Case Studies 2024

Forecasting the Electricity Price

Comparisons and Conclusions

Professors

Prof. Dr. Matei Demetrescu, Prof. Dr. Paul Navas

Author: Fedi Ghanmi

Group Members

Alicia Hemmersbach, Ketevan Gurtskaia, Lev Luskin

17^{th} July 2024

Table of Contents

1 Introduction

After Starting our forecasts with the load electricity data as the first part of the project, and then progressing to implementing tree methods to forecast electricity returns with proper preprocessing steps, the final part of this research is now reached. Hence, the focus will be on summarizing, comparing and contrasting all modeling forecasts from both previous projects on electricity returns and newly ones introduced in this project. This comparison wil be done using a statistical test called Diebold-Mariano test and will be the main measure to compare a pair of models.

As a first step, forecasts using an autoregressive model of order 1 without any additional predictors are performed, followed by a linear model based solely on each predictor, excluding the lagged returns. The result of these models is plotted and compared respectively with the actual values. Looking into that, the forecast errors of these models were similar and did not vary significantly. However, when plotting the forecasted values, a noticeable difference in the behavior of each predictor was observed. Part of this difference was that the forecasts were not as scaled as the actual returns and exhibited lower variance. These models are then incorporated in the second step and are compared in a pairwise manner using the Diebold-Mariano statistic. It was discovered that the random forest models generally performed better than all other models. In the worst-case scenario, they had equal predictive power but were never worse. The rolling window also performed better across the various types of models considered in this project. In the worst-case scenario, it demonstrated equal predictive power and was never less effective. Lastly, it was proven that a multivariate model does not necessarily lead to better forecasts than a univariate model. This was specific to the decision tree where with multiple variables, it have lost more predictive power.

In the following sections of this report, the initial chapter will detail more on the project objective and describe the data used both for modeling and computing the statistical tests. Following that, the methods employed will be explained and a theoretical background on them will be provided. A specific focus is made on two new concepts in this final part which is both the Diebold-Mariano test and the rolling window. Finally, the empirical findings that were briefly discussed in this introduction will be detailed in its corresponding section and conclusions regarding the capability of these models to make predictions of returns, their disadvantages as well as their potential improvement will be drawn.

2 Problem and Data Description

In this section, firstly the project objectives are outlined with the respective research questions needed to be answered by this study and secondly the data is described.

2.1 Project Objective

The primary objective of this study is to first analyze the predictions of each predictor using a linear model to gain a first insight of how these variables behave in a simple linear model class. Then, among all models, both from the previous project and newly added ones in this report, it is aimed to identify which models performed better and which were less effective in providing return forecasts. Therefore, the first question to address is: Is it possible to assume that random forests will consistently outperform all other models including decision trees in this setting? Additionally, is the rolling window method always superior to the fixed test set for forecasting electricity returns? Finally, do multivariate input features consistently yield better models than univariate ones?

The primary measure used for model evaluation is the the root mean squared forecast error[1]. Its minimization is the key objective while also taking into consideration the risk of overfitting using an out-of-sample set. Another principle measure also used to evaluate a pair of models, as previously mentioned in the introduction, is the Diebold-Mariano statistical test.

2.2 Data Description

In this section, two critical datasets for the project will be described. The first dataframe contains the data used to train the models. The second dataframe comprises the outputs of multiple models summarized for later statistical testing purposes.

2.2.1 Model Training Data

Below in table 2.1 the variables used, their contextual meaning and units are recapitulated. In this final project, only 4 predictors out of the original 8 are selected. These 4 were selected based on their contribution to the highest mean decrease in impurity for both the decision tree and random forest models in the previous project. The restriction was

[1]Abbreviation: MSFE

believed to be necessary because, in a subsequent step, an implementation of a rolling window technique will be performed and using all 8 predictors would significantly reduce the model's speed, making the rolling window process very slow. Hence, a tradeoff was made while taking into account the possible decrease in the forecast accuracy, in which it should be justified due to limited resources. Besides of the feature sample chosen, the original preprocessing steps are preserved. By avoiding changes to the data's behavior, it is ensured that the current work remains comparable.

Name	**Interpretation**	**Units**
datetime_clean	date and time index	Hourly freq.
Price	Electricity prices	Euro(€) / MW
return	Electricity returns	Percentage (%)
solar_forecast	Day ahead solar energy forecast	Megawatt (MW) / Hour
fossil_output	Energy produced by fossil gas	Megawatt (MW) / Hour
geothermal_output	Energy from geothermal sources	Megawatt (MW) / Hour
other_output	Energy produced by other sources	Megawatt (MW) / Hour

Table 2.1: Variables summary of the electricity data

As a brief reminder, the date frame of interest spans from 1^{st} of January 2015 until 15^{th} of March 2024. Since this sample have an hourly frequency it contains $N = 80.688$ observation. After performing the preprocessing like calculating the returns and lagging the variables, the dataset is left containing $N = 80.686$ instances.

The forecasting on this data will be conducted using 2 methods. Firstly using a usual out-of-sample test set that corresponds to 30% of the total data. The remaining 70% is used to train the model. The second method would be a rolling window where the 70% training set is not fixed, but moving with each iteration and a one step ahead forecast is made each time. The complete sequence of individual one-step-ahead forecasts will then constitute the prediction set. More of this method will be explained in section 3.1.6.

2.2.2 Forecast Evaluation Data

Another dataframe used at the last part of this research is constructed from the results of multiple models providing their predictions. In this last part, 20 models were considered resulting in 20 columns and corresponding to the 30% test set (24.206 instances) where predictions were made. Each column represents an executed model and among the 20 columns there exists 4 types of models: an autoregressive model, a linear model both

with indivudual predictors and multivariate, a tree regressor, and a random forest. Each of these 4 models is executed twice: once on a normal fixed set, and once using a rolling window. The table 2.2 below lists the name of each column.

Feature Names
ar_1
ar_1_rolling
lm_solar_forecast_lag_1
lm_solar_forecast_lag_1_rolling
lm_geo_output_lag_1
lm_geo_output_lag_1_rolling
lm_other_output_lag_1
lm_other_output_lag_1_rolling
lm_fossil_output_lag_1
lm_fossil_output_lag_1_rolling
dt_uni
dt_uni_rolling
rf_uni
rf_uni_rolling
lm_multi
lm_multi_rolling
dt_multi
dt_multi_rolling
rf_multi
rf_multi_rolling

Table 2.2: List of column names containing different model forecasts

For better interpretability, an intuitive naming convention across this project was created for each column. The abbreviations are as follows: "ar" for autoregressive, "lm" for linear model, "dt" for decision tree, and "rf" for random forest. "Uni" and "multi" indicate whether the forecast is univariate or multivariate. For the univariate case, it is the lagged return of order 1, and for the multivariate case, it is the lagged return along with the rest of the predictors summarized in table 2.2 which are also lagged. For linear models, each variable is included in the naming of each forecast column. The presence of "rolling" as a suffix indicates a rolling window forecast; otherwise, it is a normal fixed forecast. Let's take an example: lm_fossil_output_lag_1_rolling. This feature corresponds to the prediction set of the univariate linear model with the lagged variable fossil_output on a rolling window basis.

3 Methods

In this part both the statistical methods as well as the relevant tools used as part of this work will be depicted.

3.1 Statistical Methods

In this section, all statistical methods employed in our project are explained with their respective original authors.

3.1.1 Time Series Methods

Autocorrelation Function

The Autocorrelation Function [2] is one of the statistical method used to analyze the correlation between a time series and its lagged values. The autocorrelation function of a time series $\{z_t\}$ at lag k, denoted as ρ_k, is defined as:

$$\rho_k = \frac{\text{Cov}(z_t, z_{t-k})}{\sqrt{\text{Var}(z_t) \cdot \text{Var}(z_{t-k})}} \tag{3.1}$$

In equation 3.1 if ρ_k is close to 1, it indicates a positive correlation between the two terms. If it is close to -1, it suggests a negative correlation. Eventually, if it is near 0, it implies little to no correlation. The correlation terms are then plotted in a correlogram that will provide insights to the behavior of the time series. See (Brockwell and Davis, 2016). If the plot shows continuous spikes and regulard intervals, it may be an indication of seasonality.

Partial Autocorrelation Function

The Partial Autocorrelation Function[3] is a continuity of ACF and specific for identifying the order of autoregressive models (Enders, 2004). It is done by not taking into account the lags information between two terms and only including their direct influence (Brockwell and Davis, 2016).

$$\phi_{n,n} = \frac{\rho(n) - \sum_{k=1}^{n-1} \phi_{n-1,k}\rho(n-k)}{1 - \sum_{k=1}^{n-1} \phi_{n-1,k}\rho(k)} \tag{3.2}$$

[2] Abbreviation: ACF
[3] Abbreviation: PACF

In equation 3.2 $\rho_{(n)}$ is the previously calculated autocorrelation in equation 3.1 according to (Durbin, 1960) and is substracted from previous lags depending on the choice of k. The coefficient $\phi_{0,0}$ is usually initialized to 1. A significant spike at lag k in the PACF plot suggests that lag k is a meaningful lag in the autoregressive model. If the PACF cuts off [4] after lag p it then suggests an AR(p) model (Enders, 2004).

Autoregressive Process

Autoregressive processes [5] are models that are used to describe a time series using its own past values (Eshel). As written in equation 3.3, an autoregressive process of order p is written as the sum of each lagged dependant variable multiplied by a coefficient:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-k} + \epsilon_t \tag{3.3}$$

Autoregressive models usually assume non-seasonal data. In this case, one method to estimate p order of the AR process is by using the Yule-Walker Method (Kirchgässner and Wolters, 2007) .

