UNIVERSITY OF CRETE

DEPARTMENT OF ECONOMICS

M.Sc. PROGRAM IN ECONOMIC THEORY AND POLICY

Modeling and Forecasting Electricity Prices in the Nord Pool Market: An Empirical Analysis and Evaluation

MASTER THESIS

NICODEMOU IRENE

SUPERVISOR: GENIUS-PASCUAL MARGARITA

EVALUATION COMMITTEE: TSAGRIS MICHAIL,
NADER ALHARBI

RETHYMNO, FEBRUARY 2023

Copyright

University of Crete
Department of Economics
Nicodemou Irene

ABSTRACT

The purpose of the following master thesis is to evaluate and compare the predictions about the electricity price in the Nord Pool market through parametric and non-parametric models. We will deal with electricity price analysis, estimation, and forecasting using time series data of electricity price and also sometimes the estimated consumption for one day ahead from January 1 2019 to December 31 2020.

Firstly, we will analyze our time series where the data are daily using the peak hour of each day, which is 10:00 AM. Then we will estimate the in-sample electricity price through various models for the period from January 2019 to December 2019. Afterward, we will forecast the out-of-sample short-term electricity price for one day ahead, specifically making predictions for the peak hour for the whole of 2020 using rolling forecasts where the rolling window is one year. Finally, we will use the actual data of 2020 to compare the models' predictive ability using the forecasting errors.

Our models are divided into parametric models: Simple linear model (SLR), Autoregressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), and Autoregressive/Generalized Autoregressive Conditional Heteroskedasticity (ARCH/GARCH) and a non-parametric model: Singular Spectrum Analysis (SSA).

Keywords: Time series, Nord Pool Electricity Market, short-term forecasting, forecasting methods, rolling forecasts, SSA.

ΠΕΡΙΛΗΨΗ

Σκοπός της παρακάτω μεταπτυχιακής διπλωματικής εργασίας είναι η αξιολόγηση και σύγκριση των προβλέψεων της τιμής της ηλεκτρικής ενέργειας στην αγορά της Nord Pool μέσω παραμετρικών και μη παραμετρικών μοντέλων. Θα ασχοληθούμε με την ανάλυση της τιμής της ηλεκτρικής ενέργειας, όπως επίσης την εκτίμηση και την πρόβλεψη της, χρησιμοποιώντας τα δεδομένα των χρονοσειρών μας για την τιμή της ηλεκτρικής ενέργειας και σε κάποιες και την εκτιμώμενη κατανάλωση για μια μέρα μπροστά απο τον Ιανουάριο του 2019 μέχρι τον Δεκέμβριο του 2020.

Αρχικά θα αναλύσουμε τις χρονοσειρές μας όπου αφορούν ημερήσια δεδομένα για την ώρα αιχμής κάθε ημέρας και στα οποία κάθε κύκλος αποτελείται απο ενα έτος. Στη συνέχεια θα κάνουμε εκτίμηση εντός δείγματος για την τιμή της ηλεκτρικής ενέργειας μέσω διαφόρων υποδειγμάτων για την περίοδο απο τον Ιανουάριο του 2019 μέχρι τον Δεκέμβριο του 2019. Έπειτα μέσω των υποδειγμάτων αυτών θα προβλέψουμε την εκτός δείγματος βραχυπρόθεσμη τιμή της ηλεκτρικής ενέργειας για το 2020 χρησιμοποιώντας κυλιόμενη πρόβλεψη ενός έτους. Τέλος, θα χρησιμοποιήσουμε τα πραγματικά δεδομένα του 2020 για να πάρουμε τα σφάλματα πρόβλεψης και να αξιολογήσουμε την προβλεπτική ικανότητα κάθε μοντέλου, έτσι ώστε να καταλήξουμε στο καλύτερο μοντέλο.

Τα υποδείγματα μας χωρίζονται σε παραμετρικά: Simple linear model (SLR), Autoregressive Integrated Moving Average (ARIMA), Autoregressive/Generalized Autoregressive Conditional Heteroskedasticity (ARCH/GARCH) και σε μη παραμετρικά: Singular Spectrum Analysis (SSA).

Λέξεις κλειδιά: χρονοσειρές, ηλεκτρική ενέργεια στην αγορά Nord Pool, βραχυ-πρόθεσμη πρόβλεψη, μεθόδοι πρόβλεψης, κυλιόμενη πρόβλεψη, ανάλυση του ιδιάζον φάσματος.

TABLE OF CONTENTS

LIST OF ACRONYMS

ACF	Autocorrelation Function
ADF	Augmented Dickey Fuller
AIC	Akaike Information Criteria
AR	Autoregressive
ARCH	Autoregressive Conditional Heteroskedasticity
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
BIC	Bayesian Information Criteria
IID	Independent And Identically Distributed
GARCH	Generalized Autoregressive Conditionally Heteroscedastic
KPSS	Kwiatkowski Phillips Schmidt Shin
LRF	Linear Recurrent Formula
LM	Lagrange Multiplier
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
ME	Mean Error
MPE	Mean Percentage Error
MSE	Mean Squared Error
NP	Nord Pool
R^2	R Squared
RMSE	Root Mean Squared Error
RSE	Residual Standard Error
SARIMA	Seasonal Autoregressive Integrated Moving Average
SLR	Simple Linear Regression
SSA	Singular Spectrum Analysis
SVD	Singular Value Decomposition

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Object and purpose

The purpose of the following master thesis is to study the behavior of electricity prices in the Nord Pool market for the Nordic countries and to predict the electricity price. We have retrieved data for electricity prices and forecasted consumption of electricity in the Nord Pool market. First of all, we will analyze our data, and after that, we will proceed with the electricity price estimation and forecasting of peak hour electricity price (below we will see that the peak time is at 10 A.M.) through various parametric models and a non-parametric model. The data we will use was taken from the electricity market through Nord Pool's day-ahead market where a separate observation was taken every hour and our data covers the period from 01/01/2019 to 31/12/2020.

The research regarding short-term electricity price forecasting is truly extensive. As a whole (Shahidehpour, Yamin, & Li, 2002), (Weron, 2006) and (Zareipour, 2012) all bring up various modeling approaches of the day-ahead forecasting. Recently, Carlo Fezzi and Luca Mosetti (2018) conclude that different models and window sizes should be used for different hours in Nord Pool. Additional to the parametric models that has been used in the above literature, Arash Miranian, Majid Abdollahzade, Hossein Hassani (2013) add a non-parametric model for forecasting the day ahead electricity price by singular spectrum analysis. But the question is which model types are most suitable to forecast the electricity prices on Nord Pool. We have to test if the addition of the non-parametric method can give better results in predictions.

1.2 Thesis structure

Using our data, we will first estimate the electricity price through various models for the period from January 2019 to December 2019 and in the second step, we will forecast the

electricity price for January 2020 to December 2020. Finally, we will use the actual data of the period from January 2020 to December 2020 to compare the predictive ability of the models. Before proceeding with our analysis it should be noted that in cases where there were blank observations in our data, we used the average of their 2 intervening observations to fill in the blanks. Also, we fixed the problem where some data had double observations at the same time.

Short-term electricity price forecasting models are typically estimated through rolling windows for example using the most recent observation. According to Carlo Fezzi and Luca Mosetti (2018), by defining the appropriate rolling window we can greatly improve the forecasting performance. The appropriate rolling window for our data is between six months and one year, so in our analysis, we will set the rolling window equal to one year.

In the analysis of our time series that follows we will divide the models into parametric and non-parametric. Firstly, we will analyze our time series through models where we will use the peak time of each day for the whole year so our data will be data per day. Subsequently, in-sample price forecasts will be computed from estimated models, and at the end, we will do a rolling forecast for the out-of-sample data to predict the price of electricity for the year 2020. At the end of our analysis, we will compare the results of the estimations and predictions of the various models through the forecasting errors.

In Chapter 2 we will analyze the electricity market and the importance of electricity price predictions. Next, in Chapter 3 we will present the theory of the time series analysis and in Chapter 4 we will present the time series analysis for the Nord Pool Market. Subsequently, in Chapter 5 we will apply the models to the Nord Pool data, proceeding to estimations and forecasts. Finally, in the last Chapter, we will compare the results of the in-sample and out-of-sample forecasts.

CHAPTER 2: ELECTRICITY PRICE FORECASTING

2.1 Electricity market

The electricity market concerns purchases made in the electricity trade for the purchase and sale of electricity. The electricity market can be regulated or liberalized (deregulated) and can be organized in a variety of ways. As far as regulation is concerned, in regulated markets electricity prices are determined by a regulatory authority and they are usually based on production costs, demand, and supply of electricity. According to Rafal Weron (2014), in liberalized markets, prices are determined by supply and demand and at the same time are affected by factors related to energy in general such as the supply of renewable energy sources, fuel costs, and weather conditions.

Regarding the organization of the electricity market, the market can be retail, wholesale, regional, or global. In the wholesale market, the purchase and sale of electricity are made by large buyers and utilities, while in the retail market, the sale is made to consumers, i.e. households and businesses. Finally, there is the electricity market which is done at a regional and global level, or globally where the connection of the markets is done through transmission systems and interconnections.

2.2 Nord Pool electricity Market

Nord Pool was founded in 1993 and is the first market worldwide that operates at fully competitive prices. As we mentioned above, the electricity market can be organized in various ways. Nord Pool is a wholesale electricity market that is a key commercial player and serves the Nordic countries: Denmark, Finland, Norway, and Sweden as well as the Baltic States: Estonia, Latvia, and Lithuania. It belongs to the largest electricity markets in Europe and operates as a

spot market where electricity is bought and sold with delivery on the same day or within a few days, but also as a producer market. It also belongs to the global market since it is also connected to other European markets through interconnections and transport systems.

The Nord Pool market is divided into two sectors: the intraday electricity market where electricity is delivered on the day of trading and the day-ahead market where delivery takes place the day after trading. In our analysis, the data we use is for the day-ahead market where it is done on a nice basis and in which the stakeholders (buyers and sellers of electricity) quote supply and demand for each hour of the next day. After the submission by the participants, the auction process follows which is based on the above bids and requests and determines the market clearing price.

The data used in the present thesis was retrieved from the Nord Pool data portal (2022), for system prices and forecasted consumption in the Nordic countries.

2.3 Importance of Electricity price predictions

The electricity market is a market that concerns everyone and the prediction of its price is necessary for electricity producers, the retail market, and all consumers. The price of electricity concerns a non-stationary and non-linear time series, with a variable mean and variation, so the forecasting process becomes difficult. The reasons for needing electricity price forecasting are varied and depend on whom the market is concerned with at any given time. Some possible reasons may be to initially manage and avoid risk for traders so that investors are more informed about future trades. Also, this will help the investors of other markets that are related and sensitive to the price of electricity to maximize the profits of their investments. One more reason is that governments and regulators can use forecasts to inform policy decisions related to energy markets and infrastructure planning. For example, if a large increase in the price of electricity is expected, the necessary decisions to support consumers must be taken. Such a surge has recently

occurred globally due to Russia's war against Ukraine. Finally, with regard mainly to large energy consumers, through the forecasts, they can optimize energy consumption and reduce costs by using, for example, more energy-efficient devices.

2.4 Time frame of model predictions

By the meaning of the time frame of a forecasting model, we refer to the period used to make predictions, where the time frame differs according to our data, the forecasting model we use, and the purpose of the forecasts. There are data for which we want short-term predictions, for example, the prediction of the next week's stock price, but also long-term predictions, for example, the demand for a product in the next year or decade.

The time frame of a forecasting model can also vary depending on the frequency of the data being analyzed. For example, a model designed to make daily forecasts may have a different time frame than a model designed to make monthly or yearly forecasts. In the Nord Pool market that concerns us, we will make one-step ahead forecasts of the price of electricity for the next day until the last day of 2020, and these forecasts are considered short-term since we are forecasting the next day using rolling window of one year.

CHAPTER 3: TIME SERIES ANALYSIS

3.1 Definition of Time Series

By the term of time series, we mean the set of some data such as the temperature or the price of a stock that is collected at regular time intervals such as every hour, day, month, or year. They are not only applicable to statistics and economics but can also contribute to other areas such as quality control or weather forecasting. Time series due to their form create a time classification since they are collected in a specific period and allow us to analyze the time series and their characteristics. Time series analysis is used to extract important information, make estimates, and also to make future predictions. As far as data analysis is concerned, time series models can be analyzed such as the moving average model, the moving average autoregressive model, or the simple linear regression model. In addition to the above examples, there are other techniques such as smoothing, decomposition, and transformations.

3.2 Time Series Models

As we mentioned above, there are different types of time series models that we can use to analyze our data and make predictions. The most common models for time series analysis are the moving average model (MA) which uses the moving average of past data for forecasting, and the autoregressive integrated moving average (ARIMA) which is a generalization of an autoregressive moving average (ARMA) model and is a linear combination of the autoregressive (AR), variance and MA components and finally the exponential smoothing model which it is similar to the MA but places more importance on recent data and less on older data. The choice of the appropriate model depends each time on the characteristics of our data and the purpose of the forecasts.

Time series models assume that there are correlations in the time series. When examining a real process or a process with observed noise, it is important to use methods based on the theory of stochastic processes. These methods allow for the analysis and modeling of random behavior, which is often present in real-world systems. Stochastic processes can be used to model the evolution of a system over time and to analyze the impact of random events on the system's behavior. In addition, these methods can be used to estimate the magnitude of the noise or other random components present in the system. The stochastic process is a phenomenon that evolves over time, according to the laws of probability.

3.3 Regression

Regression is a statistical modeling method that commonly has applications in statistics and econometrics, and is based on the study of the relationship between a dependent variable and one or more independent variables. The regression technique is used to assign data to a true predictor variable. A necessary condition to proceed with the modeling of our data from a regression model is that our data match some known type of function (for example a linear function) and thus we choose the appropriate type of regression. There are various types of regression techniques such as linear, polynomial, logistic, and non-linear regression. The regression technique we will use to evaluate the results for our data is linear regression, which is the most basic form. The objective of linear regression is to find the best-fitting linear equation that describes the relationship between the dependent and independent variables so that predictions can be made about the value of the dependent variable, in our case the price of electricity based on the values of the independent variables i.e. the predicted consumption. Linear regression assumes that the relationship between the variables is linear and that the errors are normally distributed.

3.4 Stationarity

In the analysis of time series, stationarity is a basic condition and a basic problem that we are often asked to face is the existence of non-stationarity since it does not allow us to proceed with predictions. Non-stationarity is due to reasons such as the existence of a trend, periodicity, seasonality, or the existence of extreme values. A stochastic process is called weakly stationary if the mean and variance do not change over time and the covariance of its values in two time periods depends only on the time lags and not on the time point at which it is estimated. In a few words, a stationary process is when the effect of an unexpected change diminishes/has less effect over time, while a non-stationary process is one in which an unexpected change at time t is followed by equivalent changes for subsequent years and so volatility does not decrease over time. That is why characteristics such as trend and seasonality lead to non-stationarity and affect the value of the time series at different points in time.

The conditions of the weak stationarity represent in the following Equations (1) - (3):

$$E(Y_t) = \mu, \forall t, \quad \textbf{(1)}$$

$$Var(Y_t) = \sigma_Y^2 = \gamma_0, \forall t, \quad \textbf{(2)}$$

$$Cov(Y_t, Y_{t-k}) = E[(Y_t - \mu)(Y_{t-k} - \mu)] = \gamma_k, \forall t, \quad \textbf{(3)}$$

where E represents the expectation, μ represents the mean of a set of numbers, Var or σ^2 represents the variance, Cov represents the Covariance, γ_k represents the covariance at lag k, Y_t represents the value of the time series at time t and k represents the lag. Non-stationarity means time dependence in (1) - (3) and stationarity means no time dependence in (1) - (3).

Stationarity can first be observed by studying the graphical display of the series or by constructing and studying the autocorrelation function and its corresponding correlogram. If we want to show whether or not there is stationarity in our data through some test, we can apply the Augmented Dickey-Fuller (ADF) unit root test or the Kwiatkowski-Phillip-Schmidt-Shin (KPSS) stationarity test, where in both tests by the term null hypothesis we mean H_0. If it turns out that

we have non-stationarity, then we can use a transformation to achieve stationarity. In Table 3.1 we represent both tests, wherein each test we look at the critical value next to our control which shows us if there is a statistical significance of the results for level 0.05.

	ADF	**KPSS**
H_0	Existence of unit root	Stationarity
H_1	No unit root	Non-stationarity

Table 3.1: Stationarity tests.

