four of these parts (together representing 56% of the full dataset) are used to train the model, and the remaining one part (representing 14% of the full dataset) is used to evaluate it. After all 5 iterations, the model with the lowest validation RMSE is selected and then evaluated once on the 30% test set. The resulting RMSE is recorded in Table 1, and the process is applied to 4 methods of modelling. From the Table 1, random forest has lowest RMSE of 0.197393, followed by Multi-Layer Perceptron (MLP) Regressor and Support Vector Regression, while Ridge Regression has the highest RMSE of 0.399531, which is significantly larger than the other 3 models. As smaller RMSE indicates better performance, it has shown that random forest model makes the best prediction, and Ridge regression performs notably worse than the other models.

The performance of each model is further examined through the scatterplot of predicted values against true values shown in Figure 3. For ridge regression, a tail of data appears in left part of scatterplot that is above the 1:1 base line: the model overestimates the value when the true value is small. For the other 3 models, the data point generally scattered around the 1:1 base line with slight variation, which confirms their better performance over the ridge regression. A closer examination of graphs specifies the difference among these 3 better-performed models. For SVR model, the datapoints are scattered slightly above the base line in smaller values (similar pattern as the ridge regression) compared to the random forest model. For MLP model, the datapoints have higher spread around base line than the random forest when the true value is large (right ends on graphs). This observation also aligns with the RMSE result that random forest model performs better than the SVR and MLP model.

The Regression Error Characteristic (REC) curve visualizes how well each model predicts the target variable across different levels of tolerated error. From Figure 4, the random forest curve is on top of the other 3 curves, achieving the largest percentage of correct predictions on every tolerance level. The SVR and MLP curve are located slightly below the random forest curve with some intersections between these 2 curves. This shows that these 2 models have relatively similar performance levels across different error thresholds. The ridge regression curve is located below the 3 curves on every tolerance level, which is consistent as the RMSE and scatterplot analysis.

Since the linear model (ridge regression) performs worse than the other 3 models that are better at capturing non-linear patterns, it can be infer that this regression task is relatively complex. There are some potential reasons explaining the potential differences. First, the relationship between the predictors and the target variable is non-linear. For example, extreme weather leads to threshold jumps in price instead of the linearly affecting the price. Price is considered to be highly volatile, and more non-linearity could be expected, but polynomial features are not included in ridge regression unless manually added. Second, there would be interactions of features that linear model fails to consider. For example, it is common that weather features are affecting each other such as the interaction of wind speed and the wind direction, but the feature term such as wind speed × wind direction is not included in ridge regression unless manually added. Third, the y-variable after logarithm scaling is still slightly skewed. From Figure 1, the *log_Up* is skewed left although being less skewed than the *Up*. The linear model performs worse in skewed distribution since it is trying to average out the extreme value, while the model such as random forest applying decision tree could perform better in dealing with the situation.