3.1.2 Model Evaluation and Selection

Ordinary Least Squares

The Ordinary Least Squares [6] is a fundamental concept usually used in linear regression settings to estimate the parameters of a linear relationship between a dependent variable and one or more independent variables as it is seen in equation 3.4:

$$y = X\beta + \epsilon \tag{3.4}$$

The OLS method estimates the parameters β by minimizing a loss function. The result of this optimization is analytical and can be written in matrix format defined as in equation 3.5 where X is the so called design matrix and y the true values (Goldberger, 1964):

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T y \tag{3.5}$$

[4] All partial autocorrelations after lag p are not statistically different from zero
[5] Abbreviation: AR
[6] Abbreviation: OLS

One example of loss functions used in this study in the Mean Squared Forecast Errors [7], see equation 3.6, which is a loss function that can also be used beyond the OLS method to evaluate any model performance.

$$MSFE_z = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(z_t - \hat{z}_t)^2}. \tag{3.6}$$

The R^2, equation 3.7, is also employed as a part of a feature selection process to measure the goodness of fit (Casella, 2002) .

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.7}$$

Akaike Information Criterion

Information criteria are a type of model selection criteria usually used when the test data is not available or expensive to obtain. One widely used criterion for model selection is the Akaike Information Criterion[8]. Mainly the AIC is defined in equation 3.8 below:

$$\text{AIC} = 2k - 2\ln(\hat{L}) \tag{3.8}$$

Here, the AIC measures the goodness of fit taken by the maximized logarithmic likelihood function written as $\hat{L}$ and tries to balance it with the model's complexity using k number of parameters (Akaike, 1974). Both sides calibrate each other which reduces overfitting risk that originates from increasing the model capacity (James et al., 2013).

3.1.3 Tree Methods

Regression Trees

Decision trees are a type of supervised machine learning technique that follow the intuition of real-life trees as seen in the abstract fig. 3.1 below.

Mathematically, they model the relationship between independent variables to determine the value of a dependent variable. A split is the process of dividing the data into subsets based on a specific condition related to the independent covariates which is in this

[7] Abbreviation: MSFE
[8] Abbreviation: AIC

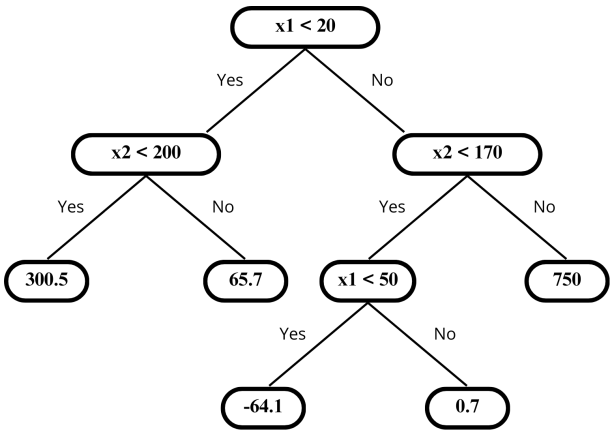


Figure 3.1: Simple abstraction of a regression tree on two random variables

example x_1 and x_2. Each split within the tree results in an internal node. These nodes can further split, leading to additional internal nodes. The final splits, which contain the last sampled observations, are referred to as leaf nodes and represent the predicted values of the dependent variable. This process of splitting aims to accurately predict the dependent variable. The topmost node in a tree is called a root node and represent the entire dataset before any splitting is made. The term regression trees was first introduced by (Breiman et al., 1984). They are an example of decision trees where each leaf node represents a continuous target and the prediction at each one of them is the average of the observations at that level. At each node, the algorithm chooses the split that minimizes some loss measure. The root mean squared forecast error can be considered in this case. As already mentioned, the tree construction starts with the root node and splits it into subsets that are as similar as possible. The splitting for a variable j at a split point s can be formulated as below. See (Hastie et al., 2009):

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right]. \tag{3.9}$$

The goal of eq. (3.9) is to minimize the squared difference between the observed values y_i and their means c_1 and c_2 after the split and to obtain $R_1(j,s)$ and $R_2(j,s)$ which are the two sets created by splitting on variable j at point s (Hastie et al., 2009). To avoid overfitting, the growth of the tree can be constrained by a stopping criteria such as minimum node size or maximum tree depth (Breiman et al., 1984). The minimum node size is a hyperparameter chosen by the modeller that dictates the smallest number of observations a node must have before a split is attempted. This prevents the creation

of nodes with small number of instances that could lead to less generalization [9]. The maximum tree depth is the maximum number of layers a tree can grow. It serves as an upper bound to prevent the tree from becoming too complex and modeling noise in the data. Many other criteria also exist and are covered in (Hastie et al., 2009).

Random Forests

The Random forests method is an ensemble learning method that extends the simple case of a single tree as seen in section 3.1.3. It operates by constructing multiple decision trees where each one is no longer fitted to the entire dataset but only to a specific bootstrap sample (Breiman, 2001). A bootstrap sample is a sampling technique with replacement. This means that each instance is randomly selected and can appear multiple times within each sample (Efron, 1979). These samples are then used to train each tree, and predictions are generated based on them. This can also be referred to as the bagging technique. For the case of regression, the algorithm averages the predictions from all individual B trees as eq. (3.10) shows:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x') \tag{3.10}$$

What characterizes random forests as an ensemble method is its ability to reduce the variance through both the bagging process and the final averaging of the trees. For the tunable parameters, the number of $n\_estimators$ or trees to be selected is unlike the maximum tree depth parameter where when it is excessively large would lead to overfitting. On the contrary, the larger the number of trees ($n\_estimators$) the more accurate the prediction will be without any risk of overfitting. This is because the model will eventually average all outputs as eq. (3.10) depicts. This concept is mathematically summarized in eq. (3.11) according to (Breiman, 2001).

$$E_{X,Y} \left(Y - av_k h(X, \Theta_k)\right)^2 \rightarrow E_{X,Y} \left(Y - E_{\Theta} h(X, \Theta)\right)^2 \tag{3.11}$$

More clearly eq. (3.11) states that as the number of trees in a random forest increases to infinity, the expected residuals of the averaged prediction av_k from a finite number of k trees obtained from the function $h(X, \Theta)$ converges to the expected residual of the ideal

[9] Generalization is simply the goal to allow the model to perform effectively in an unseen data, hence to reduce the testing error

average of the prediction over all possible tree parameters Θ. One constraint that should be mentioned is that this comes with an increased computational cost. As the number of trees grows, the computational expense increases, and the improvement in accuracy diminishes, eventually becoming negligible compared to the extra computational time required.

Feature Importance in Tree Methods

What tree methods have special that other methods do not is their ability to identify and return the most important features considered by the model during the fitting process. This is also called an explainable model. The measure used in this project, which is also used in the case of general regression tree models is the mean decrease in impurity (Gini Index) and is formulated for random forests according to (Breiman et al., 2017) as the eq. (3.12) below:

$$\text{unnormalized avg importance}(x) = \frac{1}{n_T} \sum_{i=1}^{n_T} \sum_{\text{node } j \in T_i | \text{split variable}(j)=x} p_{T_i}(j) \Delta i_{T_i}(j), \quad (3.12)$$

For a feature x, the sum is taken over all trees T_i considering all the nodes where feature x was used to make a split (split variable$(j) = x$) and for each of these nodes, the importance is calculated by multiplying the proportion of samples in the node $p_{T_i}(j)$ with the impurity decrease $\Delta i_{T_i}(j)$ of that node. What is meant by impurity decrease is to actually have final nodes with the least variance possible, such that they are homogeneous and reflects similar properties.

The feature importance is also of a similar concept in simple regression trees except that the importance is now no longer calculated over all the trees in the forest but for one single tree. However, one of the many limitation in the impurity-based importance is that they are derived from the training set. As a result, they may in some cases not necessarily indicate how effectively the variables will perform in making predictions that generalize well on the test set (Breiman, 2001). But as an approximation, when identifying these crucial variables, one can reduce the dimensionality of the data which can even lead in some cases to more accurate models due to the noise elimination. In this project, the feature importance scores generated using Python packages are presented as percentages

to facilitate comparison. They are in more sense the normalized reduction in impurity from each feature over the total reduction.

3.1.4 Winsorization

According to (Dodge, 2003) winsorization was named after Charles P. Winsor (1895-1951). It is a method that is used to handle extreme data values and reduce the effect of huge outliers. An $x\%$ winsorization replaces values below the lower $\frac{(1-x)}{2}$ percentile and above the upper $x+\frac{(1-x)}{2}$ percentile with the values at these respective percentiles. For example, a 90% winsorization replaces observations below the 5^{th} percentile with the 5^{th} percentile value and observations above the 95^{th} percentile with the 95^{th} percentile value itself.

3.1.5 Diebold-Mariano Test Statistic

The Diebold-Mariano test is a statistical method used to compare two forecasts, determining whether a forecast was actually better or the lower error measure was just due to random variation that is not significant enough (Diebold and Mariano, 2002). TThis inference is made using the DM statistic, which tests the hypothesis that the expected loss differential is significantly different from zero. Mathematically, it is formulated as eq. (3.13) below (Diebold, 2012) :

$$\begin{aligned} H_0: &\quad E(L(e(F_{1t})) - L(e(F_{2t}))) = 0 \\ H_a: &\quad E(L(e(F_{1t})) - L(e(F_{2t}))) \neq 0. \end{aligned} \tag{3.13}$$

Here, $e(F_{1t})$ and $e(F_{2t})$ refers to the computed residuals respectively between forecasts of model 1 and model 2 with the actual values. These residuals are mathematically defined as $e(F_{it}) = y_t - \hat{y}_{it}$. The loss function observed in eq. (3.13) is a transformation of these residuals. The most commonly known used one in real world applications is the squared error loss[10] taken as $(e_t)^2$. The DM test is then formulated as:

$$DM_{12} = \frac{\bar{d}_{12}}{\hat{\sigma}_{\bar{d}_{12}}} \tag{3.14}$$

In eq. (3.14), the numerator shows the mean loss differential where it is calculated as $d_{12} = \frac{1}{T}\sum_{t=1}^{T} d_{12t}$ which is the average across T instances of the loss differential where

[10]It is most known under the name of MSE or mean squared error when it is computed accross all values of the dataset.