3.4.1 Augmented Dickey-Fuller unit root test

According to James Hamilton (1994), the ADF test is a statistical test used to determine whether a time series has a unit root or not. In econometrics and time series analysis, stationarity is an important property because many statistical methods assume that the data is stationary. The null hypothesis in our test is non-stationarity or that there is a time dependence, i.e. we have a unit root. Accordingly, the alternative hypothesis is stationarity or that there is no dependence on time and the time series cannot be reproduced from a unit root.

3.4.2 Kwiatkowski-Phillips-Schmidt-Shin stationarity test

The KPSS test is a statistical test used to determine whether a time series is stationary or not. The null hypothesis in our test is stationarity and the alternative hypothesis is non-stationarity. The KPSS test statistic is calculated based on the autocorrelation of the residuals of a regression of the time series on its lags and a constant. The test statistic is then compared to critical values from a chi-squared distribution to determine whether to reject or fail to reject the null hypothesis.

3.5 Box-Jenkins approach

The goal of time series analysis in the present thesis is to study how the electricity price behaves over time. Firstly, we want to study the structure of our data for the electricity price but also for the predicted consumption which we will use in some models to identify if there exist correlations and similarities between our time series. Then we will estimate the price of electricity through the models and finally, we will proceed to predict the future values of the price of electricity. To achieve as accurate estimations and predictions as possible, we must first study the basic characteristics of our data. In the end, with the help of the appropriate comparison measures, we will evaluate the models we used.

Essentially, the above methodology can also be referred to as a Box-Jenkins approach where we have 4 steps:

1. Model Identification.
2. Model estimation.
3. Diagnostic test.
4. Time series forecasting.

Initially, the identification of the model is carried out by examining the plot of our original time series and by examining the autocorrelation and partial autocorrelation plots that show us the characteristics of our time series. If our time series is non-stationary we need to use the difference transformation to achieve stationarity. This step will lead us to select some models for evaluation. Then we will estimate the parameters defining the model identified in the first step. The third step is to examine whether the residuals of the estimated model are characterized as white noise. Finally, if the answer to the previous step is positive and after the determination of the best fitted model, we will use it to predict future values.

3.6 Quality characteristics of time series

The main qualitative characteristics likely to be present in a time series are the trend, seasonality, cyclical variations, and the existence of extreme values. Below we will present in detail the importance of each feature and whether we can deal with them so that they do not affect our prediction process.

3.6.1 Trend

The trend of a time series appears in the form of changes in the mean or variance, where the mean may increase or decrease following some linear or non-linear pattern. The time series may not be stationary in mean, variance, or both. The voltage can be represented as a simple linear function of time or perhaps a polynomial function of time or exponential. It is distinguished into deterministic when it can be described by a function that is known or can be estimated, and stochastic that cannot be described by a known parametric function of time, i.e. it shows slow changes over time but not in a deterministic way.

3.6.3 Seasonality

Seasonality is observed when there is approximately a recurring pattern in a specific time horizon, eg six months or a year. Seasonality also (like the trend) leads to the non-stationarity of time series. It is a periodic fluctuation that is constant and less than or equal to one year in length. Since this variation is seasonal, that is, it occurs systematically, we can easily measure it and isolate it so that it does not affect our data and we can lead to a stationary series.

3.6.4 Cyclicity

Cyclicity is observed in fluctuating time series with bullish and bearish phases that repeat sequentially around the trend line, and cyclical behavior is defined by two lower inflection points

and one upper inflection point (top). These cyclical changes do not repeat themselves at regular intervals, so there is no fixed period length. Cyclical movement cannot be easily treated, unlike seasonality, since it follows no regular pattern but moves unpredictably.

3.6.5 Outliers

Another characteristic of non-stationarity is isolated observations or extreme values, which appear in the time series as abrupt changes in its pattern of behavior. These points are unpredictable and their effect on the time series has a short duration. They may represent an unusual observation due to some unforeseen event (e.g. a strike) or simply an error in the observation recording system.

3.7 Quantitative characteristics of time series

Quantitative features are the statistical indicators that can be calculated from the time series data, and the most basic indicators that we will analyze below are the mean value, standard deviation, and minimum/maximum values.

3.7.1 Mean value

The mean represents the average value of a data set and it can be used to understand the distribution of data, however, it can be affected by outliers and may not always accurately represent the standard value in the data set. Equation (4) represents the equation for the mean (also known as the average) of a set of numbers as follows:

$$\overline{Y} = 1/n \sum_{i=1}^{n} Y_i \,, \tag{4}$$

where $\overline{Y}$ represents the mean, Y_i represents each value in the set and n represents the number of values in the set.

3.7.2 Standard deviation

The standard deviation (SD) is a measurement of the dispersion of a data set and is calculated as the square root of the variance, which is the average of the squared differences between the values in the data set and the data set mean. Like the mean, the SD can be used to understand the distribution of data. If the SD is large then the values in our data set are scattered, while the smaller the SD the closer to the mean our values are. Equation (5) represents the SD of a set of numbers as follows:

$$SD = \sqrt{\frac{\sum_{i=1}^{n} (Y_i - \overline{Y})^2}{n-1}} \,. \tag{5}$$

3.7.3 Minimum and maximum value

The minimum and maximum values refer to the lowest and highest value of a data set and can be used to describe the range and spread of values.

3.8 Autocorrelation study

To study correlations in stationary time series, autocorrelation is used, i.e. the normalization of the covariance with the dispersion. Autocorrelation is the correlation between the time series and itself at different lags. The definition of autocorrelation makes sense when the time series is stationary while when the series is not stationary the autocorrelation cannot be defined as a function of the lag but is defined for each time instant.

Autocorrelation can be calculated for 2 different reasons: First, it can be calculated on our real data before running any model so that we understand the characteristics of our data. This will help us identify if our data is characterized by stationarity or if there is any trend or pattern in our initial data that must be addressed subsequently in the modeling process that we will apply. The second reason is to use the autocorrelation function (ACF) after running our model so that we can

assess the quality of our model and examine whether autocorrelation is present in the residuals. That is, this way will help us to determine if our model has adequately captured the patterns or trends that we will find from applying ACF to our data before running any model and thus we will examine for any remaining autocorrelation in the residuals of the model.

3.8.1 Autocorrelation Function

The ACF is a statistical technique that we can use to determine how correlated the values of a time series are with each other. Equation (6) represents the ACF of a time series as follows:

$$\rho_\kappa = \frac{\gamma_\kappa}{\gamma_0} = \frac{\sum (Y_t - \overline{Y})(Y_{t+k} - \overline{Y})}{\sum (Y_t - \overline{Y})^2}\ , \qquad \textbf{(6)}$$

where ρ_κ represents the autocorrelation between the values at time t and time t+k and γ_0 represents the variance.

The ACF plot is a key tool for pattern recognition and plots the correlation coefficient against the lag k, which is measured in terms of several periods. A lag corresponds to a certain point in time after which we observe the first value in the time series. The autocorrelation coefficient can range from -1 (perfectly negative relationship) to +1 (perfectly positive relationship). A value of 0 means that there is no relationship between the variables.

The horizontal lines on an ACF plot are the error bars where anything inside those lines is not statistically significant. It means that for correlation values outside this range, it is very likely that there is a correlation rather than a statistical fluke. Practically, for a time series to be stationary, the autocorrelation diagram should peak immediately after the first k lags, while in a time series with a strong trend, the autocorrelation diagram will decrease slowly as the value of k increases. In the case of the existence of a strong seasonal component, we have strong autocorrelations at specific lags depending on the nature of the data. When our bar graphs are

very high and we have no stationarity, we can do a transformation (take the differences) to solve the non-stationarity problem.

In addition to the autocorrelation function, there is also the partial autocorrelation function which captures a "direct" correlation between the time series and a lagged version of itself, when the effect of other time lags remains constant. In Figure 3.1 we introduce an example of the ACF and PACF plots, where the dotted lines on the ACF and PACF plot are the error bars where anything inside those lines is not statistically significant. On the PACF plot, there is no statistically significant autocorrelation, and on the ACF plot, there is a statistically significant autocorrelation at lag 4.

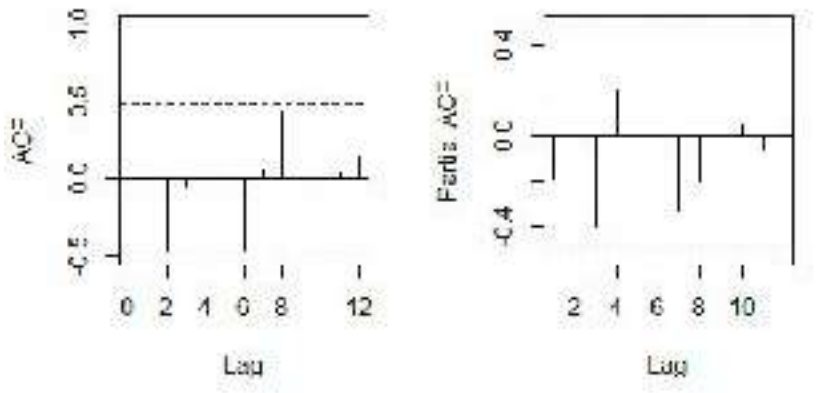

Figure 3.1: Example of ACF and PACF plots.

3.8.2 Independence test

We can test for autocorrelation through the Ljung-Box test which is a hypothesis test that tests whether a time series contains autocorrelation. The null hypothesis is that the residuals are independently distributed and the alternative hypothesis is that the residuals are not independently distributed and are autocorrelated. There is also the Box-Pierce independence test which is similar to the Ljung-Box and has the same results with the only difference being in the way they calculate the statistics, and is commonly used in time series analysis to diagnose the need for more complex modeling techniques, such as ARIMA or seasonal ARIMA (SARIMA)

models. In both tests, we have a statistical significance level of 0.05. Table 3.2 presents both independence tests. These tests can be applied to our original data before running any model to see if there is autocorrelation in the data, and also to our data after we run a model to see if there is autocorrelation in the residuals. Equation (7) is the Ljung-Box test statistic:

$$Q_{LB} = n(n+2)\sum_{i=1}^{k}\frac{\rho_i^2}{n-k}\,, \tag{7}$$

where Q_{LB} represents the Ljung-Box test statistic and ρ_i represents the autocorrelation at lag i. Equation (8) is the Box-Pierce test statistic:

$$Q_{BP} = n\sum_{i=1}^{K}\rho_i^2\,, \tag{8}$$

where Q_{BP} represents the Box-Pierce test statistic.

	Ljung-Box result	**Box-Pierce result**
H_0	No autocorrelation	No autocorrelation
H_1	Evidence of autocorrelation	Evidence of autocorrelation

Table 3.2: Independence tests.

3.9 White noise

A stationary time series with 0 autocorrelation for every lag other than 0 is called white noise, i.e. the sample ACF tends to 0. If we consider consecutive elements of the time series as random variables, then they are independent and identically distributed (iid) random variables and there are no correlations between them when for t > 1 the random variables have the same distribution and are independent between theirs. White noise is characterized by second-order stationarity, but if in addition, regularity can be ensured, then we have a strictly stationary

process. A white noise process can be represented with the following Equations:

$$Y_t = \mu + \varepsilon_t, \forall t \,, \tag{9}$$

$$E(\varepsilon_t) = 0, \forall t \,, \tag{10}$$

$$Var(\varepsilon_t) = E(\varepsilon_t^2) = \sigma^2, \forall t \,, \tag{11}$$

$$Cov(\varepsilon_t, \varepsilon_s) = E(\varepsilon_t, \varepsilon_s) = 0, \forall t \neq s \,, \tag{12}$$

where x_t represents the value of the white noise process at time t and ε_t represents the random error term at time t where the error terms are independent and normally distributed.

In the context of time series analysis, we want white noise in the error term after we apply our model rather than in our actual data. If the errors are white noise it means that the errors are randomly distributed and uncorrelated so the model correctly captures the underlying patterns in the data and any remaining variation is due to random noise. Conversely, if the data itself is white noise it means that there is no pattern or structure in the data and it is difficult for any model to make accurate predictions since there is no important information about our data to help us choose the appropriate model. In addition to the Ljung-Box independence test, there is also the Durbin-Watson test, which is a first-order autocorrelation test in the residuals, so it is only applied after some model and not to our original data before running a model. In Figure 3.2 we represent a plot of ACF, in which we have white noise because all the lags are inside the dotted lines, so there is no statistically significant autocorrelation.

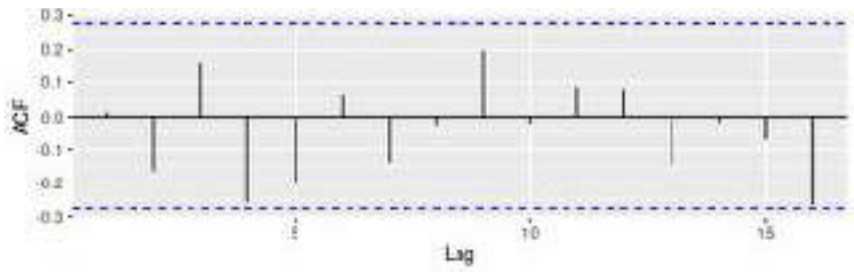

Figure 3.2: Example of white noise.

3.10 Transformations

As we mentioned above, stationarity can be achieved with the help of some transformations. Basic transformations are applied by taking the first differences or by taking the logarithm. We have the following transformations:

3.10.1 Box-Cox Transformation

The Box-Cox transformation is used to stabilize the variance of the data, improve homoscedasticity, or make the data more suitable for parametric statistical tests that assume normally distributed data. The Box-Cox transformation satisfies the criterion of power equivalence, meaning that it can transform any non-constant, positive data set to make it approximately linear. This makes the transformed data easier to model and analyze using linear regression techniques. Equation (13) presents the cases of the Box-Cox transformation

$$Y_i^{(\lambda)} = \ln(Y_i)\ if\ \lambda = 0,\ or\ Y_i^{(\lambda)} = \frac{Y_i^{\lambda} - 1}{\lambda}\ otherwise, \quad \textbf{(13)}$$

where λ represents the transformation parameter. Depending on λ we have the following results:

- If λ = 0 then the Box-Cox transform is simply the logarithmic transform.
- If λ = 1 then our data do not need transformation.
- If λ = -1 then we have the inverse transformation.
- If λ = 1/2 then we get the square root transformation.
- If λ = 1/3 then we get the cube root.

3.10.2 Logarithmic transformation

The second transformation we will perform is by taking the logarithms which helps to stabilize the variance of the time series. The logarithmic transformation is essentially a special

case of the Box-Cox transformation seen earlier where λ=0. The logarithmic transformation is given by:

$$y = log(x), \quad \textbf{(14)}$$

where x represents the number being logarithmized and y represents the logarithmic result.

3.10.3 Method of first differences

Differencing helps to stabilize the mean of the time series by removing variations in the level of the time series. Sometimes the 2nd difference is needed to make the series stationary, although practically it is rarely necessary to proceed to a 2nd difference. Differencing helps to stabilize the mean of the time series by removing variations in the level of the time series. Thus it reduces or eliminates trend and seasonality. The first difference of a time series can be represented mathematically as follows:

$$\Delta Y_t = Y_t - Y_{t-1}\ , \quad \textbf{(15)}$$

where ΔY_t represents the first difference of the time series at time t.

3.10.4 Method of seasonal differences

Seasonal differentiation is the difference between our observation and the previous observation of the same period, i.e. it takes into account the seasonality of our time series. The seasonal difference of a time series can be represented mathematically as follows:

$$\Delta sY_t = Y_t - Y_{t-s}\ , \quad \textbf{(16)}$$

where ΔsY_t represents the seasonal difference of the time series at time t, and s represents the number of time units in a season (e.g., s=7 for weekly data and s=12 for monthly data). In time series analysis, seasonal differentiation is used as a preprocessing step before fitting models like SARIMA.

3.11 Parametric prediction models

3.11.1 Simple Linear Regression model

The first model that we will use is the Simple Linear Regression model (SLR). It is a linear regression model with only one explanatory variable ie it involves one independent variable and one dependent variable and finds a linear function (a non-vertical straight line) that as accurately as possible predicts the values of the dependent variable as a function of the independent variable. The SLR model is represented by Equation (17):

$$Y_i = a + \beta X_i + \varepsilon_t \,, \tag{17}$$

where Y_i represents the dependent variable, X_i represents the independent variable, α represents the intercept, β represents the slope coefficients and ε represents the error term that captures the difference between the observed value of Y and its mean. Estimation of the model by least-squares gives us estimates of the coefficient β of the regression and the p-value of the t-test is used to assess its significance.

	T-statistic results
H_0	The coefficient are equal to zero/there is no relationship between x and y
H_1	The coefficient are not equal to zero/there is relationship between x and y

Table 3.3: T-statistic.