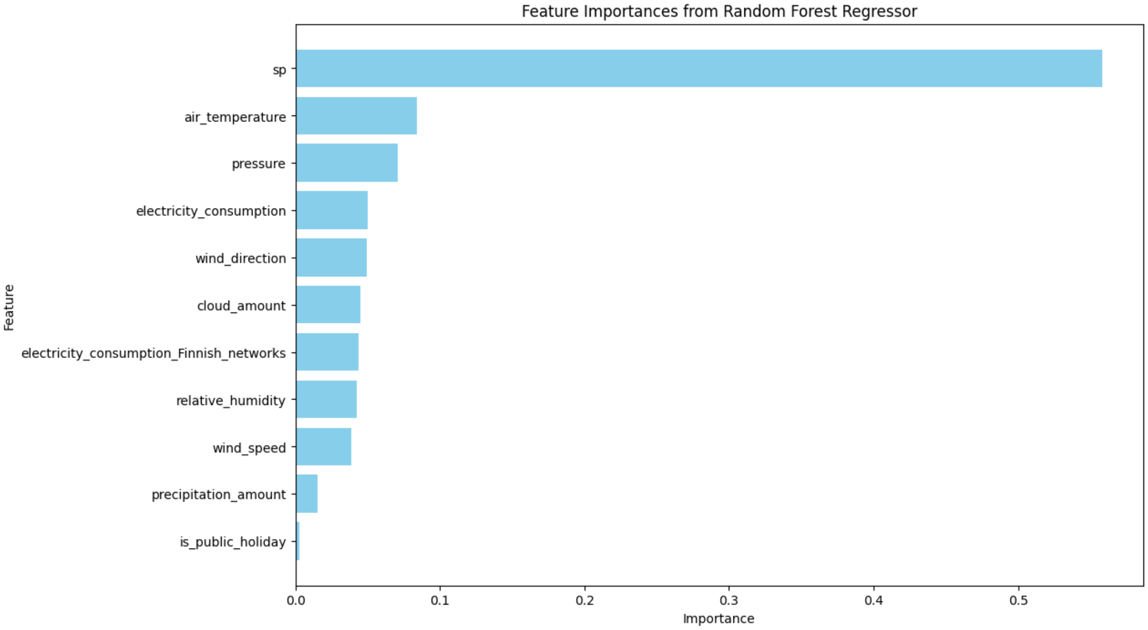


Figure 5: Feature importances from the random forest model

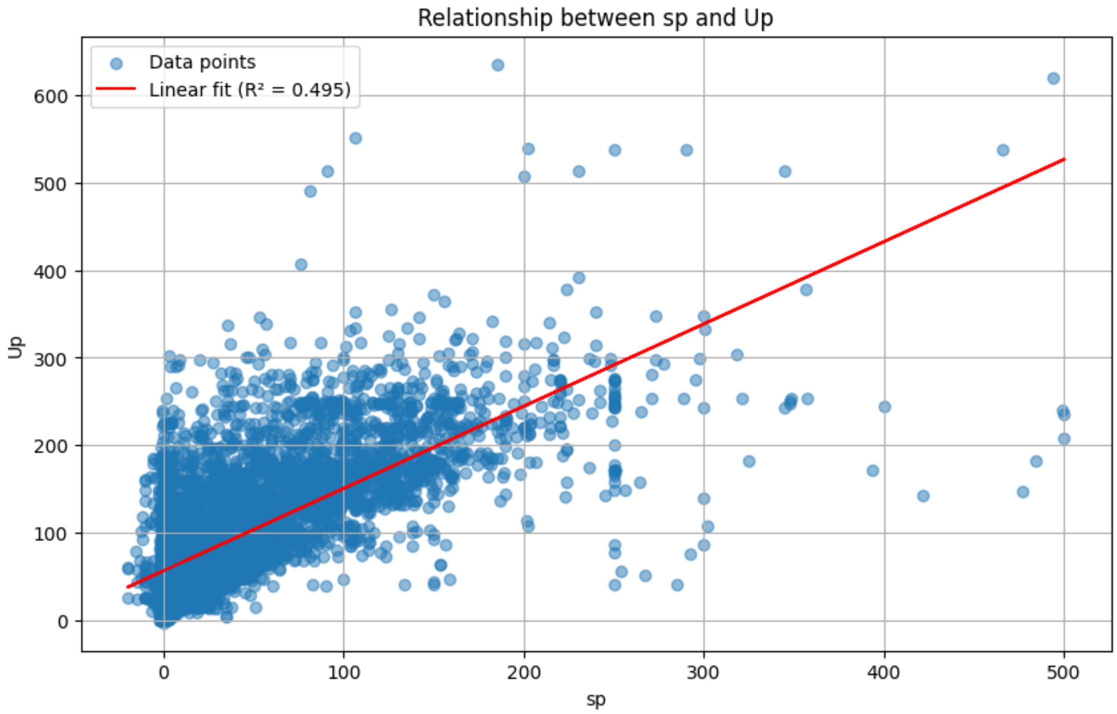


Figure 6: Correlation between spot price and aFRR Up-regulation price

The random forest model is shown to have the best performance among the 4 models. An examination on random forest feature importance brings some insights on how each parameter contributes to the prediction. Feature importance in a Random Forest model measures how much each input variable contributes to improving the model's predictive accuracy. From Figure 5, the spot price, *sp*, has the highest feature importance 0.558440 followed by other features such as *air temperature* and *pressure*. This justifies the inclusion of the spot price for prediction since it provides a market reference for *Up*. Another scatterplot of *Up* against *sp* (Figure 6) is generated to demonstrate the potential relationship between them. These 2 variables generally follow a weak to medium positive correlation with R^2 of 0.495, but the error increases as the spot price increases. This shows that spot price provides a fairly good indication of the up-regulation energy price when the price is low, demonstrating a more typical situation. However, as the price goes up, it represents more extreme cases typically involving extreme weather or spikes in consumption, where spot prices are less accurate. This makes sense as the day-ahead spot price is the electricity price determined one day before the actual delivery of power, which reflects expected balance of supply and demand under normal conditions. In more extreme cases, real-time energy consumption and weather conditions could be more important in affecting the up-regulation price.

Conclusion

In conclusion, the project predicts aFRR Up-regulation energy price based on weather data, demand-side data, and spot price through four regression models: Ridge Regression, Support Vector Regression (SVR), Random Forest Regression, and a Multi-Layer Perceptron (MLP) neural network. Through cross-validation, Random Forest Regression is shown to have best performance with RMSE of 0.197393 and the most favorable REC curve, while ridge regression is less effective among the four models.

Potential reasons including non-linearity are suggested to explain the difference in model performance, and feature importance in random forest is computed to explore the contribution of parameter. Spot price has the highest importance of 0.558440, followed by air temperature and pressure. Correlation analysis further highlights the crucial role of the day-ahead spot price as a strong baseline indicator under normal market conditions, while also revealing that its predictive value weakens during extreme price events driven by sudden spikes in demand or severe weather.

Overall, the project demonstrates the application of machine learning on predicting energy market and highlights the effectiveness of mainstream regression model in this application setting. However, as the project only limits the energy market data in 1-year, further studies could be conducted by training models with data in extended timeframe. Also, this project focuses solely on application-based machine learning approaches to predicting market behavior, and does not incorporate theoretical studies on energy market operation. Such theoretical analysis could serve as a valuable complement in future work.

References

Saleh, A., Amangeldina, A., Tesfay, W. T., Lafond, S., Rexha, H., & Izhak, O. (2025). Finland aFRR Energy Market and Weather Data (Hourly, June 2024 – March 2025). *Zenodo*. https://doi.org/10.5281/zenodo.17494556