$d_{12t} = L(e(F_{1t})) - L(e(F_{2t}))$ as written in the hypothesis test (Diebold, 2012). For the denominator, the estimated standard errors should be calculated in a "robust" way (Diebold, 2012) since the loss differentials may be serially correlated and that would violate the test assumption of covariance stationarity. Hence, in the original paper, the authors define this robust standard deviation as follows:

$$\hat{\sigma}_{\bar{d}_{12}} = \sqrt{\frac{2\pi \hat{f}_d(0)}{T}} \tag{3.15}$$

In eq. (3.15) $\hat{f}_{d(0)}$ is identified as the estimated spectral density of the loss differential at frequency 0 and is a consistent estimate of $f_{d(0)}$ expressed as $f_d(0) = \frac{1}{2\pi} \sum_{\tau=-\infty}^{\infty} \gamma_d(\tau)$ (Diebold and Mariano, 2002). This asymptotic formula would only serve to shows that the adjustment of the serial correlations violation can be significant even if the loss differentials exhibit only a weakly correlation. To use it in real practices, the authors go from an asymptotic definition to express it now as a weighted sum of the available sample autocovariances as shown below:

$$2\pi \hat{f}_d(0) = \sum_{\tau=-(T-1)}^{T-1} \mathbf{1}\left(\left|\frac{\tau}{S(T)}\right| \leq 1\right) \hat{\gamma}_d(\tau) \tag{3.16}$$

Where

$$\hat{\gamma}_d(\tau) = \frac{1}{T} \sum_{t=|\tau|+1}^{T} (d_t - \bar{d})(d_{t-|\tau|} - \bar{d}) \tag{3.17}$$

For each lag τ during the summation in eq. (3.16) the autocovariance in eq. (3.17) is invoked to also sum across the terms between $|\tau| + 1$ and T. However, not all $\hat{\gamma}_d(\tau)$ terms will be summed in the spectral density since the lag window function $\mathbf{1}\left(\left|\frac{\tau}{S(T)}\right| \leq 1\right)$ will return 1 only if the absolute value between the lag τ and the truncation lag $S(T)$ is lower or equal than 1 and 0 otherwise. Obviously, this fraction determines the number of autocovariances to include in the estimation. The application of this rule is pertinent to the Heteroscedasticity and Autocorrelation Consistent estimator[11], such as the Newey-West estimator (Newey and West, 1994). In order for this estimator to be consistent, the truncation lag $S(T)$ should be defined as a function of the sample size T. A known rule of thumb used in general econometrics applications is to set $S(T) = 0.75T^{1/3}$ (Newey and

[11] Also known as HAC Estimator

West, 1994). This rule of thumb would also be used in this research study. To compute the significance of a DM statistic, the p-value is used. Since in large samples the test statistic is standard normal under certain assumptions (Diebold and Mariano, 2002), the student-t cumulative distribution function is used to this end. If the p-value is lower than a chosen significance level (e.g., 5% for a 95% confidence interval), then the null hypothesis is rejected, concluding a significant statistical difference between the loss differentials.

After rejecting the null hypothesis, the output of the DM statistic can be interpreted in two ways. If it is positive, it indicates that the loss of the forecasts from the first model F_{1t} is higher compared to the second model suggesting that the forecasts from the second model are closer to the actual values. Vice versa, if the DM statistic is negative, it indicates that $L(e(F_{1t})) - L(e(F_{2t})) < 0$, meaning $L(e(F_{1t})) < L(e(F_{2t}))$. Hence the errors from the first model are lower than those from the second model.

As mentioned in the beginning, the DM test should be used to compare forecasts since the errors are extracted from the forecasts themselves and not from the models (Diebold, 2012), however this did not prevent multiple research work later on to use it as a model comparison technique. This could arguably be possible if one assumes that "the forecasts are from fully-articulated econometric models that are claimed to be known to the researcher" (Clark and McCracken, 2001). Nevertheless, the original authors of the Diebold-Mariano test digress in this topic and consider it somehow sub-optimal (Diebold, 2012).

3.1.6 Rolling Window

Unlike the traditional fixed train-test splits used in forecasting, the rolling window technique involves a continuously shifting training set where with each step forward the training sample shifts ahead by one period. The rolling window then updates it by adding new observations that were used in the previous step for testing. Mathematically, the objective can be simply defined as to forecast $\hat{y}_{t+1}$ using information available until time t. This information can be summarized in a window W that contains instances from i until $n - w + 1$ where w is the window size. Mathematically, it is written as eq. (3.18)

below:

$$W_i = x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+w-1} \tag{3.18}$$

$$\text{where } 1 \leq i \leq n - w + 1.$$

This rolling principle helps the model to constantly update to new observations and make it more adaptive. However, some scientists argue that it may not be suitable for all sets of problems (Inoue et al., 2017). This is completely acceptable, as the no free lunch theorem suggests that an approximate model will never fully capture the entire truth of the underlying process.

3.2 Relevant Tools

As part of the work, Python version 3.11 (Guido van Rossum, 1989) with its open source development environment Jupyter Notebook (Jupyter, 2014) were used to conduct our study. External packages used are summarized in table 3.1 below.

Package	**Version**
pip3	23.0.1
Jupyter Notebook	6.5.2
Pandas	1.5.1
matplotlib	3.7.1
seaborn	0.12.2
statsmodels	0.13.5
numpy	1.23.4
scikit-learn	1.5
mlxtend	0.23.1
tqdm	4.66.4
scipy	1.10.1
time	*
collections	*
datetime	*
warnings	*

Table 3.1: List of Python packages used across the project

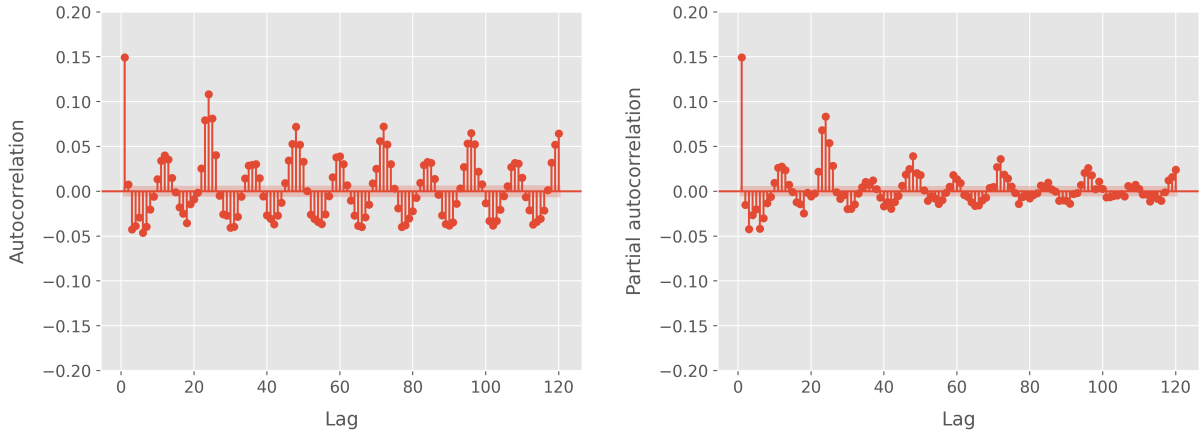
* Packages that are a part of python core distribution and hence do not have a specific version.

4 Empirical Results

The results obtained from the practical implementation of this final part of the study are detailed in this section and are divided into parts.

4.1 Univariate Autoregressive Model with Lagged Returns

In this part, an autoregressive model is considered for the electricity returns. To determine the appropriate order p, the ACF and PACF of the returns are plotted in fig. 4.1 below.



(a) ACF Plot

(b) PACF Plot

Figure 4.1: Autocorrelation and partial autocorrelation plots of returns of electricity between 2015 and 2024 in Germany

Unlike the electricity load, the returns exhibit lower autocorrelation values, with the highest term reaching only 0.15 in this case. Seasonality is also observed in fig. 4.1a accompanied with frequent positive and negative spikes. The most significant one in the PACF plot is the first lag, suggesting that $p = 1$ could be considered. Based on these informations, different AR processes are fitted, each time including an additional lag to the model while measuring its AIC. The fig. 4.2 is obtained.

It is observed that the AIC continues to decrease as the lag is increased, even beyond a lag of 100. This issue is common when using certain information criteria with time series data. In this example, the substantial gap between the number of observations and the number of parameters in the AR model is so large that the AIC continually suggests increasing the model's complexity. However, this may not necessarily be the best approach. Additionally, there is usually a noticeable drop in AIC after every 24 lags, corresponding to one day of returns. However, these drops become progressively less

significant as the number of lags increases. Extending the lags further in the figure would eventually show a rise in AIC. However, in this research, a maximum limit of 10 lags will be applied for the sake of comparability, so no need for further extension of the lags of the graph [12]. Since the AIC does not provide additional value in this context, the choice of lag p will be determined using the PACF plot in fig. 4.1.

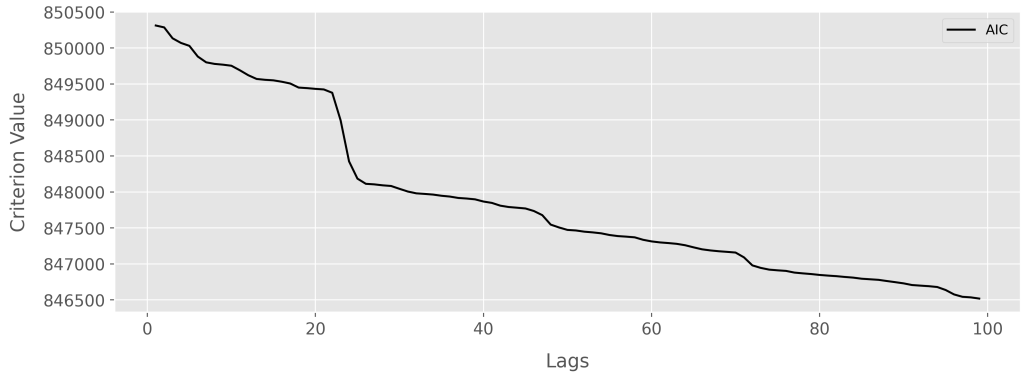


Figure 4.2: Variation of AIC across different lags for the autoregressive of first order model AR(1)

After the decision to take $p = 1$ as our AR model, an $RMSE = 55.49$ is obtained. A uniform sample across the data of both the true values and the predictions set is obtained and are plotted to observe the difference between the actual values and the forecasted ones as the fig. 4.3 below shows.