If even one predictor variable is significantly related to the output, we need to proceed to the accuracy of the model, in which diagnosis is based on checking how well the model fits the data. Thus, to see overall the quality of the fit of the linear regression, we will use the evaluation measures: residual standard error (RSE), R squared (R^2), and F-statistic. This process is also known as goodness-of-fit.

3.11.2 Autoregressive Integrated Moving Average Model (p,d,q)

The Autoregressive Integrated Moving Average Model (ARIMA) model is a time series analysis and forecasting technique used in economics, statistics, and signal processing to characterize relationships between variables. It can predict future values based on past values and is a linear combination of its past values, current errors (also known as innovation term), and past errors. The ARIMA model is widely used in time series analysis and is particularly useful for modeling data with trend and seasonality. According to Hyndman, R. J., & Athanasopoulos, G. (2014), it has three parameters that respectively define the order of the autoregressive (AR) part with the number p i.e. the number of AR terms representing the number of previous values used to predict the current value, with d being the number of differences representing the number of times the data has been differenced to be stationary (i.e. to remove trend and seasonality) and finally q is the number of moving average (MA) terms which represents the number of past errors used to predict the current error.

The ARIMA model is the Autoregressive Moving Average Model (ARMA) with the order of differencing (d) which is an important parameter of ARIMA and determines the success of the model. If the ARMA model is not stationary, then we use the first-order difference (d=1), or larger order when seasonality doesn't exist yet. The ARMA is represented by the following Equation:

$$Y_t = \beta + (\Phi_1 Y_{t-1} + \ldots + \Phi_p Y_{t-p}) + (\omega_1 \varepsilon_{t-1} + \ldots + \omega_q \varepsilon_{t-q} + \varepsilon_t) \,, \quad \textbf{(18)}$$

where β is the mean of the time series, $\Phi_1, \Phi_2, \ldots, \Phi_p$ are parameters that control the weight given to past values, $Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}, \omega_1, \omega_2, \ldots, \omega_q$ are parameters that control the weight given to past errors $\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q}$, and ε_t is the error term at time t that captures any unpredictable part of the time series.

For better understanding before our analysis, we will define the two parts AR(p) and MA(q) separately. Before that, we need to know that both AR and MA models require stationarity. Firstly, the autoregressive model expresses the dependence of a variable on a previous period, where the signal depends only on its previous values and is represented by the following Equation:

$$Y_t = \mu + \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} \ldots + \Phi_p Y_{t-p} + \varepsilon_t \,, \tag{19}$$

where μ is the mean of the time series, $\Phi_1, \Phi_2, \ldots, \Phi_p$ are parameters that control the weight given to past values and $Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}$. Also, the moving average model calculates its forecast value by taking a weighted average of past errors. It can capture trends and patterns in time series data. The MA is represented by the following Equation:

$$Y_t = \mu + (\omega_1 \varepsilon_{t-1} + \omega_2 \varepsilon_{t-2} + \ldots + \omega_q \varepsilon_{t-q} + \varepsilon_t) \,, \tag{20}$$

where Y_t is the value of the time series at time t, μ is the mean of the time series, and $\omega_1, \omega_2, \ldots, \omega_q$ are parameters that control the weight given to past residuals $\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q}$. The errors have a mean of zero.

	AR(p)	MA(q)	ARMA(p,q)
ACF	Tails off (trends to zero gradually)	Cuts off after lag p (disappear or zero)	Tails off after lags (p-q)
PACF	Cuts off after lag p	Tails off (decays for slowly than AR)	Tails off after lags (p-q)

Table 3.4: Behavior of ACF and PACF.

As we said before, ARIMA is a combination of the AR(p), MA(q), and the differentiation number. To find the best order of the ARIMA, we will compare the Bayesian Information Criterion of each ARIMA. The lower the value of these criteria, the better the model. Furthermore, except for ARIMA, we will try an extension: Seasonal ARIMA (SARIMA), which includes additional seasonal terms in the ARIMA. SARIMA adds three new parameters to specify

the AR, differencing, and MA for the seasonal component of the series, as well as an additional parameter for the period of seasonality.

3.11.3 ARCH/GARCH model

The ARCH model is defined as the Autoregressive Conditional Heteroscedasticity model and the GARCH as the Generalized Autoregressive Conditional Heteroscedasticity model. Before proceeding to choose an ARCH or GARCH model, we need to perform the heteroskedasticity test on the residuals to see if the ARCH or GARCH models fit our data. According to Tim Bollerslev, Robert F. Engle and Daniel B. Nelson (1994), ARCH/GARCH models describe the conditional variance of the error term as a function of the actual magnitude of the error term of previous periods. Both of them, are statistical models used to capture the time-varying volatility of a time series. The ARCH(q) model is appropriate when the error variance follows an AR(q) model, while if the error variance follows an ARMA model then we want the GARCH(p, q) model where p is the order of the GARCH terms and q is the order of ARCH the terms. Firstly, to model a time series using an ARCH process, we denote the ε_t which are split into a stochastic piece z_t (the random variable z_t is a strong white noise process) and a time-dependent standard deviation σ_t so that:

$$\varepsilon_t = \sigma_t z_t \ . \tag{21}$$

The ARCH Equation is represented as follows :

$$\sigma_t^2 = a_0 + a_1\varepsilon_{t-1}^2 + \ldots + a_q\varepsilon_{t-q}^2 = a_0 + \sum_{i=1}^{q} a_i\varepsilon_{t-i}^2 \ , \tag{22}$$

where σ_t^2 represents the conditional variance at time t, a_0 is a constant larger than 0, and $\alpha_1, \alpha_2, \ldots, \alpha_q$ are parameters which are larger or equal to 0 and control the weight given to past residuals. The GARCH model is represented by the Equations (23)-(25):

$$\sigma_t^2 = \omega + a_1\varepsilon_{t-1}^2 + \ldots + a_q\varepsilon_{t-q}^2 + \beta_1\sigma_{t-1}^2 + \ldots + \beta_p\sigma_{t-p}^2 = \omega + \sum_{i=1}^{q} a_i\varepsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i\sigma_{t-i}^2 , \quad \textbf{(23)}$$

$$Y_t = X_t^{'}b + \varepsilon_t , \quad \textbf{(24)}$$

$$\varepsilon_t | \psi_{t-1} \sim N(0,\sigma_t^2) , \quad \textbf{(25)}$$

where σ_t^2 is the conditional variance, ω is a constant, and $\beta_1, \beta_2, \ldots, \beta_q$ are parameters that control the weight given to past residuals and past variances.

3.11.3.1 Engle ARCH LM test

Engle's ARCH Lagrange Multiplier test, tests whether or not there is heteroscedasticity in the residuals and follows the Chi-square distribution with the number of degrees of freedom which is equal to the number of ARCH terms included in the model. Table 3.5 shows the hypothesis of the ARCH LM test, according to its results if there is no ARCH form in the residuals then the autocorrelations and partial autocorrelations of the squared residuals should be zero at all time lags and the Q statistics should be non-statistically significant at the 0.05 significance level.

	LM statistic Results
H_0	The residuals are not characterized by the ARCH form of heteroscedasticity up to the k lag, i.e. we have homoscedasticity.
H_1	The residuals are characterized by the ARCH form of heteroscedasticity up to the k lag.

TABLE 3.5: Engle's ARCH LM test.

To proceed with the test, we enter the number of time lags k that we want. Then we run an auxiliary regression with the squared residuals as a dependent variable and the time lags k of the squared residuals as explanatory variables. In this test, the number of observations n on the coefficient of determination R-squared is calculated. The formula is as follows: $LM = n * R^2$. If

our testing leads us to apply the ARCH/GARCH models (when the residuals are characterized by the ARCH form of heteroscedasticity), the volatility will depend on the recent past of the time series.

3.11.3.2 Apply ARCH/GARCH models

It is important to note that ARIMA models are designed to linearly model time series data, which can limit their ability to capture complex nonlinear patterns in the data. Additionally, the forecast range for ARIMA models is fixed and cannot be adjusted easily, which can make them less suitable for predicting future changes in the data. Furthermore, it is worth noting that ARIMA models are typically based on historical data and do not reflect recent changes or incorporate new information in real time. That's why ARIMA models give the best linear prediction and play a minimal role in predicting non-linear models. To model volatility, we use ARCH or GARCH models, but before that, we need to make sure if the ARCH or GARCH model is necessary for our time series, so we will do the following 3 steps. In the first step, we will check if our ARIMA residuals chart has any signs of volatility (minimal). Then, in the second step, we will make the square plot of the residuals of the ARIMA model. If it shows signs of volatility then we use the ARCH or GARCH model which will reflect the recent changes and fluctuations of our series. In the third step, the ACF and PACF of the squares of the residuals from the previous step will help us see if the residuals, i.e. the term containing the noise, are not independent and can be predicted. Should be noted that strict white noise cannot be predicted either linearly or non-linearly, while simple white noise can be predictable non-linearly, so if we have strict white noise we will not be able to make a prediction.

3.12 Non-parametric prediction models

3.12.1 Singular Spectrum Analysis model

Singular Spectrum Analysis (SSA) is a non-parametric method for time series analysis and forecasting which is used in fields such as economics and natural sciences. SSA does not require specific conditions on the structure of the time series such as stationarity, linearity, or a training stage to make a prediction. It can identify and remove trends, cyclicality, seasonality patterns, and noise components from time series, achieving good forecasts.

SSA is a way of decomposing a time series into a much smaller series of eigenvectors and their corresponding eigenvalues, which can be used to extract information about trends, seasonal patterns, and noise in the data. The main idea of SSA is to break down time series into different components and reconstruct our data without noise for further analysis. It depends on the choice of the window length: L and the number of required singular values/eigenvalues: r for the reconstruction, where we have to separate the periodic components and signal from the noise components. The main idea of separability, which characterizes how well signal and noise elements can be separated, is to choose the appropriate number of eigenvalues where we will proceed by choosing the periodic components and signal and removing the noise components.

According to Nader Alharbi and Hossein Hassani (2016), the length of the window should be large enough but not larger than n/2 (where n is the number of observations). As regards the eigenvalues r, there are several methods for selecting the required number of r, such as the analysis of the singular value plot, the eigenvector analysis plot, the periodogram, the convexity plot, and the correlation matrix: W-correlation.

The SSA process can be summarized in the following four steps:

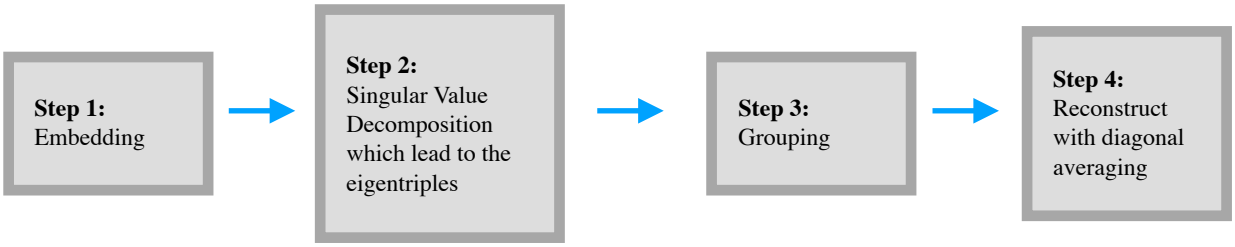

Figure 3.3: Steps for the SSA process.

We will mention the steps in detail, according to Miranian, Abdollahzade, and Hassani (2013). In the first step, we have the embedding where the original time series is transformed into a matrix of lag vectors. Consider a time series: $Y_n = (y_1, y_2, \ldots, y_n)$ where the length of n is more than 2. We will let L (which represents the length of the window) < n/2 and we will construct a trajectory matrix X from the L=lagged vectors of the original time series Y_n as follows:

$$X = [X_1, X_2, \ldots, X_K] \,, \tag{26}$$

$$X_i = (y_i, y_{i+1}, \ldots, y_{i-L+1})^T \,, \tag{27}$$

and L is the trajectory matrix window size, T stands for the transposition, and K= n-L+1 where $1 \leq i \leq K$. The trajectory matrix captures the dependence structure of the time series data within a sliding window of size L. The trajectory matrix of lagged vectors is mapped as follows:

$$X = \begin{bmatrix} y_1 & y_2 & \cdots & y_k \\ y_2 & y_3 & \cdots & y_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_L & y_{L+1} & \cdots & Y_n \end{bmatrix} , \tag{28}$$

where the matrix X is a Hankel Matrix as the elements on the off-diagonals (i+j=constant) are identical.

In the second step, we have the Singular Value Decomposition (SVD) of the trajectory matrix in Equation (28) which is obtained through the extraction of the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_L$ and eigenvectors $U_1, U_2, \ldots, U_L$ of the matrix the covariance matrix $S = XX^T$ of size L x L. The integration matrix is decomposed into a set of orthogonal series of components (eigenvectors) and the corresponding eigenvalues where the eigenvalues are arranged in decreasing order. The square roots of the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_L$ of length L in the matrix XX^T are called singular values of X, where the trace of the matrix $XX^T = \sum_{i=1}^{L} \lambda_i$. If we divide $A = \frac{XX^T}{\sum_{i=1}^{L} \lambda_i}$, we are

provided with important assumptions about the point and noise separation process. We must note that any time series can be written as $Y_n = \{y_t\}_{t=1}^{n} = S_n + E_n$, where S_n are the signal components and E_n the noise components. Considering that d=max{i} and setting the factor vectors $V_i = \frac{X^T U_i}{\sqrt{\lambda_i}}$, then the SVD of the trajectory matrix in Equation (28) can be stated as follows:

$$X = X_1 + X_2 + \ldots X_d \,, \tag{29}$$

where $X_i = \sqrt{\lambda_i} U_i V_i^T$ are elementary matrices with rank 1 and i=1,2,…,n. The collection { λ_i, U_i, V_i} is termed ith eigentriple in the SSA.

The third step is grouping, i.e. the process of selecting the eigentriples where the eigenvectors are grouped into component series according to the magnitude of their eigenvalues which is the basic criterion for grouping in SSA. In this step, we will proceed with the grouping, which partitions the set of elementary matrices indices {1,…,d} into m disjoint subsets $I_1, I_2, \ldots, I_m$(termed as eigentriple grouping). We will group $I_K = \{i_{k,1} + \ldots i_{k,p}\}$ where $1 \leq k \leq m$ and $1 \leq p \leq d$, so the resultant matrix associated with the group I_K can be defined as:

$$X_{I_k} = \{X_{i_{k,1}} + \ldots + X_{i_{k,p}}\} \,. \tag{30}$$

The components which form structured time series can be considered for the grouping. Subsequently, the grouping of the elementary series in Equation (29) results in the following decomposition:

$$X = X_{I_1} + X_{I_2} + \ldots X_{I_m} \tag{31}$$

In the fourth step, we have the reconstruction of the original time series by applying diagonal averaging to the m subsets of the grouped elementary matrices from Equation (31). We will assume that G is a matrix L x K with elements g_{ij} ,$1 \leq i \leq L$, $1 \leq j \leq K$, where

$L^* = min(L,k)$, $K^* = max(L,k)$ and $n = L + K - 1$. Applying the diagonal averaging procedure to the matrix G, the series $f = f_1, f_2, \ldots, f_n$ produced as in the following 3 cases:

- If $0 \le k \le L^* - 1$, then $f_{k+1} = \frac{1}{k+1} \sum_{m=1}^{k+1} g^*_{m,k-m+2}$. **(32)**

- If $L^* - 1 \le k \le K^*$, then $f_{k+1} = \frac{1}{L^*} \sum_{m=1}^{L^*} g^*_{m,k-m+2}$. **(33)**

- If $K^* \le k < n$, then $f_{k+1} = \frac{1}{n-K} \sum_{m=k-K^*+2}^{n-K^*+1+1} g^*_{m,k-m+2}$, **(34)**

where $g^*_{ij} = g_{ij}$ if $L < K$ and $g^*_{ij} = g_{ji}$ otherwise.

In Equations (32-34), we averaged the matrix elements along the diagonal $i + j = k + 2$. From the above diagonal averaging procedure of the resultant matrix in Equation (30), a sub-series $f^k = (f^k_1, f^k_2, \ldots, f^k_n)$ with length n is produced. Finally, the original time series $Y_n = (y_1, y_2, \ldots, y_n)$ we considered in the beginning, can be reconstructed by summation over the produced sub-series $f^k = (f^k_1, f^k_2, \ldots, f^k_n)$.

The reconstructed series are represented as follows:

$$\tilde{Y}_n = \sum_{k=1}^{m} f^n_n , \quad \textbf{(35)}$$

where $\tilde{Y}_n$ represents the reconstructed series. After the above 4 steps, we can proceed with the predictions with the method of Vector SSA or the method of Recurrent SSA. We will proceed to the method of the Vector SSA which is more widely applied and the basic condition for predicting the reconstructed time series is that the series follows a Linear Recurrent Formula (LRF).

3.13 Evaluation of models

Time series models are evaluated through forecasting errors to compare accuracy and measure how well they fit actual data, using metrics and visual comparisons. Choosing the model with the lowest forecasting error increases accuracy and informs decision-making. In models that require the selection of the number of lags such as ARIMA, the AIC/BIC criteria can be used, but we will continue by using the BIC criterion.

3.13.1 Akaike Information Criterion and Bayesian Information Criterion

The Akaike Information Criterion (AIC) is a statistical measure that we can use to see the relative quality of a model and also to compare different models for their relative quality. It measures the quality of the model in terms of its goodness of fit to the data, its simplicity, and how much it depends on tuning parameters. The formula for AIC is given by

$$AIC = 2k - 2ln(\hat{L}) \,, \tag{36}$$

where k is the number of model parameters and $ln(\hat{L})$ is the logarithm of the maximum value of the model's likelihood function. The Bayesian Information Criterion (BIC) Equation is defined as:

$$BIC = kln(n) - 2ln(\hat{L}) \,. \tag{37}$$

BIC is similar to AIC but is more strict when we have a large number of observations and is more robust to increasing degrees of lags. The criteria are useful not only for evaluating a model but also for comparing models with different numbers of parameters. The lower the value of the criteria, the better the model. We must note that for the comparison of models, we must use the same data set because these measures depend on the sample size.

3.13.2 Forecasting errors

The forecasting error represents the difference between predicted and actual values. To evaluate the accuracy of our electricity price forecasting models, we utilize various forecasting error metrics. These metrics allow us to compare the performance of different models and measure how well they fit the actual price data. We have to note that the difference between the below simple means and the absolute means is that the absolute means are resistant to extreme values.

3.13.2.1 Mean absolute error

The mean absolute error (MAE) is a measure of forecast error calculated as the mean absolute difference between the actual values and the predicted (or estimated) values i.e. it is the sum of the absolute differences between the predicted and actual values divided by the number of predictions. It measures the average size of the errors in a set of estimates or forecasts without regard to their direction. The MAE metric is widely used because it is easy to understand and interpret and is a relatively insensitive measure to outliers. However, it does not take into account the relative size of the errors. The formula for MAE is given by

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}, \tag{38}$$

where $\hat{y}_i$ is the prediction of y_i. The absolute difference between the predicted and actual values is taken to ensure that positive and negative differences are treated equally and the sum of the absolute differences is then divided by the number of observations to obtain the average magnitude of the errors.

3.13.2.2 Mean percentage error

The mean percentage error (MPE) is expressed as a percentage of the true value and is calculated by taking the average of the percentage error in a set of predictions or observations. However, MPE has a limitation when actual values are close to zero, as small errors can result in large percentage errors. The formula for MPE is given by

$$MPE = \frac{100\,\%}{n} \sum_{t=1}^{n} \frac{y_t - \hat{y}_t}{y_t} \,. \tag{39}$$

Again the lower the measure i.e. here the MPE the better the fit of our model to the real data.

3.13.2.3 Mean squared error

The mean square error (MSE) is calculated as the average of the squared differences between the actual and predicted values. It is a common metric for evaluating the performance of regression models but is sensitive to outliers, and unusually large or small errors and can be affected by the scale of the data. It is defined as the average of the squared differences between the predicted and actual values over all observations as follows:

$$MSE = \frac{1}{n} \sum_{t=1}^{n} (y_i - \hat{y}_i)^2 \,. \tag{40}$$

3.13.2.4 Root mean square error

Root mean square error (RMSE) is a useful metric because it is expressed in the same units as the original data, which makes it easier to interpret. Furthermore, RMSE is sensitive to large errors. It is defined as the square root of the mean of the squared differences between the predicted and actual values over all observations as follows:

$$RMSE = \sqrt{(MSE)} \,. \tag{41}$$

3.13.2.5 Mean absolute percentage error

Mean absolute percentage error (MAPE) is robust to outliers because it has the absolute value, but it can be difficult to interpret when actual values are close to zero as the error rate becomes infinite. MAPE represents by

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| . \quad \textbf{(42)}$$

CHAPTER 4: NORD POOL MARKET TIME SERIES ANALYSIS

4.1 Data Preprocessing

Before analyzing our time series we should mention that in cases where data were missing for an observation we used the mean of the intermediate observations to fill in the blank. Also, we fixed the problem where in some cases we had double observations at the same time. Our data concern the price of electricity and the predicted consumption in the Nordic countries, namely Norway, Sweden, Finland, and Denmark. Our prices refer to the System prices where according to Nord Pool the System prices for each hour are based on the intersection of the total supply and demand curves that represent all the offers in the entire Nordic market. The system price is the main component for tradable long-term Nordic financial contracts. Regarding the intended consumption we mean according to the needs of the consumer what the consumption will be.

4.2 Peak time selection

We will present the statistics for each hour separately from 2019 through 2020, namely the mean value and the standard deviation so that we can decide what time of the day it is the peak hour. In Table 4.1 we have the statistics for the consumption measured in megawatt-hour and in Table 4.2 we have the statistics for the price of electricity with currency in euro. From the data in Table 4.1 we conclude that the peak time is at 10 in the morning since at this time we observe the highest consumption.

TIME	MEAN	STANDARD DEVIATION	TIME	MEAN	STANDARD DEVIATION
00:00	39095.11	6350.72	12:00	46325.25	7268.98
01:00	38231.04	6360.63	13:00	45911.38	7245.93
02:00	37871.35	6394.96	14:00	45783.44	7355.28
03:00	37858.34	6511.36	15:00	45849.02	7636.57
04:00	38254.05	6795.77	16:00	46223.90	8006.15
05:00	39650.61	7253.52	17:00	46844.94	8190.30
06:00	42380.11	7987.87	18:00	46747.71	8044.53
07:00	45071.57	8570.11	19:00	46052.67	7691.08
08:00	46276.34	8265.45	20:00	44900.59	7263.80
09:00	46685.33	7812.87	21:00	43965.23	6922.07
10:00	**46901.00**	**7570.46**	22:00	42485.16	6537.58
11:00	46716.33	7392.86	23:00	40564.07	6335.64

Table 4.1: Statistics for forecasted consumption in the Nord Pool market for the years 2019-2020.

TIME	MEAN	STANDARD DEVIATION	TIME	MEAN	STANDARD DEVIATION
00:00	22.59	15.50	12:00	26.12	16.21
01:00	21.90	15.31	13:00	25.75	16.20
02:00	21.41	15.17	14:00	25.44	16.16
03:00	21.25	15.11	15:00	25.46	16.24
04:00	21.45	15.22	16:00	26.04	16.52
05:00	22.66	15.60	17:00	26.86	16.88
06:00	24.29	16.10	18:00	27.04	16.76
07:00	26.23	16.65	19:00	26.69	16.42
08:00	27.52	17.00	20:00	25.85	16.18
09:00	27.32	16.69	21:00	25.16	16.00
10:00	**26.98**	**16.50**	22:00	24.33	15.90
11:00	26.56	16.38	23:00	23.07	15.63

Table 4.2: Statistics for the electricity price in the Nord Pool market for the years 2019-2020.

4.3 Analysis of time series for the price of electricity

We will first present in Figure 4.1 how the forecasted consumption fluctuates during peak hours for the period from January 2019 to December 2020 and then how the price of electricity fluctuates for this period.

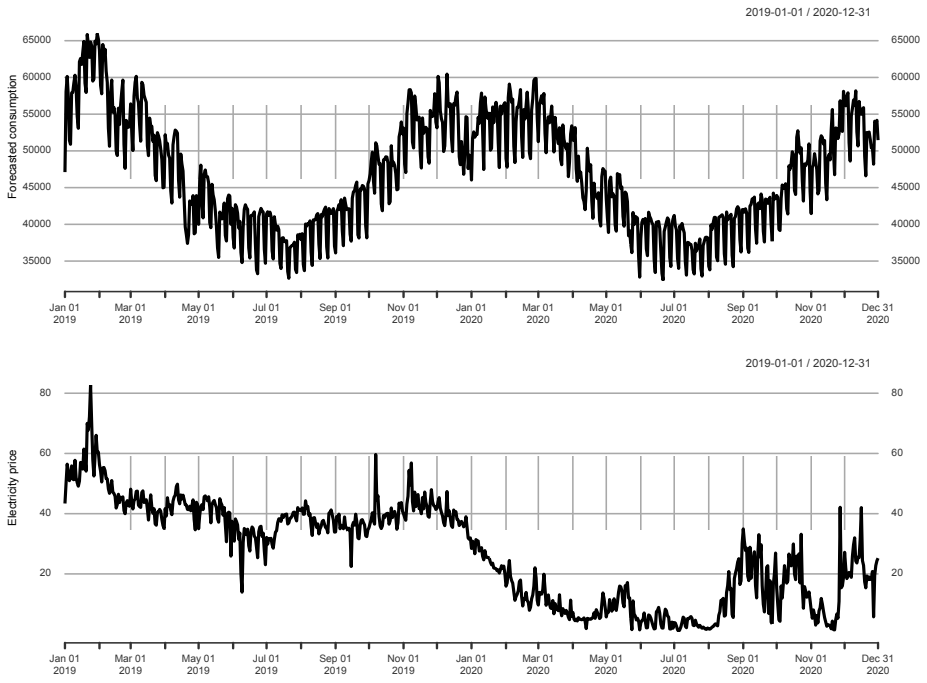

Figure 4.1: Electricity price and forecasted consumption for 2019-2020.

In both time series, we conclude that there is no stationarity there is no constant statistical properties over time such as constant mean and variance. Firstly, for the predicted consumption, it is obvious that consists of the characteristics of the same seasonality every year. Provided that the colder temperatures during the winter season typically from November to March increase the electricity demand, particularly for heating purposes, electricity prices tend to be highest during the winter season. In addition, the shorter daylight hours during winter also increase the demand for lighting which further contributes to the higher prices. On the other hand predicted consumption tends to be lower during the summer months, typically from June to August, due to

reduced energy demand for heating and lighting and also to the availability of renewable energy sources, such as wind and solar, which can help to bring down electricity prices.

In terms of price, there is a decrease in Nord Pool Nordics electricity prices in 2020 compared to 2019 which may have been caused by several factors. One of the most significant factors was the reduction in electricity demand due to the COVID-19 pandemic. The restrictions imposed in many countries, particularly during the second quarter of the year, resulted in a decrease in electricity consumption by businesses and industries. As a result, the supply of electricity was higher than the demand, leading to lower electricity prices. Another reason for the lower electricity prices in 2020 was the increase in renewable energy capacity in the Nordics, particularly in hydroelectric power where the increase in wind power capacity also contributed to the reduction in electricity prices. When renewable energy sources are producing more electricity, it can reduce the demand for other sources of electricity and lower prices. Also, a role in lowering electricity prices in 2020 played the decrease in fossil fuel prices such as natural gas or coal which are used to generate electricity in many regions. Finally, the Nord Pool electricity market is highly competitive, with prices determined by the balance of supply and demand. Increased competition or changes in market dynamics in 2020 may have also led to lower prices. Overall, the reduction in Nord Pool Nordics' electricity prices in 2020 was likely influenced by a combination of these and other factors. It may reflect both short-term changes in market conditions and longer-term trends in the energy sector.

Applying the ADF and KPSS test for the predicted consumption and price of electricity we have the following results in Table 4.3, left and right respectively.

	Predicted consumption			Price		
	Level	**Lag**	**p-value**	**Level**	**Lag**	**p-value**
ADF Test	-1.995	9	0.58	-2.491	9	0.37
KPSS Test	0.955	6	0.01	7.668	6	0.01

Table 4.3: ADF and KPSS tests for the Nord Pool.

In the ADF test, the null hypothesis in our test is non-stationarity or that the time series is dependent on time, i.e. it has a unit root, while the alternative hypothesis is the existence of stationarity or no dependence on time. In both cases, the p-value is greater than 0.05 and thus we accept the null hypothesis, so we have non-stationarity or our series can be reproduced by a unit root. Furthermore, the KPSS test helps us determine exclusively whether our time series is stationary so the null hypothesis in our test is the existence of stationarity, while the alternative hypothesis is the non-stationarity of our time series. In both cases, the p-value is less than 0.05 and thus we reject the null hypothesis, so we do not have stationarity.

4.4 Transformations

As we mentioned above, to proceed to the model estimation and forecasting for our time series we need to make our time series stationary, so we try to stabilize the variation of the time series and eliminate a possible trend by applying certain transformations to the data. The basic transformations which we can try are:

- Logarithmic transformation.
- Method of first differences.

We will present the transformed time series and we will apply stationarity tests to our transformed time series so that we can see the results of the transformations. First, we will test the Logarithm transformation and secondly the Method of first differences.

4.4.1 Logarithm transformation

The Logarithm transformation is a special case of the Box-Cox transformation where λ=0. In Table 4.4 we represent the KPSS test for the logarithm of the forecasted consumption and the logarithm of the price of electricity and after that, we represent Figure 4.2 which shows our time series with the logarithm transformation.

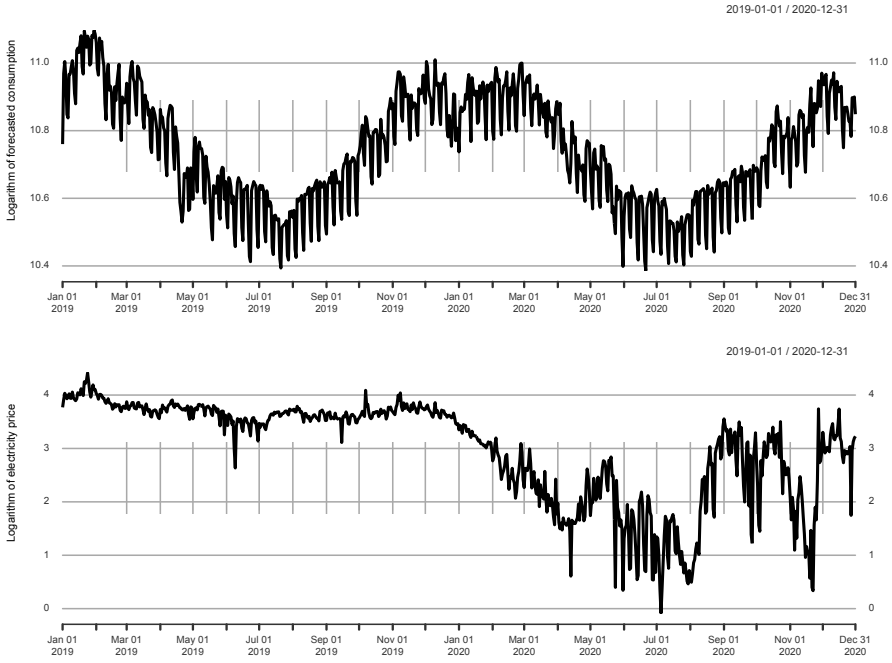

Figure 4.2: Logarithm transformation of forecasted consumption and electricity price for 2019-2020.

	KPSS test-statistic	lag parameter	p-value
Forecasted consumption	0.907	6	0.01
Electricity price	6.278	6	0.01

Table 4.4: KPSS tests for the Logarithm of our time series for 2019-2020.

We conclude that in the KPSS test for both time series, the p-value is less than 0.05 and thus we reject the null hypothesis, so we do not have stationarity. Also, in the plots for both time

series, there is no constant mean or variance so the logarithm transformation does not give us stationarity and we have to try the Method of first differences.

4.4.2 Method of first differences

In Table 4.5 we represent the KPSS test for the predicted consumption and price of electricity with the first differences and after that, we represent Figure 4.3 which shows our time series with the first differences. From the results, we conclude that in the KPSS test in both time series, the p-value is more than 0.05 and thus we do not reject the null hypothesis, so we have stationarity.