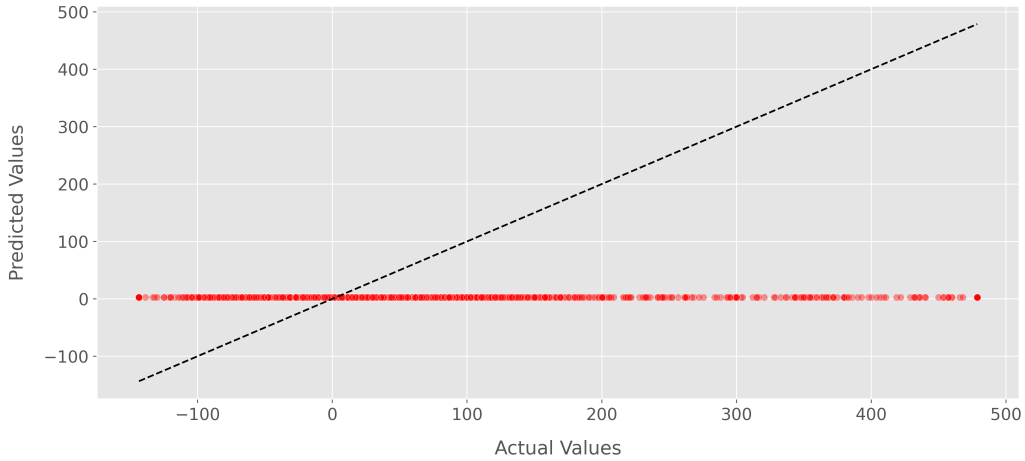


Figure 4.3: Scatter plot of actual electricity returns against predicted ones between January 2021 and March 2024 for the AR(1) model

If points happens to be exactly at the dashed black line, it means that a perfect forecast

[12] AIC is typically used when test data is expensive to obtain or not available. However, in this case, we have sufficient test data, so it is assumed that disregarding it will not have a significant negative impact.

was made, if not, then it would be observed how far was that instance from the actual result. It appears that previous values of the returns are not significantly influencing the current values since the red dots remain almost constant across the x-axis. This might not be entirely accurate, as fig. 4.1a indicates that the returns exhibit seasonality, which could explain the observed behavior. Additional preprocessing steps may perhaps be required for future research to achieve better forecasts. This pattern also persists even when considering other p lags, so changing the autoregressive order won't change much.

One other approach to observe the fluctuations of forecasted instances over time relative to the actual returns is by averaging them using a rolling window, as demonstrated in the fig. 4.4 for this AR(1) case.

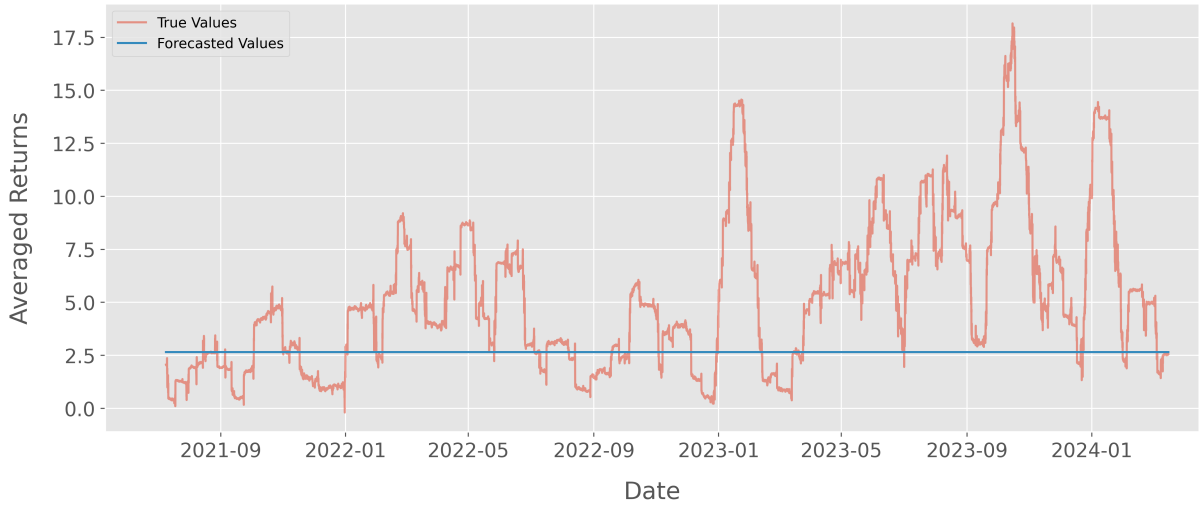


Figure 4.4: Plot of the Averaged Actual Electricity Returns Against Predicted Ones Between September 2021 and March 2024 for the AR(1) Model

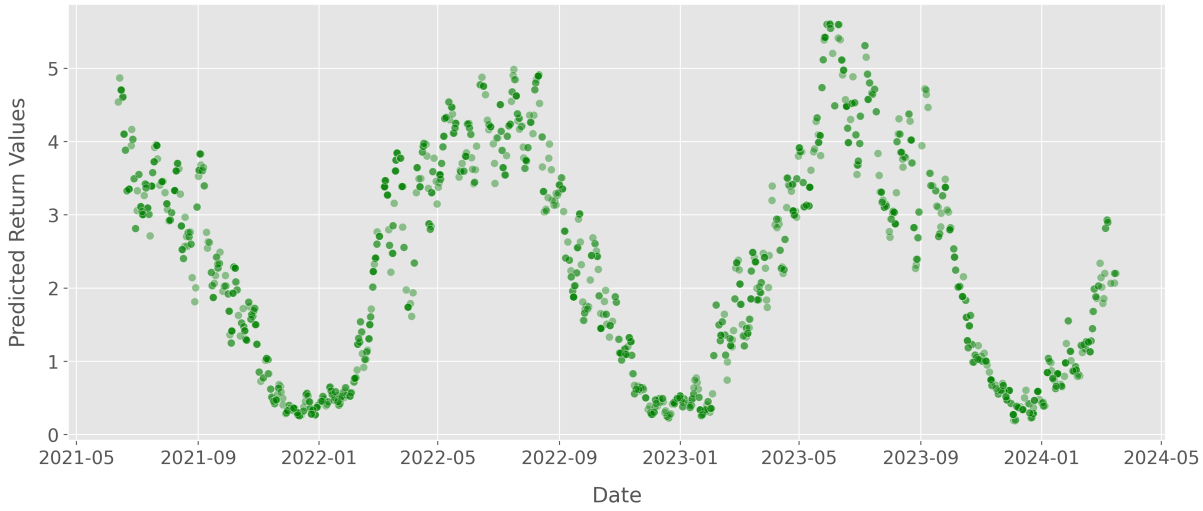
There, we can observe that the averaged values of the forecasted returns remain almost constant. In this case, it does not significantly alter the conclusion compared to the previous fig. 4.3 [13]. Throughout this project, including other similar figures, a smoothing window size of one month $W = 762$ hours will be used.

4.2 Univariate Linear Model With Other Predictors

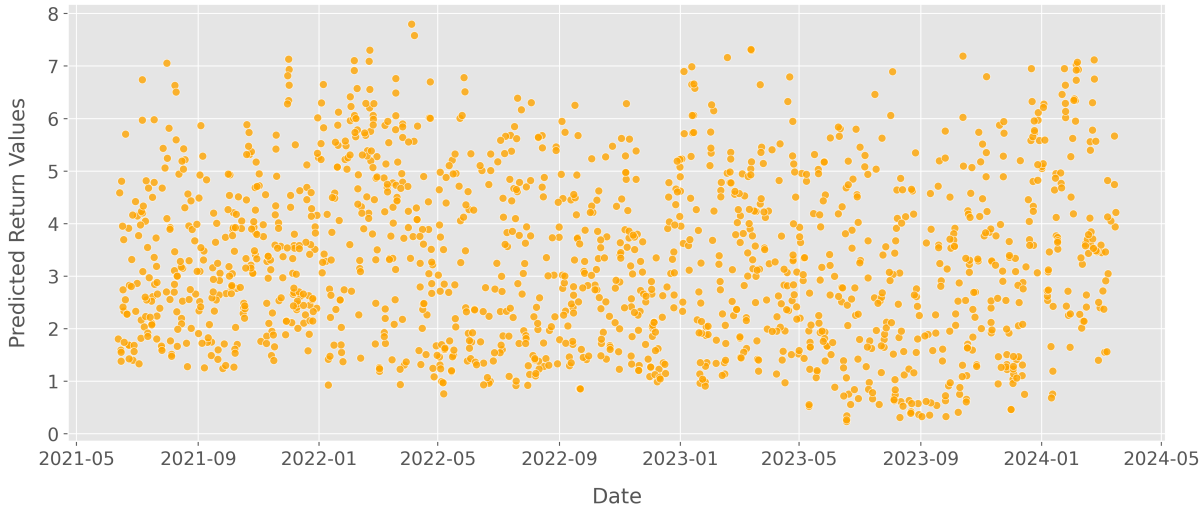
The focus is altered in this section, where each lagged predictor will be selected to be fitted individually to assess their predictive powers. The four predictors of solar forecast, other

[13] This was only an exception when working with a simple lagged return. It will be discovered later that with the inclusion of other predictors, the predictions exhibited more variance, though still insufficient for direct comparison with the actual values. This is why the rolling averaged graph was proposed initially.

output, geo output and fossil output led to an RMSE of respectively 55.52, 55.51, 55.32 and 55.65. Interestingly, the RMSE values are closely similar to that of the autoregressive setting and also do not vary much between each other. However, if the individual instances for these univariate linear models are plotted using the same previous uniform sample, it becomes clear that they exhibit noticeable variances, unlike the AR model constant predictions. These values have a small range of fluctuation, which is shown for example in both fig. 4.5 and fig. 4.6.

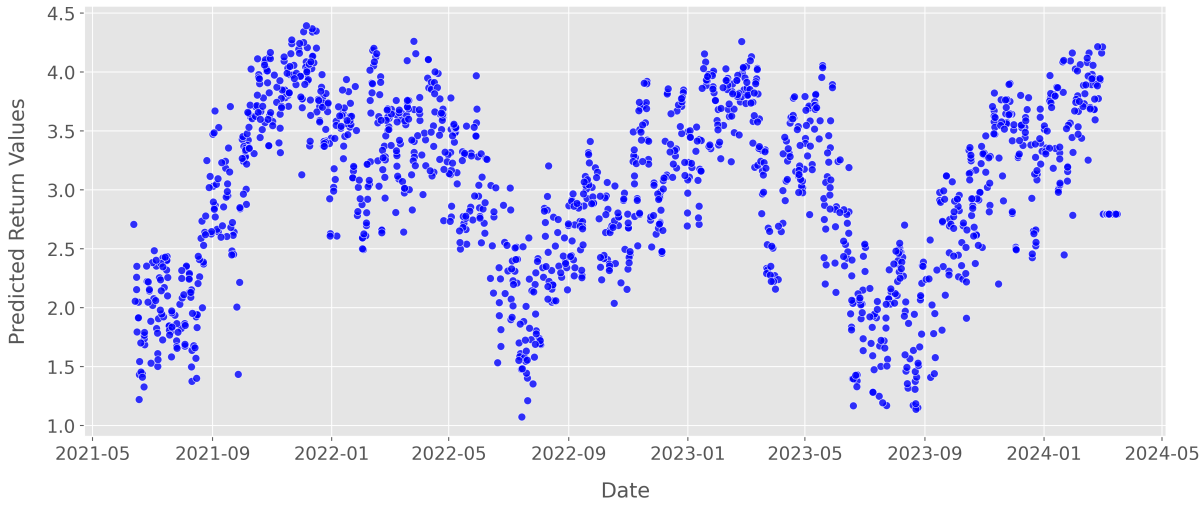


(a) Predicted scatter points of electricity returns using the solar forecast variable over time

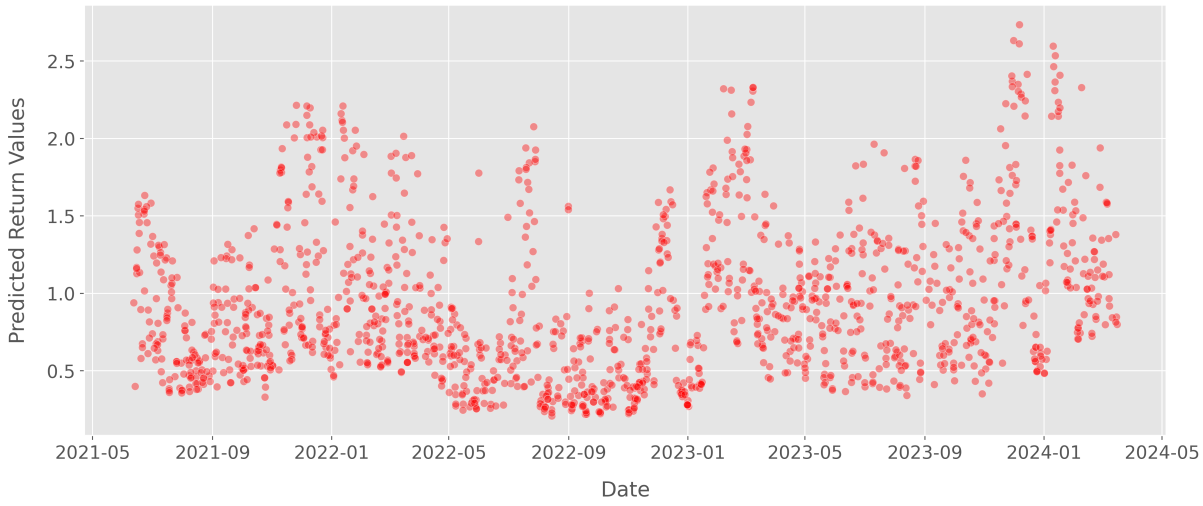


(b) Predicted scatter points of electricity returns using the other output of electricity variable over time

Figure 4.5: Scatter plot of Forecast of Returns Across Time Using Various Predictors - Part 1



(a) Predicted scatter points of electricity returns using the geo output variable over time



(b) Predicted scatter points of electricity returns using the fossil output variable over time

Figure 4.6: Scatter plot of Forecast of Returns Across Time Using Various Predictors - Part 2

The lowest variance is observed in fig. 4.6b with the fossil variable where values are bounded between zero and 2.6 approximately. Then fig. 4.6a for the geo_output where it reaches 4.5 as an upper bound. 5 for fig. 4.5a for the solar_forecast and finally a value of 8 in fig. 4.5b for the other_output. Since the actual values have a larger scale than the predicted values, plotting them with the forecasted ones, as done in the previous figure, will be ineffective. This is because the difference in the ranges between them will make the forecasted instances appear constant when displayed on the same axis and no information can be derived from such plots. To narrow this gap, the same rolling window size used for the prediction plot of the AR model is chosen, and the averaged predictions from each predictor are plotted against the averaged actual values across time. It is totally

acknowledged that plotting real, non-averaged instances and comparing them directly would be more insightful. However, due to the simplicity of our models, the returns would typically exhibit low ranges. Therefore, it is decided to proceed with the averaged plots.

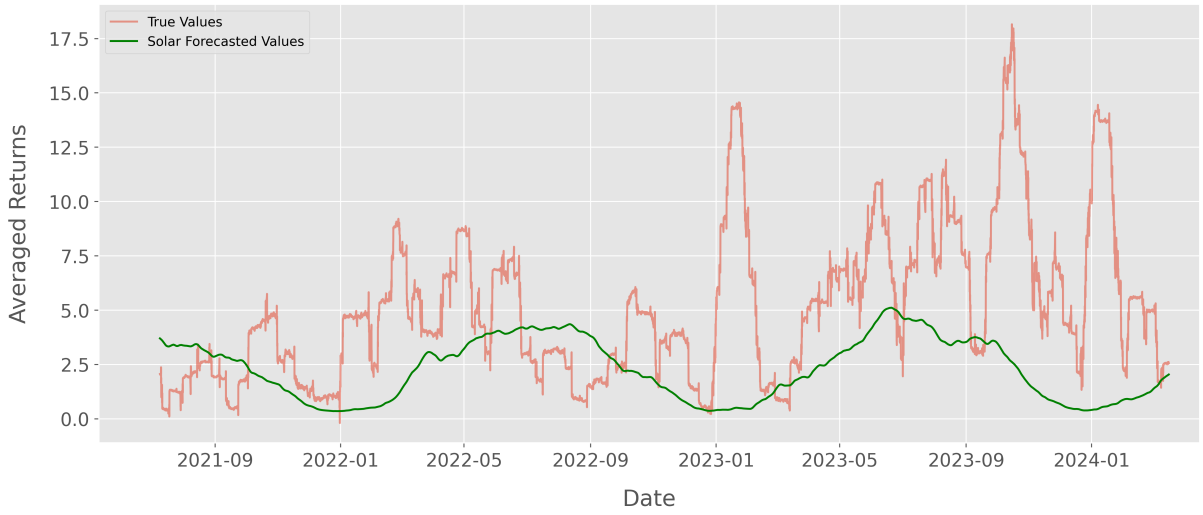


Figure 4.7: Plot of the averaged actual electricity returns against predicted ones between January 2021 and March 2024 using the simple linear model with the solar forecast variable

When the returns are summed and averaged, the scales of both variables become closer, making the plots more interpretable. With the solar forecast in fig. 4.7, the behavior of the forecasted returns over time exhibits a seasonal pattern, likely influenced by the seasonality of the predictor itself. On average, lower electricity returns are predicted for the winter period, while higher returns are estimated for the summer period.

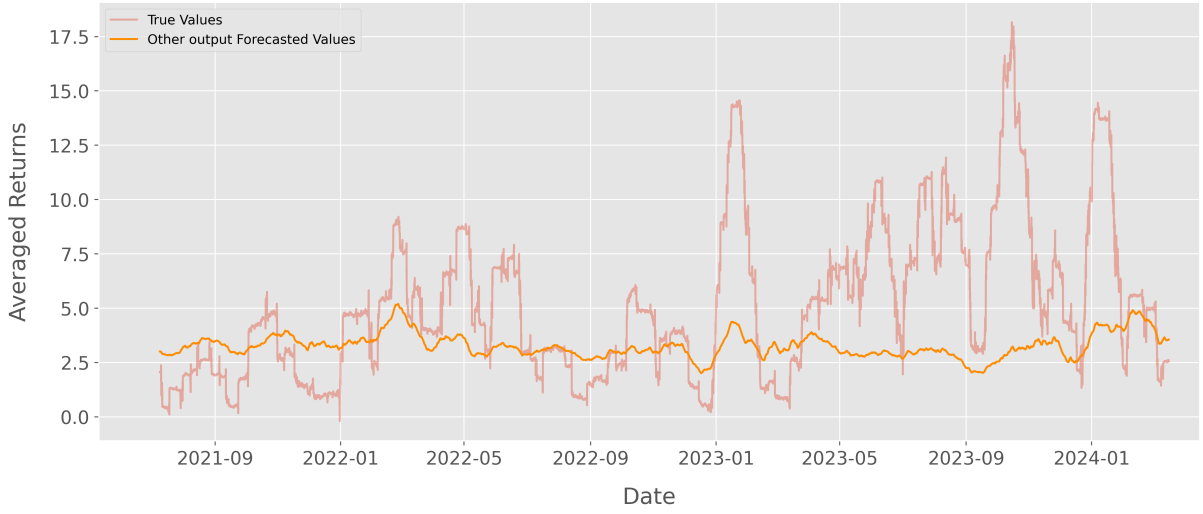


Figure 4.8: Plot of the averaged actual electricity returns against predicted ones between January 2021 and March 2024 using the simple linear model with the other output variable

For the other output variable in fig. 4.8, the estimated returns exhibit less seasonality but more closely follow the actual fluctuations of the returns, despite being on a smaller scale. For instance, between January and May 2022, both curves appear to exhibit similar upward and downward deviations.