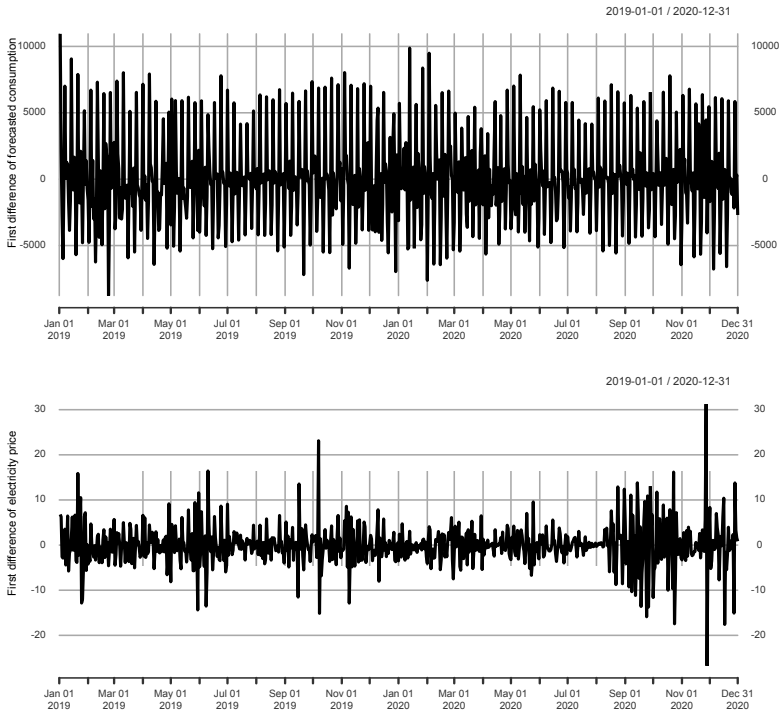

Figure 4.3: Forecasted consumption and electricity price first differences for 2019-2020.

	KPSS test-statistic	lag parameter	p-value
Forecasted consumption	0.077	6	0.1
Electricity price	0.001	6	0.1

Table 4.5: KPSS tests for the first difference of the time series for 2019-2020.

As for the specific models we will run in our analysis (seasonal or not) in which we will use the original data: ARIMA and mixed ARIMA-GARCH, we expect that our best model in each case will have d=1 (which is the number of differences representing the number of times the data has been differenced to be stationary).

4.5 ACF plot of our data after the transformation

In Figure 4.4 we present the ACF of our data with the first difference before running any model so that we understand the characteristics of our data. This will help us to identify if there is any trend or pattern in our data that must be addressed subsequently in the modeling process that we will apply.

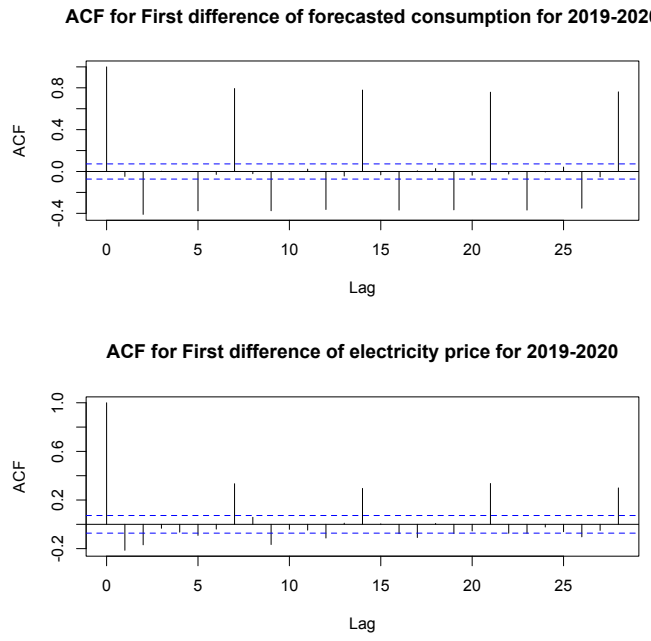

Figure 4.4: ACF for forecasted consumption and electricity price for 2019-2020.

Although our series with the first difference led to stationarity, moving to the ACF plot we observe that there are out-of-bounds lags in specific periods. For the price of electricity, we see a lag beyond limits at every multiple of 7, so we see seasonality that seems to be weekly. In the forecasted consumption we see the same cycle of seasonality, where there is an out-of-bounds lag in every multiple of 7 (weekly seasonality where the length of the seasonal cycle s is 7) but also in the 2nd and 5th lag of each cycle. This makes sense because, for data such as the electricity price, the weekly seasonality is common, as demand patterns tend to vary depending on the day of the week. For example, electricity prices may be higher on weekdays when commercial and industrial demand is highest and lower on weekends when demand is lower. So in addition to the ARIMA, we must also try the SARIMA model to see if using seasonality with frequency=7 in our model we get better results.

CHAPTER 5: APPLICATION OF MODELS IN NORD POOL

The purpose of the following chapter is to apply in-sample estimation and out-of-sample forecasting models for the price of electricity in the Nord Pool market. We will proceed with the performance of our models, where the ACF plots for the estimations are presented so that we can assess the quality of our model and examine whether there is autocorrelation present in our time series. As we said before, this way will help us to determine if our model has adequately captured the patterns or trends we found by applying the ACF charts to our data before performing any model and thus we will examine for any remaining autocorrelation in the model residuals. Then we will make predictions out-of-sample for the peak time in the days of 2020.

5.1 Parametric models

5.1.1 Simple linear regression model

The first model we will use is the SLR which is a model with only one explanatory variable. That is, it takes one independent variable and one dependent variable and finds a linear function (a non-vertical straight line) that predicts as accurately as possible the values of the dependent variable as a function of the independent variable. First, we estimate using the least squares method for the period from 1 January 2019 to 31 December 2019. The function applied by the model is:

$$\Delta Price_t = \beta_0 + \beta_1 \Delta PredictedConsumption_t + \varepsilon_t,$$

where β_0 is a constant, β_1 is the coefficient for the Δpredicted consumption, $\Delta Price_t$ is the electricity price with the first difference, and $\Delta PredictedConsumption_t$ the first difference of predicted consumption.

From the application of the model, we have the following results in Table 5.1, where df represents the degrees of freedom.

	Coefficients Estimation	SE	t value	p-value
Intercept	-0.003	0.162	-0.223	0.824
ΔPredicted consumption	0.001	0.001	17.361	0.001

Table 5.1: Results for the coefficients of the SLR model.

RSE	Multiple R^2	Adjusted R^2	F-statistic	p-value
3.094 on 362 DF	0.454	0.452	301.4 on 1 and 362 DF,	0.001

Table 5.2: Results of the quality of the SLR model.

From the statistical inference Table 5.1, the function is written as follows:

$$\Delta Price_t = -0.003 + 0.001 \Delta PredictedConsumption_t.$$

First, we observe the estimates for the coefficients, where the constant β_0 is -0.003 and β_1, i.e. the coefficient for the Δpredicted consumption is 0.001. We see that there is a statistically significant correlation (since we have a p-value lower than 0.05) for the Δpredicted consumption and affects the price of electricity but no statistically significant correlation for the constant. Continuing with the quality of the linear regression fit, we will first analyze the RSE which is equal to 3.094 and means that the observed values deviate from the true regression line by about 3.094 units on average, so it is considered average to bad (as first difference of electricity price has prices from scale almost 20 to -20). Next, we will analyze the adjusted R^2 which is close to 0 (since it is 0.452), and shows us that the regression model does not explain much of the variability in the results. Finally, the F-statistic gives us the overall significance of the model and estimates whether at least one predictor variable has a non-zero coefficient. A large F-statistic corresponds to a statistically significant p-value (<0.05) and in the model F-statistic equals 365.5 which produces a p-value of 0.001, which is highly significant. The F-test is identical to the t-squared test: $301.4 = (17.361)^2$.

Subsequently, we want to convert our SLR results for the first difference of price to the actual data of price and plot the estimations of the above SLR model in actual data. Converting the Function from $\Delta Price_t$ to $Price_t$, it is written as:

$$\Delta Price_t = Price_t - Price_{t-1},$$

$$Price_t = \Delta Price_t + Price_{t-1},$$

$$Price_t = -0.003 + Price_{t-1} + 0.001(PredictedConsumption_t - PredictedConsumption_{t-1}).$$

Next, we will present Figure 5.1 with the in-sample estimations of the electricity price (not with the first difference). The actual and estimated observations move together when we have sharp changes and have a little deviation in the remaining observations. Also, from the quantitative characteristics of the estimations of our time series, we get the following forecasting errors in Table 5.3.

MAE	RMSE	MAPE	MSE
2.012	2.819	0.050	7.951

Table 5.3: In-sample forecasting errors with SLR.

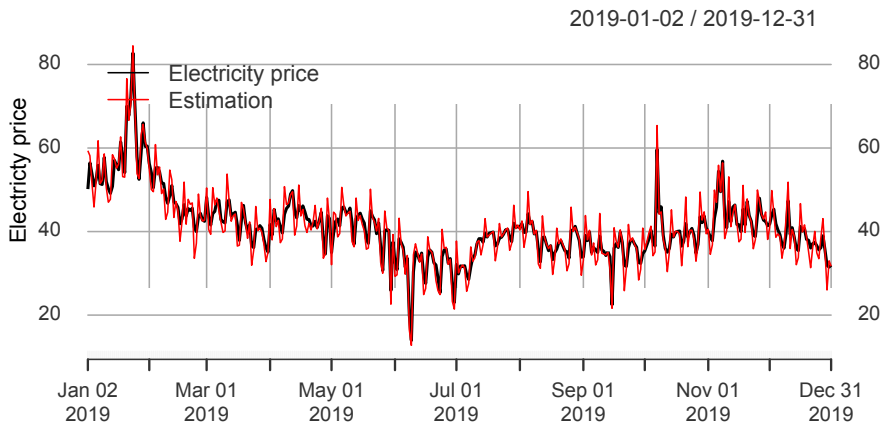

Figure 5.1: Estimation of the electricity price in 2019.

Next, we have to look at the residuals through the ACF and PACF and the Ljung-Box test which tests whether a time series contains autocorrelation. Looking at the autocorrelation and partial autocorrelation diagrams for the residuals we notice that we have lags out of bounds so we do not have white noise.

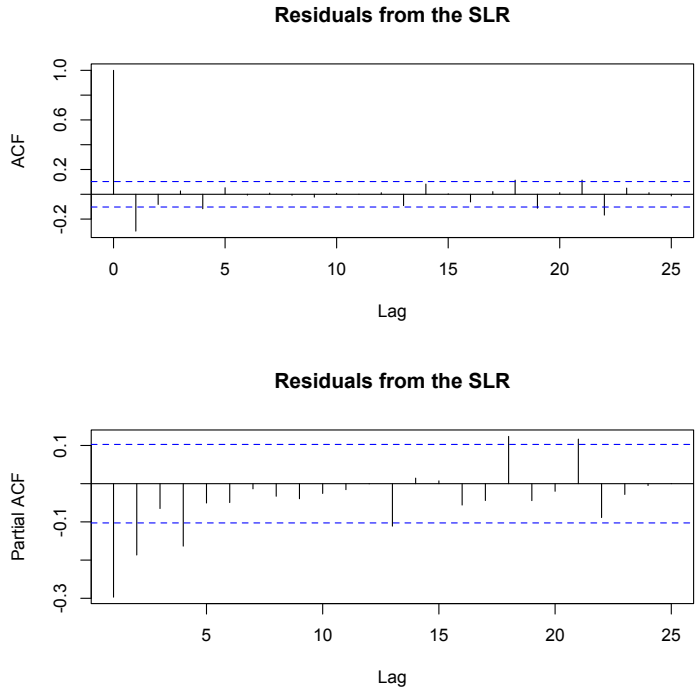

Figure 5.2: ACF and PACF respectively for in-sample estimations through the SLR model for 2019.

Looking at the diagrams for the residuals in Figure 5.2, we notice that we have lags out-of-bounds and are not characterized by white noise as they should be to be a good prediction model. We can confirm this with the Ljung-Box test in Table 5.4, where the null hypothesis is that the residuals are independently distributed and the alternative hypothesis is that they are correlated. Furthermore, the lag in each test is the same as the degrees of freedom. In the test from the 1st to the 10th lag, our p-value is 0.001, so we reject the null hypothesis and there is a correlation between the residuals.

Lag	X^2	p-value
1	32.382	0.001
2	34.855	0.001
…		
10	41.489	0.001

Table 5.4: Ljung-Box test of the SLR model estimations.

Proceeding to out-of-sample forecasting with a rolling forecast, we expect that we will not get good forecasts. Applying the one-step-ahead forecasts for all of 2020 we get the actual and predicted data in Figure 5.3, where again the actual and estimated observations move together when we have sharp changes and have a little deviation in the remaining observations. Finally, in Table 5.5 we present the results for the out-of-sample forecast errors.

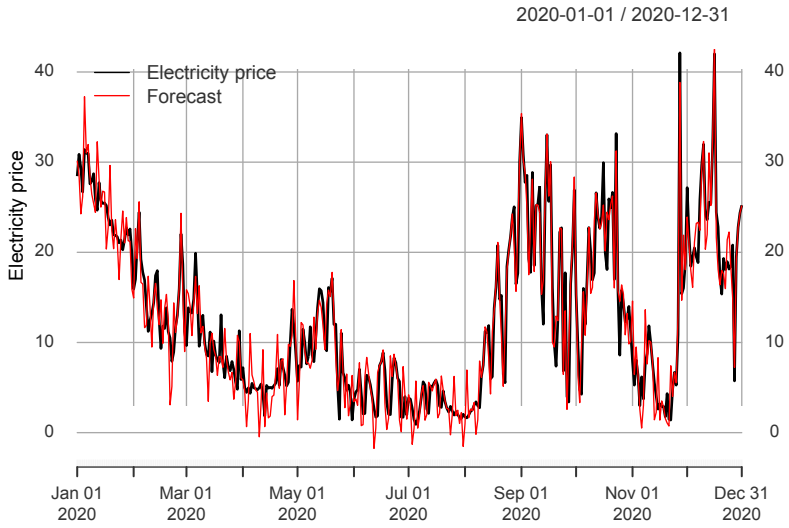

Figure 5.3: Electricity Price Prediction out-of-sample by the SLR model.

MAE	RMSE	MAPE	MSE
1.971	2.767	0.551	7.656

Table 5.5: Out-of-sample forecasting errors with the SLR model.

5.1.2 Autoregressive Integrated Moving Average model

Next, we will apply the ARIMA model but first, we have to choose the best ARIMA model by testing different models with drift or not. The BIC from each ARIMA model order is presented in Table 5.6. We want to choose the ARIMA with the minimum value of BIC, so we will continue with the ARIMA model with order: p=1, d=1, and q=2. After we proceed to the estimation of the best model, we will show the results of the estimated coefficients in Table 5.7 wherein in the first row we have the coefficients and in the second row we have the SE of the coefficients.

ORDER	BIC	ORDER	BIC
ARIMA(2, 1, 2) with drift	2014.217	ARIMA(2, 1, 1) with drift	2014.262
ARIMA(0, 1, 0) with drift	2074.985	ARIMA(2, 1, 3) with drift	2015.967
ARIMA(1, 1, 0) with drift	2059.228	**ARIMA(1, 1, 2)**	**2011.587**
ARIMA(0, 1, 1) with drift	2045.169	ARIMA(0, 1, 2)	2016.876
ARIMA(0, 1, 0)	2072.983	ARIMA(1, 1, 1)	2011.968
ARIMA(1, 1, 2) with drift	2012.996	ARIMA(2, 1, 2)	2012.963
ARIMA(0, 1, 2) with drift	2018.667	ARIMA(2, 1, 1)	2012.948
ARIMA(1, 1, 1) with drift	2013.186	ARIMA(1, 1, 3)	2013.415
ARIMA(1, 1, 3) with drift	2014.793	ARIMA(0, 1, 1)	2043.244
ARIMA(0, 1, 3) with drift	2020.186	ARIMA(0, 1, 3)	2018.396

Table 5.6: BIC for each ARIMA order.

AR(1)	MA(1)	MA(2)
0.1470	-0.5015	-0.2075
0.1877	0.1852	0.1125

Table 5.7: Coefficients and their SD for ARIMA(1, 1, 2).

We want to accurate the estimates in-sample for 2019, so we will present Figure 5.4 which represents the actual data for electricity prices in 2019 and the estimated by ARIMA(1, 1,

2). Also, we will show Table 5.8. with the results from the forecasting errors for the estimated period.

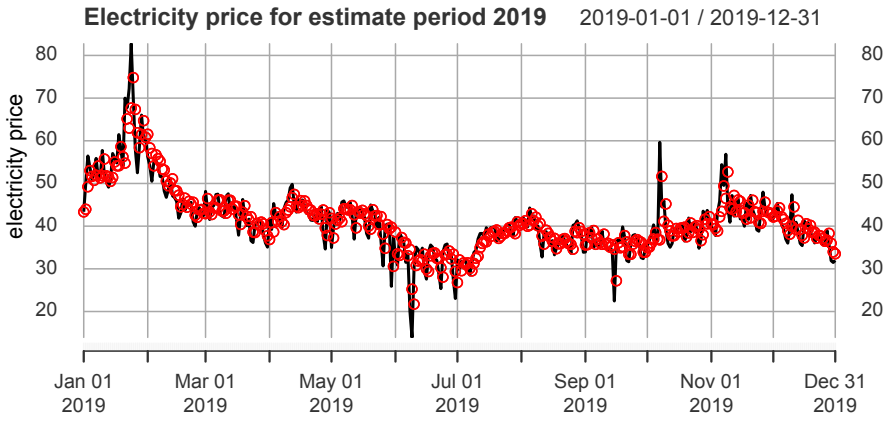

Figure 5.4: Estimation with ARIMA(1, 1, 2).