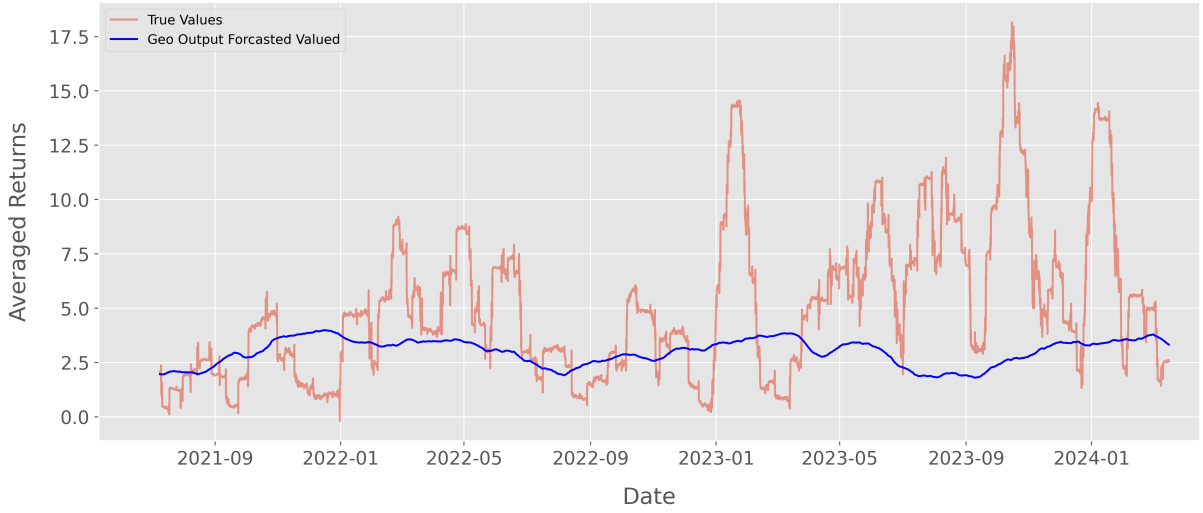


Figure 4.9: Plot of the averaged actual electricity returns against predicted ones between January 2021 and March 2024 using the simple linear model with the geo output variable

In fig. 4.9, the geo output feature has, on average, less deviation compared to the solar forecast and follows the patterns of the real returns less closely than the other output feature. During the same period previously chosen (between January and May 2022), the electricity forecast remains nearly constant.

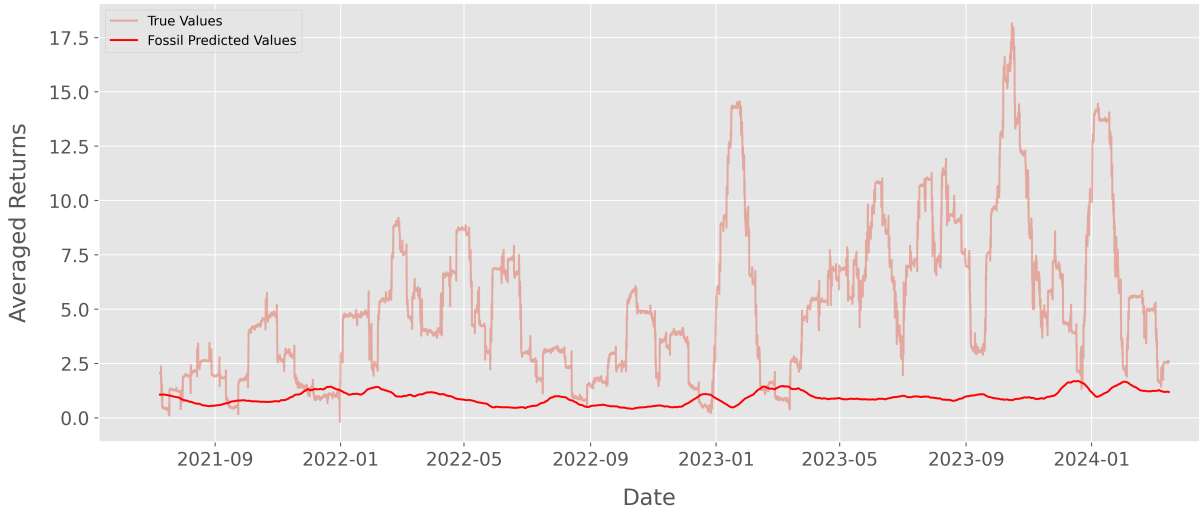


Figure 4.10: Plot of the averaged actual electricity returns against predicted ones between January 2021 and March 2024 using the simple linear model with the fossil output variable

In fig. 4.10, the fossil output provides return estimations with the highest error mea-

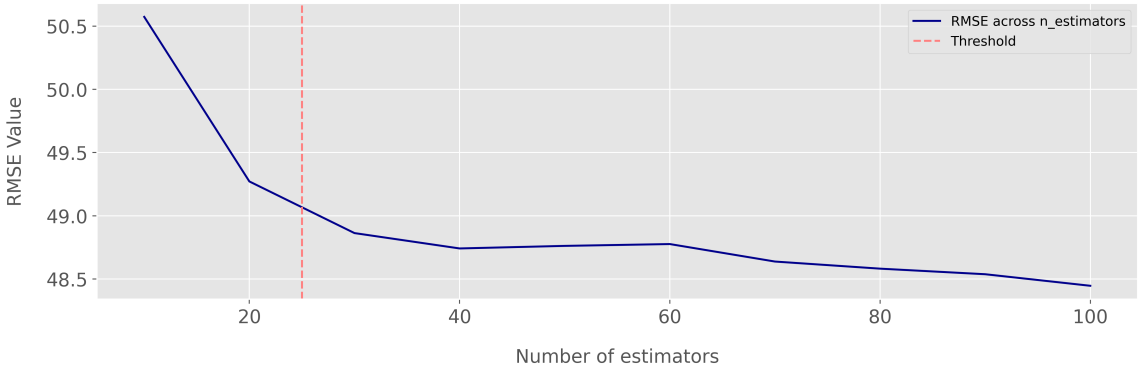
sure compared to other predictors, and also exhibits the most stable and least deviating predictions among them. Even the level of the curve is much more scaled down compared to the prior predictors. The previous 3 variables usually had an average return variation between 5 and 2.5, but the fossil output consistently shows an average return below 2.5.

As previously mentioned, the RMSE values did not vary significantly between the models due to the small ranges of predictions. However, when averaged, the differences in behavior among the predictors become apparent. This averaging acts as a form of up-scaling the return predictions. In future research, with more focused data preprocessing, averaging might not be necessary, and the gap between the actual and forecasted returns would be reduced.

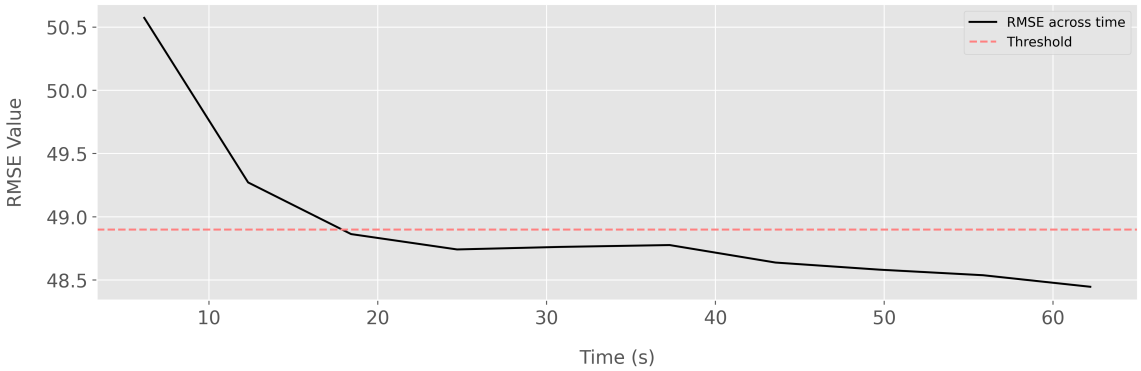
After analyzing each forecast, now it is proceeded to incorporate additional models. This includes tree methods from the previous project applied to both univariate and multivariate data, as well as new models from this project that encompass both the aforementioned univariate forecasts and new multivariate ones. The outcome of this process is a dataframe containing predictions of 20 models summarized in 20 features. For a clearer understanding of the naming conventions for each feature and the respective model behind it, please refer to section 2.2.2

4.3 Random Forest Parameter Tuning

When running the rolling window, the random forest boosting method, which uses different trees to reduce variance, may cause the rolling window to take an excessive amount of time. Two options were considered to address this issue: The first option was to reduce the test set, arguing that the reduced set would be representative enough of the whole test data. However, this option was not preferred for several reasons. One major concern was that implementing the Diebold-Mariano test later would be difficult to interpret because the results would not have the same test size as other models. The second, more preferred option, was to tune the $n\_estimators$ parameter of the random forest to reduce the necessary computational time while keeping the test set unchanged. When resource constraints exist, it is important to balance the trade-off as much as possible, which is why the second option was chosen. Specifically, to determine the optimal value for $n\_estimators$, the runtime of the random forest was plotted across different $n\_estimators$ while balancing the forecast error measure. This resulted in fig. 4.11 below.