MAE	RMSE	MAPE	MSE
2.707	3.822	0.069	14.609

Table 5.8: In-sample forecasting errors with ARIMA(1, 1, 2).

We notice that the actual and estimated values move together with little deviation when there are large fluctuations, so the ARIMA(1, 1, 2) described well the data of electricity price. Next, we have to look at our residuals through the autocorrelation plot in Figure 5.5 and the Ljung-Box test which tests whether a time series contains autocorrelation in Table 5.9 in which we test for the lags separately up to the 8th lag and if we reject the null hypothesis our residuals will not be characterized by white noise. The null hypothesis is that the residuals are independently distributed and the alternative hypothesis is that the residuals are not independently distributed and show a serial correlation.

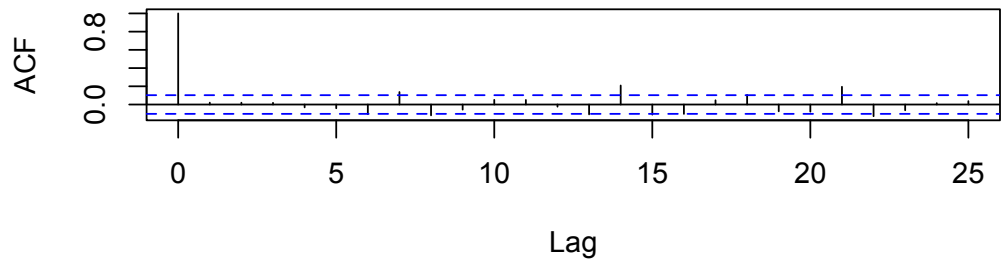

Figure 5.5: ACF from the regression with the ARIMA(1, 1, 2).

Lag	X^2	p-value	Lag	X^2	p-value
1	0.004	0.948	5	3.761	0.584
2	0.034	0.982	6	3.906	0.689
3	0.034	0.998	7	49.071	0.001
4	2.200	0.699	8	53.915	0.001

Table 5.9: Ljung-Box for the ARIMA(1, 1, 2).

From the results in Table 5.9 we do not have white noise because, in the Ljung-Box test, we have a correlation of the residuals. Furthermore, from Figure 5.5 we have statistically significant autocorrelation outside the confidence limits for the 7th, 14th, and 21 lags, so as we said before, we are thinking that maybe there is a weekly seasonality, so we have to test the SARIMA.

We will proceed to the rolling forecast for 2020 through rolling windows for 1 step forward and the range of our window will be equal to one year so since we want to forecast 2020 we will use the data from 2019. Applying the rolling forecast, we can see in Figure 5.6 our actual data for 2019-2020 and the predicted data for the year 2020. Furthermore, we show Table 5.10 which includes the forecasting errors out-of-sample.

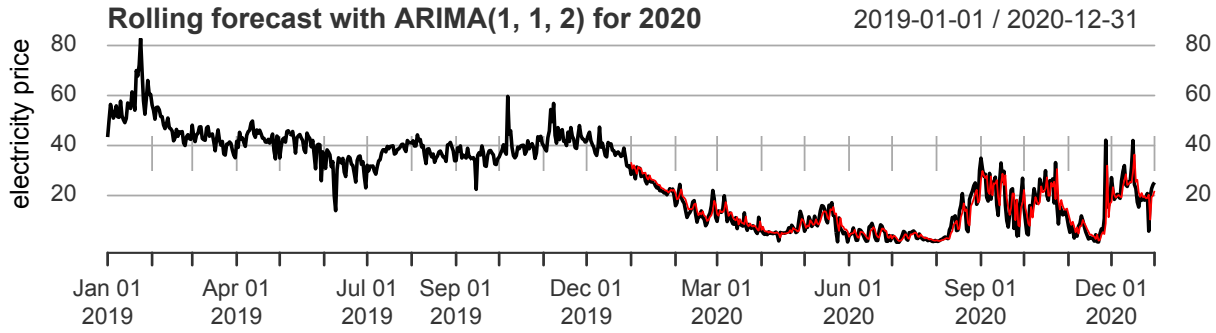

Figure 5.6: Rolling forecast for 2020 with ARIMA(1, 1, 2).

MAE	RMSE	MAPE	MSE
2.995	4.529	0.338	20.512

Table 5.10: Out-of-sample forecasting errors with ARIMA(1, 1, 2).

5.1.3 Seasonal Autoregressive Integrated Moving Average model

Next, we will try the SARIMA model but firstly we have to denote the frequency equal to 7 (for weekly seasonality as we saw in the lags of ACF plots). Denoted seasonal = True, the best model is the ARIMA(4, 1, 1)(2, 0, 0)[7]. So the introduction of seasonality in our data, change the optimal ARIMA replacing it with SARIMA with p=4, d=1, q=1, seasonal P=2, D=O, and Q=0 with frequency=7. After we proceed to the regression of the best model, we will show the results of the regression coefficients in Table 5.11.

	AR(1)	AR(2)	AR(3)	AR(4)	MA(1)	SAR(1)	SAR(2)
Coefficients	0.6083	0.1082	0.1213	-0.0439	-0.9864	0.3008	0.3217
SE	0.0544	0.0624	0.0616	0.0545	0.0131	0.0513	0.0514

Table 5.11: Coefficients and SE for SARIMA.

We want to check the accuracy of the estimates in-sample for 2019, so we will present Figure 5.7 which represents the actual data for electricity prices in 2019 in black color and the

estimated by SARIMA in red color. Also, we will show the table with the results from the forecasting in-sample errors for the estimated period in Table 5.12.

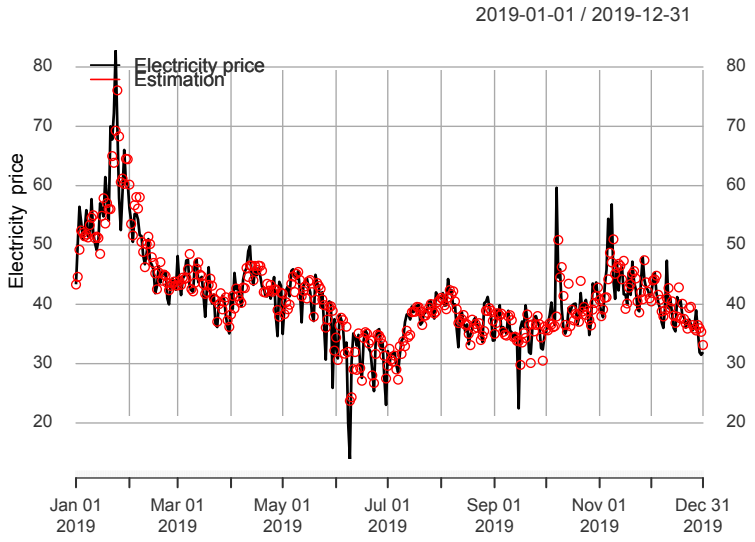

Figure 5.7: Forecasting in-sample with SARIMA.

MAE	RMSE	MAPE	MSE
2.252	3.347	0.056	11.203

Table 5.12: In-sample forecasting errors with SARIMA.

We notice that the results are similar to the ARIMA(1, 1, 2) with minimal better in-sample forecasting errors. Next, we have to look at our residuals through the autocorrelation plot in Figure 5.8 and the Ljung-Box test which tests whether a time series contains autocorrelation in Table 5.13 in which we test for the lags separately up to the 10th lag and if we reject the null hypothesis our residuals will not be characterized by white noise.

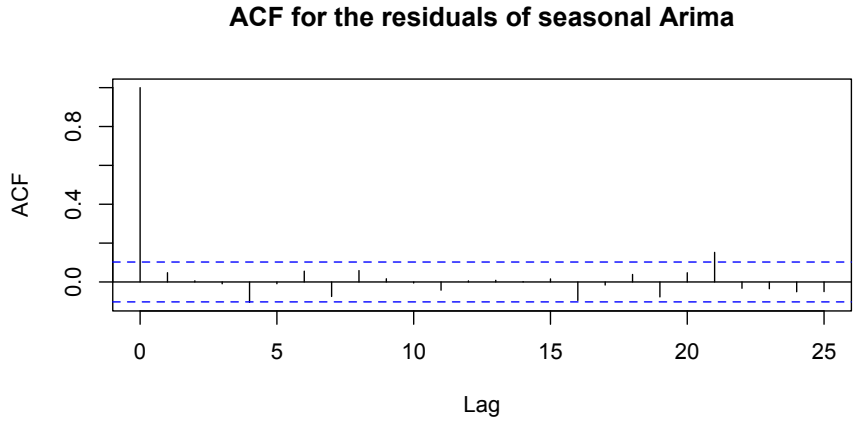

Figure 5.8: ACF from the regression with SARIMA.

Lag	X^2	p-value
1	0.002	0.957
2	0.026	0.986
3	0.234	0.971
…		
10	7.472	0.680

Table 5.13: Ljung-Box for SARIMA.

From the results in Table 5.13 and Figure 5.8, we have no correlation of the residuals because in the Ljung-box we have a p-value larger than 0.05 for all lags and in the ACF plot we don't have lines outside the blue bar (except lag 21), so the SARIMA(4, 1, 1)(2, 0, 0) describes very well the electricity price.

We will proceed to the rolling forecast for 2020 through rolling windows for 1 step forward and the range of our window will be equal to one year so since we want to forecast 2020 we will use the data from 2019. Applying the forecast, we can see in Figure 5.9 our actual data for 2019-2020 and the forecasted rolling window data for the year 2020. Furthermore, we show

Table 5.14 which includes the out-of-sample forecasting errors which are similar to the forecasting errors from the ARIMA(1, 1, 2).

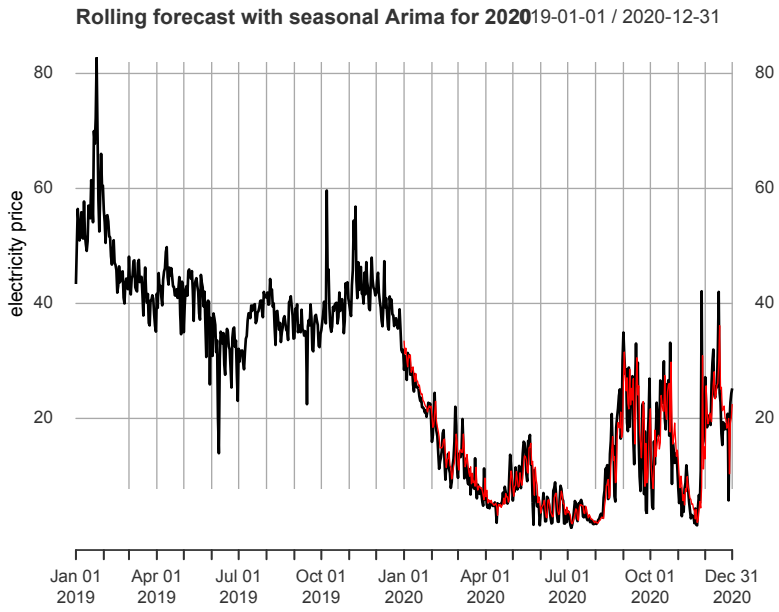

Figure 5.9: Out-of-sample forecast with SARIMA.

MAE	RMSE	MAPE	MSE
2.979	4.516	0.331	20.402

Table 5.14: Out-of-sample forecasting errors with SARIMA(4, 1, 1)(2, 0, 0).

5.1.4 Autoregressive/Generalized Autoregressive Conditional Heteroskedasticity

The model we used to decide whether we need to apply the ARCH or GARCH model is the SARIMA that we applied above. Applying the steps for checking for GARCH in SARIMA we get the information about the squared residuals of SARIMA, the ACF, and PACF plots in Figure 5.10. In summary, ACF and PACF plots have significant lags out of the limits (at 1st and 3rd lag), and in the squared residuals plot, there are signs of volatility. So the residuals show

some patterns that need to be modeled. Then, Table 5.15 shows the application of the ARCH test to the residuals, where the null hypothesis is homoscedasticity and the alternative hypothesis is heteroscedasticity in the residuals. The p-value is equal to 0.008, so we reject the null hypothesis, and the residuals for SARIMA are characterized by ARCH heteroscedasticity up to lag 2. Consequently, we will proceed to find the appropriate ARCH/GARCH model and in Table 5.16 we present the results for the different orders of the GARCH model.

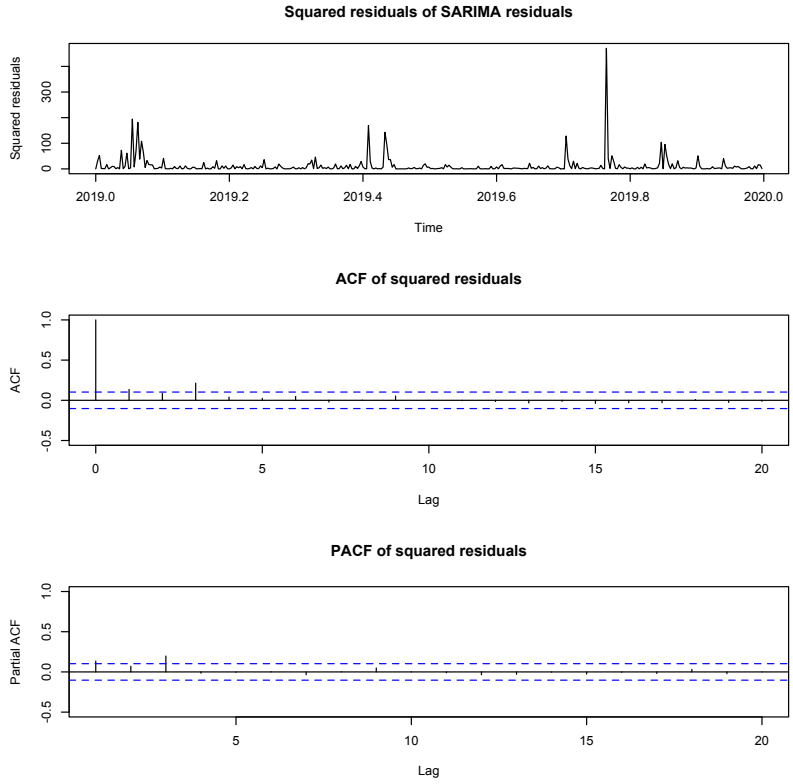

Figure 5.10: Analysis of squared residuals of SARIMA.

Lag	X^2	p-value
2	9.608	0.008

Table 5.15: ARCH test on residuals of SARIMA.

MODEL	AIC	CONVERGENCE RESULTS
GARCH(0, 1)	1906.162	RELATIVE FUNCTION CONVERGENCE
GARCH(0, 2)	1895.38	RELATIVE FUNCTION CONVERGENCE
GARCH(0, 3)	**1856.6**	**RELATIVE FUNCTION CONVERGENCE**
GARCH(0, 4)	1865.918	FALSE CONVERGENCE
GARCH(1, 0)	1924.125	FALSE CONVERGENCE
GARCH(1, 1)	1892.865	FALSE CONVERGENCE
GARCH(1, 2)	1885.757	FALSE CONVERGENCE
GARCH(2, 1)	1900.858	FALSE CONVERGENCE

Table 5.16: Results for GARCH model orders.

The GARCH(p, q) model includes p terms from ARCH and q terms from GARCH. The optimal model we will use is GARCH(0, 3) which has the smallest AIC with relative function convergence. In Table 5.17 we present the results for the coefficients of GARCH(0, 3) where in the first column we have the intercept and the parameters from Equation (23), in which the intercept and parameter a_3 are statistically significant. Also, in Table 5.18 we present the results for the residuals Ljung-Box test of GARCH(0, 3), where the p-value is 0.666 and we cannot reject the null hypothesis that the autocorrelation of residuals is equal to 0. The model thus adequately represents the residuals.

	Coefficients Estimation	**SE**	**t value**	**p-value**
Intercept	6.238	0.442	14.096	0.001
a_1	0.077	0.040	1.908	0.056
a_2	0.038	0.035	1.077	0.281
a_3	0.368	0.050	7.327	0.001

Table 5.17: Results for the coefficients of the GARCH model.

Lag	X^2	**p-value**
1	0.185	0.666

Table 5.18: Ljung-box test for the squared residuals of the GARCH model.

Subsequently, we will present in Figure 5.11 the in-sample forecasting[1] where our actual data move with the estimated data so our model is very good. In Table 5.19 we present the forecasting errors. Then, we will show in Figure 5.12 the Q-Q plot of the residuals of GARCH which shows that the residuals are normally distributed while there are some outliers.

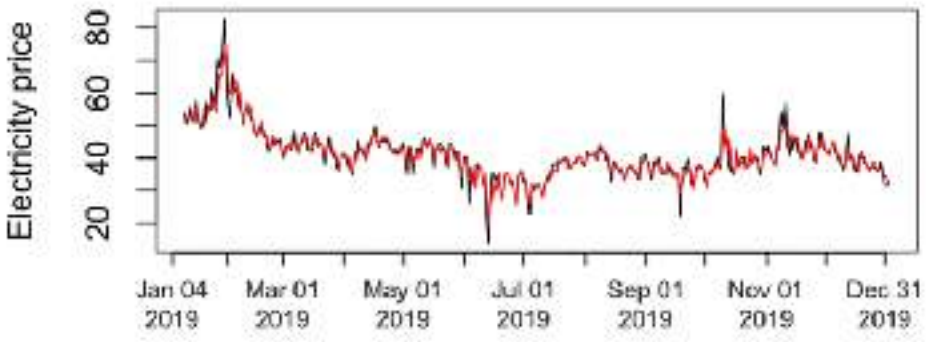

Figure 5.11: In-sample forecast with GARCH.

MAE	RMSE	MAPE	MSE
1.564	2.423	0.038	5.872

Table 5.19: In-sample forecasting errors with GARCH.

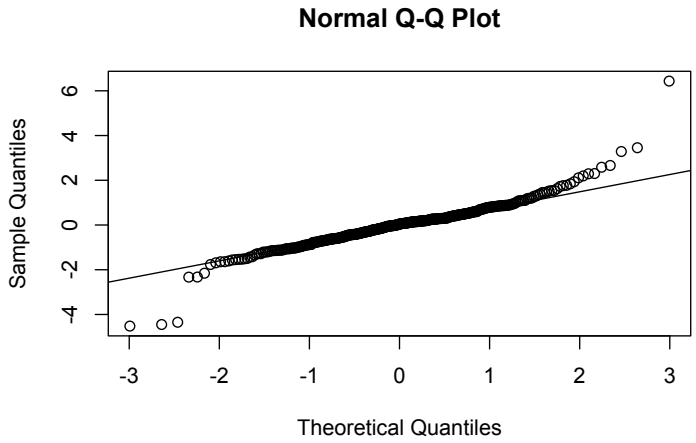

Figure 5.12: Q-Q plot of GARCH residuals.

[1] Black color represents the real data and red color the in-sample forecasted data.

Finally, we will proceed to forecast out-of-sample for 2020 through rolling windows for 1 step ahead and the range of our window will be equal to one year. In Figure 5.13 we present the predictions[2] for the forecasting out-of-sample where the black color represents the real data and the red color the out-of-sample forecasted data. Our actual data move near the predicted data but not as well as in the in-sample forecasts. Finally, in Table 5.20 we present the out-of-sample forecasting errors.

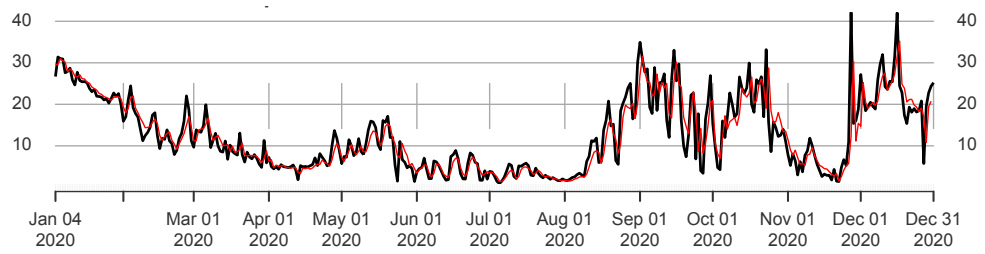

Figure 5.13: Out-of-sample forecast with GARCH.

MAE	RMSE	MAPE	MSE
2.455	3.861	0.261	14.912

Table 5.20: Out-of-sample forecasting errors with GARCH.

5.2 Non-parametric prediction model

5.2.1 Singular Spectrum Analysis Model

After the parametric models, follows the application of the non-parametric SSA model which does not require stationarity or other characteristic and so applied to the original data. Firstly, we will make an in-sample estimate of the price of electricity in the Nord Pool market for the period 2019 and we will take the in-sample forecasting errors to see how accurate our

estimates are with this model. Next, using a rolling window we shall make an out-of-sample forecast for the electricity price for all days of the year 2020 and we will use the actual data for 2020 to evaluate the accuracy of the model.

It is important to note that the best value of L depends on the goals of the analysis and the characteristics of the data, so it may be useful to experiment with different values of L to see how they affect the results of the analysis. Using the daily data, N is equal to 365 days since we have observations every day for 1 year. It has been shown that the appropriate window length L is the median, or slightly less than half of N. In Table 5.21 we have the execution of SSA.

Series length	Window length
365	182

Table 5.21: SSA for 2019.

To decide the number of eigentriples we need for the reconstruction step we will study the eigenvalue diagram, the eigenvector diagram, and the correlation matrix. The required number of eigentriples is determined by the characteristics of the voltage and harmonic components, which provide us with information and must remain, while the noise components must remove from the reconstruction process.

Figure 5.14 shows the first eight eigenvectors, where we can identify a trend and periodic components and the absence of noise components. We notice that the first eigenvector has almost constant coordinates and therefore corresponds to pure smoothing, while the following have fluctuations. Then, Figure 5.15 presents the result of reconstruction by each of the eight eigentriples. Both figures confirm that the behavior of the eigentriples is interpreted as a trend (1st eigentriple) and harmonic components (the rest eigentriples). The first eigenvector has a large percentage (97.79%), the second eigenvector 0.76%, the third 0.25%, the fourth 0,19%, and the rest have smaller percentages.

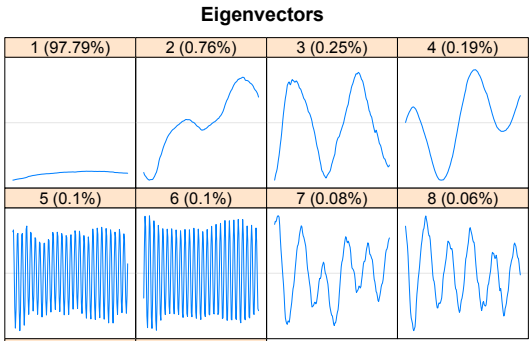

Figure 5.14: Eigenvectors for L = 182.

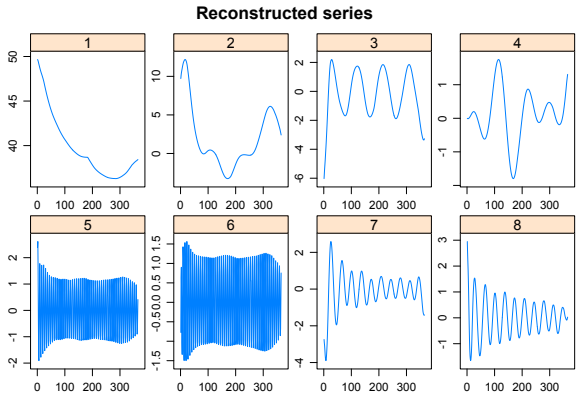

Figure 5.15: Elementary reconstructed series for L = 182.

As we didn't identify noise components so far, we will use the plot of eigenvalues and the correlation matrix to choose the required number of eigenvalues. Figure 5.16 on the left presents the first 50 eigenvalues (for better number reading) and on right all the eigenvalues. Also, Figure 5.17 on the left presents the correlation matrix for the first 50 eigenvalues (for better number reading), and on right the correlation matrix for all eigenvalues.

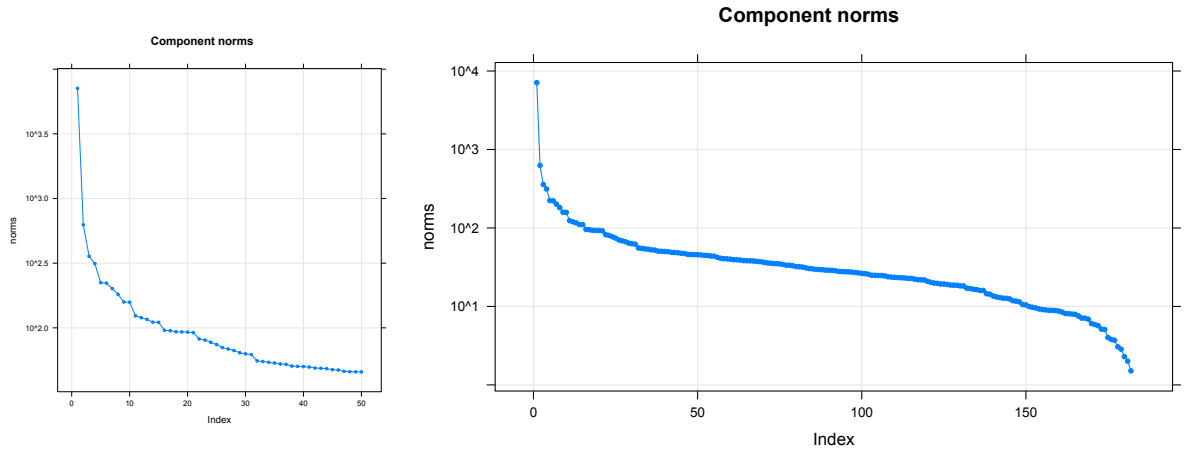

Figure 5.16: Eigenvalues for L = 182 in the original series.

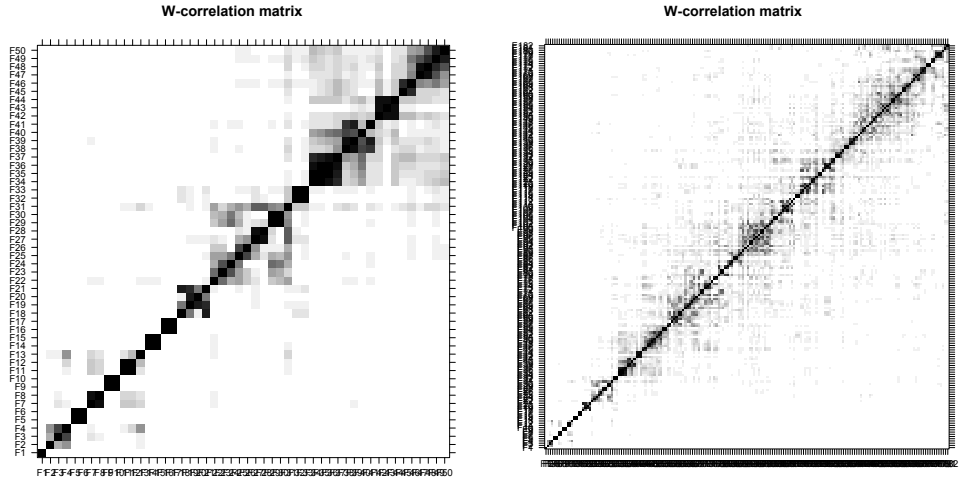

Figure 5.17: Correlation matrix for L = 182 in the original series.

From Figure 5.16 we can see several steps produced by approximately equal eigenvalues until the 22 eigenvalues[3] (except for the first 4 eigenvalues and the pair {7,8}), where each step is likely to be yielded a pair of eigenvectors. After the leading 22 singular values, we have a gap and then there is a slowly decreasing sequence of the remaining singular values without clear pairs or eigenvectors. Then, the correlation matrix in Figure 5.17 can equally show us the pairs to do the appropriate grouping, as well as the separability analysis. It has a scale of 20 colors from

[3]Where exists the black line.

black to white with values from 1 to 0, respectively. On both sides with the black line, we note the same point which is again in the 22 eigenvalues and we can separate the noise components on the right square of each correlation matrix.

Subsequently, we have to reconstruct the trend and the useful components that give us pieces of information, so we will use the first 22 components and the result presents in Figure 5.18, where the black color represents the actual data and the red color the trend (and useful components) line. Also, the time represents the 365 days of 2019. The next stage after we have extracted the trend is the extraction of seasonality from the residuals. We will use again the eigenvalues which are present in Figure 5.19 and the correlation matrix w of the main components in Figure 5.20.

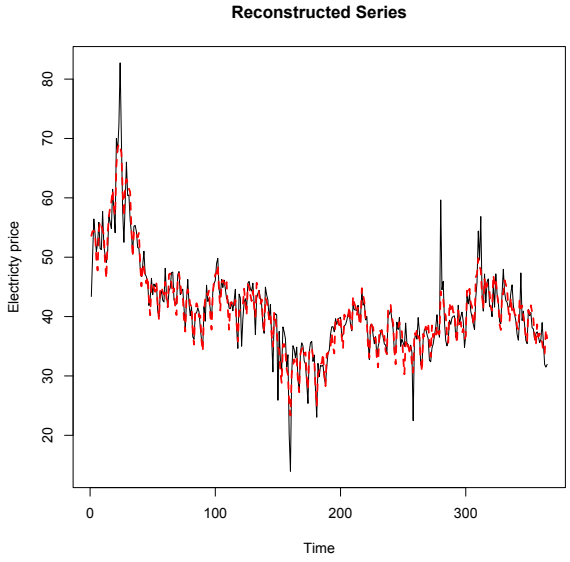

Figure 5.18: Trend and harmonic components reconstruction.

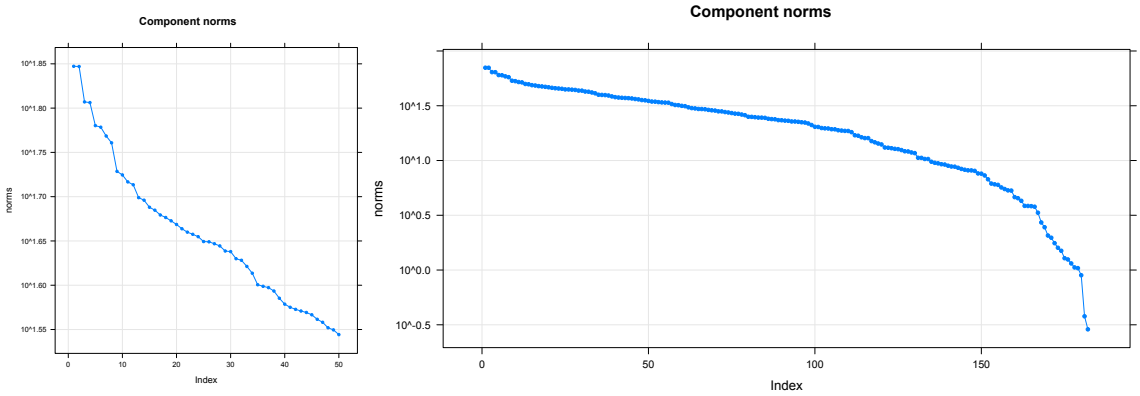

Figure 5.19: Eigenvalues for L = 182 after trend and harmonic components reconstruction.

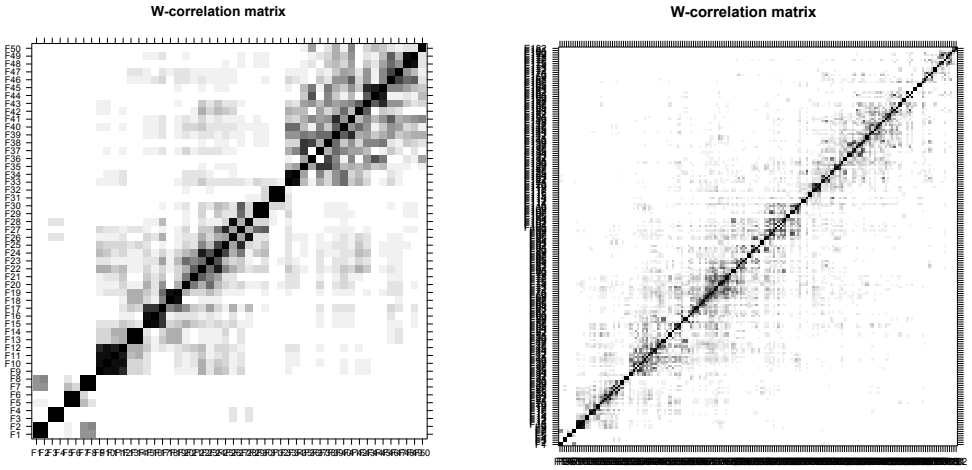

Figure 5.20: Correlation matrix for L = 182 after trend and harmonic components reconstruction.