(a) Variation of the root mean squared error of the mulitvariate random forest model as a function of the number of estimators



(b) Variation of the root mean square error over time as the number of estimator increases in the Random Forest model

Figure 4.11: Combined variation plot of RMSE over number of estimators and RMSE over time

In fig. 4.11a, it can be observed that as the number of trees on the x-axis increases, the RMSE correspondingly decreases. However, beyond a certain point, the decrease in RMSE becomes more stable compared to the initial phase (before the red dashed threshold line and after it). Choosing $n\_estimators = 25$, which corresponds to the point where the error rate begins to stabilize, results in a significant reduction in runtime. As shown in fig. 4.11b, the runtime decreases from approximately 60 seconds for $n\_estimators = 100$ to around 18 seconds, representing nearly a 70% improvement in the time required to run the random forest. This change is considered worthwhile, despite the slight increase in the error measure.

For the remainder of this empirical analysis, a random forest model with the hyperparameter $n\_estimators = 25$ will be used across all combinations, both with and without the rolling window, in order to ensure comparability. It is also worth noting that $n\_jobs = -1$ is used when running the model in Python, which utilizes all CPU cores of

the machine to parallelize the jobs across the trees during fitting.

4.4 Implementation of the Rolling Window

Before proceeding with the pairwise Diebold Mariano test, we first extend our previous models with the new rolling window forecast. For all models the table 4.1 below is obtained.

Model	RMSE	RMSE *
ar_1	55.493	55.464
lm_solar_forecast_lag_1	55.521	55.506
lm_geo_output_lag_1	55.506	55.494
lm_other_output_lag_1	55.319	55.263
lm_fossil_output_lag_1	55.646	55.644
dt_uni	56.172	57.928
rf_uni	56.211	52.4774
lm_multi	54.310	53.794
dt_multi	66.309	55.486
rf_multi	49.023	48.272

Table 4.1: Error comparison of models forecasts both without the rolling window (RMSE) and with the rolling window (RMSE *)

As previously mentioned, the naming convention that was set for all these models is explained in section 2.2.2. As the title of the table 4.1 indicates, it is written as (RMSE*) the column with the root squared forecast error obtained with the rolling window and (RMSE) the error rate without the rolling technique. The initial analysis revealed that, for almost all models, the RMSE of their predictions increased when implementing a rolling window approach compared to a fixed test set. The only exception was the decision tree model (highlighted in yellow) using only lagged returns, where the rolling window RMSE* (57.928) was greater than the fixed test set RMSE (56.172). This suggests that the rolling window technique does not necessarily lead to better forecasts.

The most significant improvement was moreover observed in the multivariate decision tree model, where implementing an iterative one step ahead forecast resulted in a 16% increase in predictive performance compared to the fixed test set. Other models showed a slight decrease in RMSE. However, as observed in section 4.2, even small changes in RMSE can lead to significant differences in forecast variation when comparing predictive instances to actual returns. Therefore, evaluating model errors at this stage is only an

initial step. A more comprehensive comparison using a statistical test will be conducted later on.

In both the non-progressive and progressive forecast, the random forest model achieved a better RMSE than all other models in the multivariate case. For the univariate case, it only proved better under the dynamic forecast set. In the constant test set it surprisingly performed worse than all other models. This highlights that it is not necessarily evident that random forests will always perform better. Lastly, transitioning from a univariate to a multivariate model, almost all models showed improved forecast results in both forecast settings. However, the decision tree in the fixed test set was an exception, as its performance actually worsened. This may indicate that adding more variables does not always lead to better forecasts.

In this section, forecasts were compared using their direct error measures. A more advanced method to compare them is the Diebold-Mariano test, which is discussed in detail in section 3.1.5. Now, our previous observations will be reinforced with a statistical perspective.

4.5 Implementation of the Diebold-Mariano Test

Before determining which model is better, we first check whether the differences between their forecasts is significant or not. The heatmap fig. 4.12 below shows the $p - value$ result of the pairwise comparison between all of our models.

When the intersection between two models is red, it indicates that the p-value of the respective DM test is higher than the 5% significance level. When the p-value exceeds this threshold, the test statistic concludes that the difference is not statistically significant, suggesting that the two models have approximately the same predictive power. Typically, non-significance is observed on the upper border of the heatmap because models placed closer to each other have similar model complexities, which is where non-significance can be expected. This pattern holds true in this visualization, except for two occurences: the insignificant variance in forecasts of the dt_multi_rolling with both the lm_solar_forecast_lag_1_rolling and the lm_geo_output_lag_1_rolling. This suggest that, for the rolling test set, the decision tree performed similarly to the solar forecast and geo output features when each was fitted individually in a linear model.

Earlier, it was observed that the univariate decision tree had a lower RMSE in the

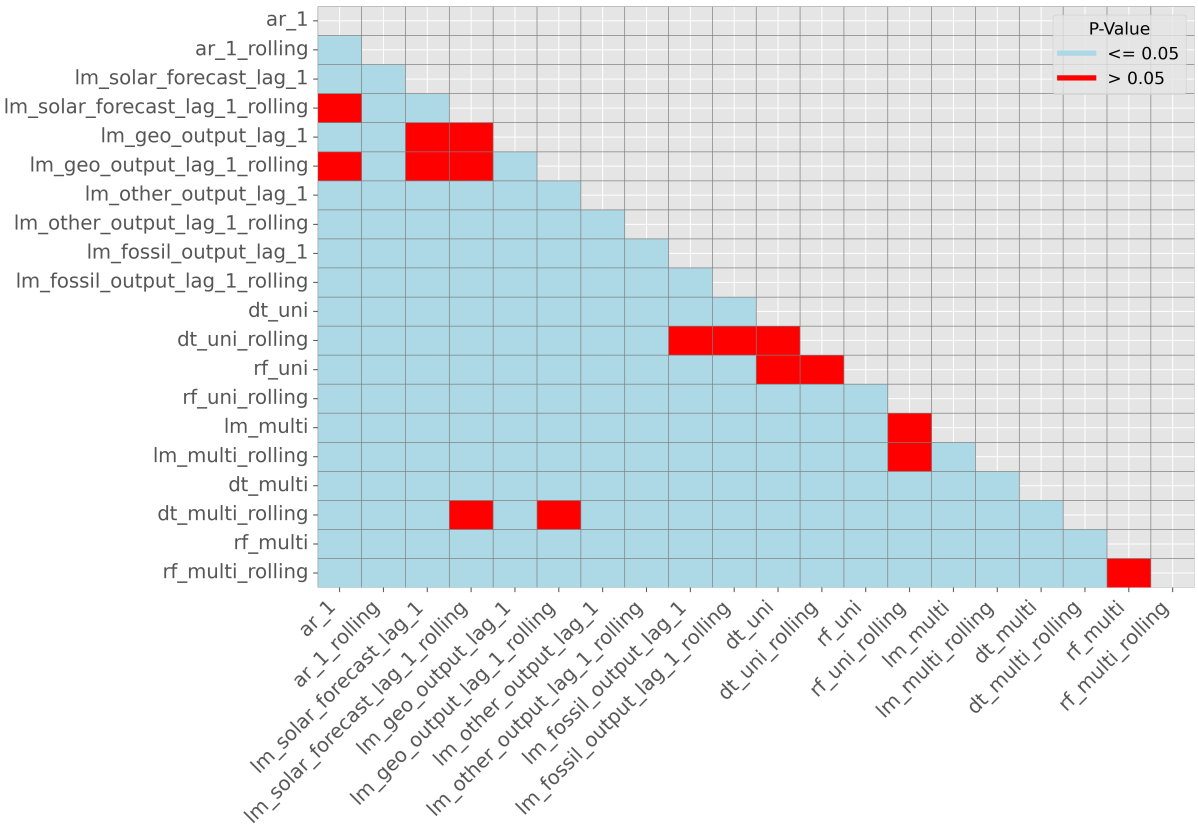


Figure 4.12: P-values of the pairwise comparison of all the models introduced

fixed set compared to the rolling set. However, the DM test indicates that the difference in their predictions is not significant, suggesting that the difference in RMSE was merely due to randomness. Additionally, the pairwise comparison of the univariate decision tree in both rolling and non-rolling cases with the rf_multi also showed non-significance. Hence, there is no difference between the results of the decision tree and the random forest in the univariate case. Apparently, both models appear to learn equally in both scenarios. This is not true in the multivariate case. The same applies to rf_multi and rf_multi_rolling, where both models are concluded to have no statistical difference.

Regarding the linear models, the geo_output have the highest number of similar forecasts compared to other models which haves the number of 4 and are composed of the ar_1, the linear model with the solar forecast in both settings (rolling and non rolling), and with the dt_multi_rolling.

It is evident that the general performance may vary, being better or worse depending on the specific model. To investigate this further, the heatmap is plotted again, this time displaying the respective DM statistics instead of p-values. This will enable us to make preferential inferences about which model performed better in cases where their

differences were significant.

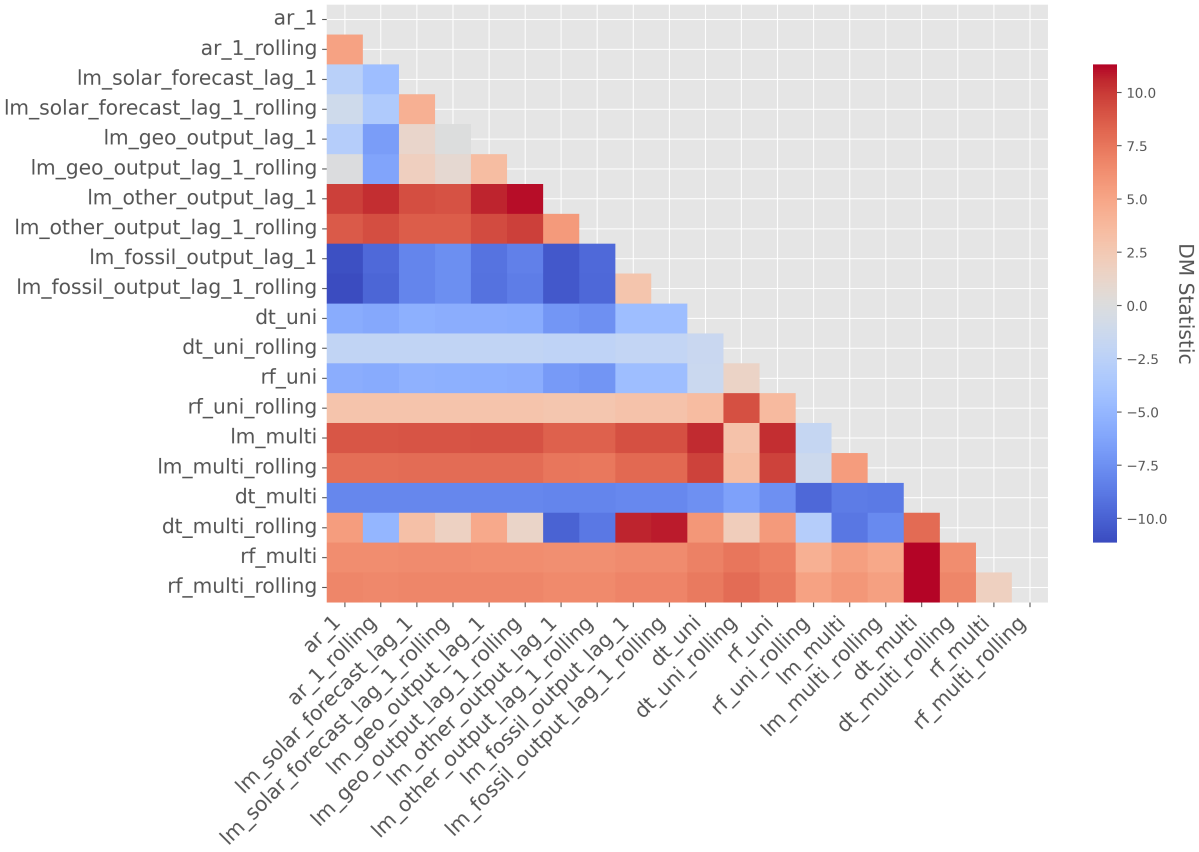


Figure 4.13: Pairwise comparison of the DM-statistics of all the models introduced

In fig. 4.13, whenever the color is reddish at the intersection of two horizental and vertical lines of the heatmap, the model on the y-axis is considered better than that on the x-axis, and vice versa. A pattern emerges horizontally for most models: if the squares in front of a model's is red, it tends to be red for the remaining models compared with, and similarly, if the axis range color is blue, it remains blue across the other models. For instance the rf_multi and rf_multi_rolling are the most reddisch ones showing that they mostly outperform any other model in our set. However, it was already proven that the difference between the both of them is insignificant, so they have the same predictive power eventually. The univariate random forest have mostly blue suggesting that it always performs relatively worse even for simple univariate linear models. The same goes for the univariate and multivariate decision tree. The only notable difference emerges when the latter is used with a rolling window, showing improvement, particularly with the linear model using the fossil output feature.

Switching our focus to the linear models, we observe that the other_output feature

demonstrates stronger performance relative to the other univariate linear models in both rolling and non-rolling test sets. Nevertheless, it is still weaker than the multivariate random forest and linear model. The fossil_output, solar_forecast, geo_output, and ar_1 models on the other hand exhibit more varied performance depending on the models they are compared with.

The previous analysis was exhaustive, attempting to incorporate all pairwise comparison results. When combining all informations extracted, it is possible to summarize them in the following section.

4.6 Shortcomings and Potential Improvements

The initial discovery is that, despite the random forest's general reputation of a powerful predictive tool, it was not consistently better than all other models in this case. In the worst-case scenario, the random forest as good as the decision tree when the model was univariate. This is also evident from the RMSE analysis, where the univariate random forest performed worse than all the rest. However, this conclusion is not statistically robust, highlighting the added value of a statistical test. In such cases, RMSE alone does not always provide an accurate comparison.

In the electricity return predictions, the rolling method typically provided better results, with the worst case being no difference at all. While the RMSE indicated that the rolling method worsened the univariate decision tree model remarkably, the DM test concluded that the difference was not statistically significant. Moreover, a multivariate model will not always outperform a univariate one. This was demonstrated by the DM test, which showed that the univariate decision tree was favored over the multivariate decision tree. Lastly, tree methods do not always outperform simple linear models, as evidenced by the univariate random forest demonstrating the same predictive power as the multivariate linear model.

The research questions we posed generally received negative responses due to some exceptions found during the final analysis. Our univariate tree models used only the one-time lagged return, which likely explains the absence of significant differences between some models. This highlights a potential drawback of using tree methods when there is a lack of a significant number of variables. This may be particularly true for random forests, as the decision tree results did not show much difference between the pairwise

comparisons of the univariate decision tree with other models and the multivariate decision tree with other models. The same applies to the linear setting, which, when considered in a multivariate sense, were sometimes more powerful than the decision tree. In cases where additional variables are more difficult to obtain, these models show a general potential improvement when incorporating a rolling window.

The time series autoregressive model was also not utilized to its full potential due to multiple reasons. The first issue is that the returns had to be processed because their original variability made our model results hard to model. The second is the seasonality of the data and the fact that only an AR(1) was used. The winsorization assisted in reducing the variability of returns but had an effect on the interpretability of them. Nevertheless, this did not prevent the time series from outperforming some other univariate linear models despite it having low variance forecasts. Further deseasonalizing of the data can enhance its performance, which is an area of refinement for future research.

5 Conclusion

Having a model that describes how prices fluctuate would allow to make informed decisions about electricity usage. After implementing multiple modeling techniques, This research combined all the techniques and aimed to identify the benefits and drawbacks of each. It is reminded one more time that this study takes place in the German market and all the analysis is relevant to the electricity data in this region.

The principal questions we began with were whether random forests were better than all other models, whether the rolling window method outperformed the fixed window method in all settings, and whether multivariate models were superior to univariate ones in all cases. Among many other discoveries, the main ones were that the random forests generally had a significant advantage in forecasts but in some cases proved to be equally effective as other models. The same conclusion applies to the rolling window technique where the fixed test set sometimes was equally effective as the dynamic case. For the multivariate models, they were not consistently providing better forecasts, especially with the decision tree. This was believed to be the cause for multiple reasons, one of which is that decision trees may not capture as much variability as random forests do. Nevertheless, tree methods still showed a good advantage given the electricity data that was used. Linear models and other time series models exhibited varying performance depending on the models they were compared with and were not consistently the worst performers.

Throughout all the projects, fixed settings, assumptions and most importantly the preprocessing methods were maintained. By holding these constant, comparable results were achieved that not only aided in selecting the most effective model in forecasting the returns but also demonstrated that using other techniques like the rolling forecast and including more predictors would potentially yield lower errors only under certain model choices. This study can pave the way for further future research that could for instance explore the comparison of these methods using other model comparison tools and the inclusion of more comprehensive data processing techniques to further reduce data seasonality. These studies followed one of many possible approaches, and while they had their limitations, they still proved to be effective in approximating the behavior of the models in the electricity domain in Germany.

Bibliography

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 1974.

Leo Breiman. Random forests. *Machine Learning*, 2001.

Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.

Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. *Classification and Regression Trees*. Routledge, 2017.

Peter J Brockwell and Richard A Davis. *Introduction to Time Series and Forecasting*. Springer, 3rd edition, 2016.

Georges Casella. *Statistical Inference*. Duxbury/Thomson Learning, second edition, 2002.

Todd E Clark and Michael W McCracken. Tests of equal forecast accuracy and encompassing for nested models. *Journal of Econometrics*, 2001.

Francis X. Diebold. Comparing predictive accuracy, twenty years later: A personal perspective on the use and abuse of the diebold-mariano test. `https://www.nber.org/system/files/working_papers/w18391/w18391.pdf`, 2012. NBER Working Paper 18391.

Francis X. Diebold and Roberto S. Mariano. Comparing predictive accuracy. `https://www.jstor.org/stable/1392155`, 2002. Accessed: June 29, 2024.

Yadolah Dodge. The oxford dictionary of statistical terms. *Oxford University Press*, 2003.

J Durbin. The fitting of time-series models. *Review of the International Statistical Institute*, 28, 1960.

Bradley Efron. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 1979.

Walter Enders. *Applied Econometric Time Series*. Wiley, 2 edition, 2004.

Gidon Eshel. The yule walker equations for the ar coefficients.

Arthur S. Goldberger. *Classical Linear Regression*. John Wiley & Sons, New York, 1964.

Guido van Rossum. Python software foundation. `https://www.python.org/`, 1989. Accessed: May 12, 2024.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

Atsushi Inoue, Lu Jin, and Barbara Rossi. Rolling window selection for out-of-sample forecasting with time-varying parameters. `https://www.sciencedirect.com/science/article/abs/pii/S0304407616301713`, 2017. Accessed: July 5, 2024.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*. Springer, 2013.

Project Jupyter. Jupyter notebook. `https://jupyter.org`, 2014. Version 6.5.2.

Gebhard Kirchgässner and Jürgen Wolters. *Introduction to Modern Time Series Analysis*. Springer, 2007.

Whitney K. Newey and Kenneth D. West. Automatic lag selection in covariance matrix estimation. `https://users.ssc.wisc.edu/~bhansen/718/NeweyWest1994.pdf`, 1994. Accessed: June 30, 2024.