Figure 5.19 on the left presents the first 50 eigenvalues (for better number reading) and in right all the eigenvalues, where the black line it's the same point which is in the 8 eigenvalues. On the left side we can see that after the leading 8 singular values, we have a slowly decreasing sequence of the remaining singular value. On the right side, we can see several steps produced by approximately equal eigenvalues until the 8 eigenvalues and after the leading 8 singular values, we can see an obvious gap. Also, Figure 5.20 on the left presents the correlation matrix for the first 50 eigenvalues (for better number reading), and in right the correlation matrix for all eigenvalues. On both sides with the black line, it's the same point which is again in the 8

eigenvalues (as in Figure 5.19) and we can separate the noise components in the right square of each side. So the required eigenvalues for the grouping option to establish the seasonality is 8. Therefore, we will examine whether our non-parametric model provides a good estimate of the price of electricity. Thus we would present in Figure 5.21 the in-sample forecasts with SSA, where we see that SSA has the best in-sample forecast as the estimated electricity prices move together with the actual data except for a few observations. The results for the in-sample forecasting errors are presented in Table 5.22.

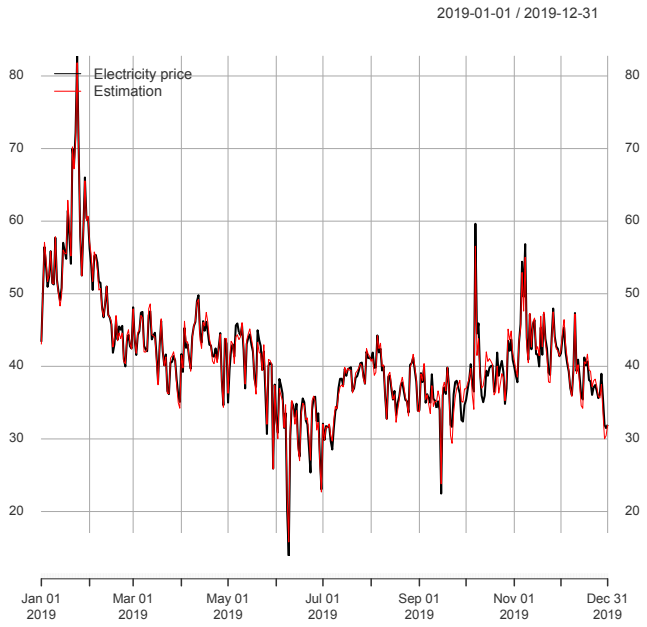

Figure 5.21: In-sample forecast with SSA.

MAE	RMSE	MAPE	MSE
0.7392	0.954	0.019	0.911

Table 5.22: In-sample forecasting errors with SSA.

Finally, we will proceed to forecast the electricity price out-of-sample for 2020 through rolling windows for 1 step ahead and the range of our window will be equal to one year so since we want to predict 2020 we will use the data of 2019. Figure 5.22 presents the forecasts and the

results for the evaluation measures are as follows in Table 5.23. As in the in-sample forecast, we see that SSA has the best out-of-sample forecast as the estimated electricity prices move together with the actual data.

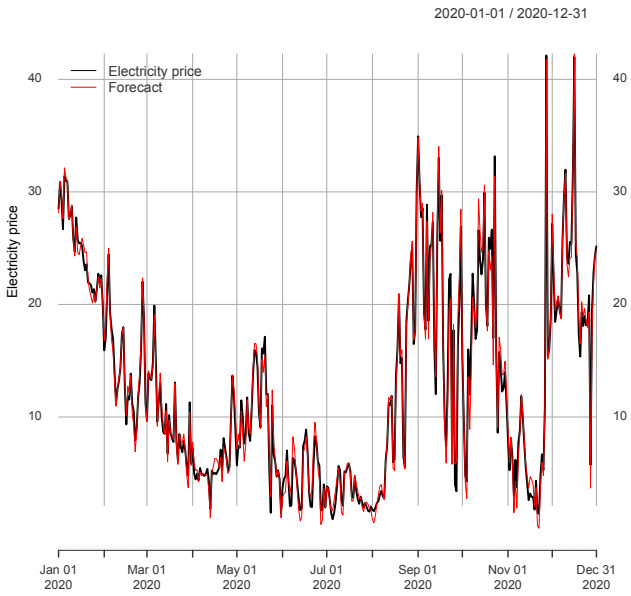

Figure 5.22: Out-of-sample forecast for 2020 with SSA.

MAE	RMSE	MAPE	MSE
0.738	0.953	0.160	0.908

Table 5.23: Out-of-sample forecasting errors with SSA.

CHAPTER 6: COMPARING THE RESULTS

After the analysis of the models, we will proceed with the comparison of them. Firstly, we will compare the evaluation of the forecasts in-sample for 2019 and after that, we will compare the evaluation of the forecasts out-of-sample for 2020. The common forecasting errors we can get from all models are the following: MAE, RMSE, MAPE, and MSE. The key difference between simple error and the absolute error is that absolute error provides a more accurate measure of the magnitude of the error, while simple error only tells you whether the prediction was too high or too low. Therefore, the forecasting errors from the above that we will use to compare the results, are the MAE and MAPE. In Table 6.1 we will present the results of the forecasting errors in-sample, i.e. for 2019, and in Table 6.2 we will present the results of the forecasting errors out-of-sample, i.e. for 2020 using rolling forecasts.

6.1 Comparison of the in-sample forecasts

	MAE	RMSE	MAPE	MSE
SLR	2.012	2.819	0.050	7.951
ARIMA	2.707	3.822	0.069	14.609
SARIMA	2.252	3.347	0.056	11.203
GARCH	1.564	2.423	0.038	5.872
SSA	**0.7392**	**0.954**	**0.019**	**0.911**

Table 6.1: Results of the in-sample forecasting errors.

We noticed that the best model for the estimations in-sample is SSA, which has the smallest values in all evaluation errors, and after that is GARCH. In the optimal model, the MAE which is calculated as the mean absolute difference between actual and estimated values and is relatively insensitive to outliers equals 0.7392 which is very small. Also, the RMSE which is calculated as the root mean square error between the predicted and actual values equals 0.954.

The MAPE which is calculated as the mean absolute error between predicted and actual values divided by the actual value multiplied by 100 equals 0.019. Finally, the MSE which is calculated as the average of the squared differences between the actual and predicted values equals 0.911.

6.2 Comparison of the out-of-sample forecasts

	MAE	RMSE	MAPE	MSE
SLR	1.976	2.770	0.553	7.677
ARIMA	2.995	4.529	0.338	20.512
SARIMA	2.979	4.516	0.331	20.402
GARCH	2.455	3.861	0.261	14.912
SSA	**0.738**	**0.953**	**0.160**	**0.908**

Table 6.2: Results of the out-of-sample forecasting errors.

We noticed that the best model for forecasting out-of-sample is the SSA, as in the case of forecasting in-sample, and after that is SLR (different result from the in-sample forecasts). In the optimal model, the MAE equals 0.738 which has again a small value so the actual data have no significant difference from the estimated. Also, the RMSE equals 0.953, the MAPE equals 0.160 and the MSE equals 0.908.

SUMMARY

The master thesis aimed to evaluate and compare the predictions about the price of electricity in the Nord Pool market using both parametric and non-parametric models. The analysis included the forecast in-sample and out-of-sample electricity prices using time series data from January 2019 to December 2020 and focusing on the peak hour of each day, which was 10:00 AM. Before the application of the models, we checked if there is stationarity in our time series, and since the result was negative, we ended up with the first difference transformation which made our time series stationary.

In the thesis, we applied various parametric models: SLR, ARIMA, SARIMA, and ARCH/GARCH, and a non-parametric model: SSA. The thesis was structured in a way that we first estimated the electricity price through the above models for the period from January 2019 to December 2019 (in-sample estimation) and then forecasted the electricity price for January 2020 to December 2020 (out-of-sample forecast). For the out-of-sample forecasts for short-term electricity prices for one day-ahead, we used rolling forecasts with a rolling window of one year. Finally, we used the forecasting errors to compare the results and find out the models' predictive ability.

In conclusion, using the forecasting errors from the various models, we found out that the best predictions in and out-of-sample were given by the non-parametric SSA model, where the forecasts were moving together with the actual electricity price data, so we are able to predict very well the electricity price.

APPENDICES

R programming language Commands

#install the packages:

```
install.packages("PerformanceAnalytics")
library(PerformanceAnalytics)
install.packages("quantmod")
library(quantmod)
install.packages("dplyr")
library(dplyr)
install.packages("tidyverse")
library(tidyverse)
install.packages("tseries")
library(tseries)
install.packages("rugarch")
library(rugarch)
install.packages("xts")
library(xts)
install.packages("lubridate")
library(lubridate)
 install.packages("Metrics")
 library(Metrics)
install.packages("forecast")
 library(forecast)
install.packages(“datetime")
library(datetime)
install.packages("Rssa")
library(Rssa)
```

#Read the data:

```
dat<-read.csv(“nordic1.csv”, header=T)
dd10<-subset(dat,(dat[,2]=="10:00:00 AM"))
dd10<-dd10[ , c(1, 3, 4)]
qxts <- xts(dd10[,-1], order.by=as.Date(dd10[,1],"%d/%m/%Y"))
consumption<-qxts[,1]
price<-qxts[,2]
Price<-price[1:731]
Price2019<-price[1:365]
Price2020<-price[366:731]
Consumption<-consumption[1:731]
Consumption2019<-consumption[1:365]
Consumption2020<-consumption[366:731]
logarithmOfPrice<-log(price)
logarithmOfconsumption<-log(consumption)
```

#time series & trasformations:

```
par(mfcol=(c(2,1)))
  plot(Consumption,main="Forecasted consumption for 2019-2020",ylab="Forecasted consumption “,cex = 0.3)
 plot(Price,main="Electricity price for 2019-2020",ylab="Electricity price”,cex = 0.3)
adf.test(Consumption)
adf.test(Price)
kpss.test(Consumption)
kpss.test(Price)
logarithmofconsumption<-log(Consumption)
logarithmofprice<-log(Price)
 par(mfcol=(c(2,1)))
  plot(logarithmofconsumption, main=“Logarithm of forecasted consumption for 2019-2020",ylab="Logarithm of forecasted consumption “,cex = 0.3)
```

```
plot(logarithmofprice,main="Logarithm of electricity price for 2019-2020",ylab="Logarithm of electricity price",cex = 0.3)
kpss.test(logarithmofconsumption)
kpss.test(logarithmofprice)
Firstdifferenceofprice<-diff(Price,diff=1)
Firstdifferenceofconsumption<-diff(Consumption,diff=1)
kpss.test(Firstdifferenceofprice)
kpss.test(Firstdifferenceofconsumption)
par(mfcol=(c(2,1)))
plot(Firstdifferenceofconsumption, main="First difference of forecasted consumption for 2019-2020",ylab="First difference of forecasted consumption ",cex = 0.3)
plot(Firstdifferenceofprice,main="First difference of electricity price for 2019-2020",ylab="First difference of electricity price",cex = 0.3)
par(mfcol=(c(2,1)))
acf(Firstdifferenceofconsumption[2:731],main="ACF for First difference predicted consumption for 2019-2020",ylab="ACF First difference of predicted consumption",cex = 0.3)
acf(Firstdifferenceofprice[2:731],main="ACF : First difference of electricity price for 2019-2020",ylab="ACF First difference of electricity price",cex = 0.3)
```

 # SLR model:

#Estimation period with SLR:

```
slr<-lm(formula = Firstdifferenceofprice[2:365] ~ Firstdifferenceofconsumption[2:365], data = dat)
summary(slr)
slr_fit= Firstdifferenceofprice[2:365]-residuals(slr)
mycolors<-c("black","red")
x<-(Price2019)[-365]
plot(Price2019[2:365],main="In-sample forecast",ylab="Electricty price")
lines(x+ slr_fit,col="red")
addLegend("topleft", legend.names = c("Electricity price", "Estimation"), lty = 1, col = myColors)
```

```
AIC(slr)
BIC(slr)
mae(Price2019[2:365],x+ slr_fit[1:364])
mape(Price2019[2:365],x+ slr_fit[1:364])
mse(Price2019[2:365],x+ slr_fit[1:364])
rmse(Price2019[2:365],x+ slr_fit[1:364])
acf(slr$residuals, lag.max=10)
pacf(slr$residuals, lag.max=10)
Box.test(slr$residuals, lag=1, type="Ljung-Box")
Box.test(slr$residuals, lag=2, type="Ljung-Box")
….
Box.test(slr$residuals, lag=10, type="Ljung-Box")
par(mfcol=(c(2,1)))
acf(slr$residuals, lag.max=10,main="Residuals from the SLR")
pacf(slr$residuals, lag.max=10,main="Residuals from the SLR”)
```

#Rolling forecast with SLR:

```
fslr <- c()
for (i in 1:366) {   training_price <- as.numeric (Firstdifferenceofprice[(1+i):(364+i)])
  training_consumption <- as.numeric(Firstdifferenceofconsumption[(1+i):( 364+i)])
  refit <- lm(formula = training_price ~ training_consumption)
 forecast <- predict(refit, h=1,newdata = data.frame(training_consumption =fslr[1:366]))
 fslr <- c(fslr, forecast) }
fslr<-fslr[2:length(fslr)]
 error<-Firstdifferenceofprice[366:length(Firstdifferenceofprice)]-fslr[1:366]
predictedslr=Firstdifferenceofprice-error
x1<-(price[365:731])[-365]
plot(price[366:731],main="Out-of-sample forecast",ylab="Electricity price")
lines(x1+ predictedslr,col=“red")
addLegend("topleft", legend.names = c("Electricity price", "Forecast"), lty = 1, col =
myColors)
mae(x1+ predictedslr,price[366:731])
```

```
mape(x1+ predictedslr,price[366:731])
rmse(x1+ predictedslr,price[366:731])
mse(x1+ predictedslr,price[366:731])
```

 # ARIMA model:

#Estimation period with ARIMA:

```
ARIMA<-auto.arima(Price2019,trace=TRUE)
RESIDUALSARIMA<-residuals(ARIMA)
residualsARIMA2019<-RESIDUALSARIMA[1:365]
ARIMAFIT<-Price[1:365]-residualsARIMA2019
plot(Price[1:365],main="Electricity price for estimate period 2019",ylab="electricity price")
points(ARIMAFIT,col="red",lty=2)
summary(ARIMA)
acf(residualsARIMA2019)
pacf(residualsARIMA2019)
Box.test(residualsARIMA2019, lag=1, type="Ljung-Box")
Box.test(residualsARIMA2019, lag=2, type="Ljung-Box")
….
 Box.test(residualsARIMA2019, lag=10, type="Ljung-Box")
mae(Price[1:365],ARIMAFIT[1:365])
rmse(Price[1:365],ARIMAFIT[1:365])
mse(Price[1:365],ARIMAFIT[1:365])
mape(Price[1:365],ARIMAFIT[1:365])
```

#Rolling forecast with ARIMA:

```
w_size = 365
n_windows =366  #total data - year 19
f1<-0
for (i in 1:n_windows) {training<-Price[(i):(w_size+i-1)]
```

```
refit <- Arima(training, order=c(1,1,2), method="ML")
f1<-cbind(f1,as.numeric(forecast(refit, h=1)$mean))}
f1<-f1[2:length(f1)]
ferror<-Price[366:731]-f1
forecastprice=Price[366:731]-ferror
myColors <- c("black","red")
plot(x = Price, xlab = "Time", ylab = "electricity price",main = "Rolling forecast with ARIMA(1, 1, 2) for 2020 ", major.ticks= "months", minor.ticks = TRUE,col="black")
lines(x = forecastprice, col = "red")
addLegend("topleft", legend.names = c("electricity price", "forecast"), lty = 1, col = myColors)
mae(Price[366:731],forecastprice)
mape(Price[366:731],forecastprice)
mse(Price[366:731],forecastprice)
rmse(Price[366:731],forecastprice)
```

 # SARIMA model:

#Estimation period with SARIMA:

```
Price7<-ts(Price2019,frequency=7)
ARIMAseasonal<-auto.arima(Price7,trace=TRUE,seasonal=TRUE)
residualsseasonal2019<-residuals(ARIMAseasonal)
ARIMAseasonalFIT<-Price7-residualsseasonal2019
plot(Price[1:365],main="Electricity price for estimate period 2019",ylab="electricity price")
points(ARIMAseasonalFIT,col="red",lty=2)
summary(ARIMAseasonal)
acf(residualsseasonal2019)
pacf(residualsseasonal2019)
Box.test(residualsseasonal2019, lag=1, type="Ljung-Box")
Box.test(residualsseasonal2019, lag=2, type="Ljung-Box")
….
Box.test(residualsseasonal2019, lag=10, type="Ljung-Box")
```

```
mae(Price[1:365],ARIMAseasonalFIT[1:365])
mape(Price[1:365],ARIMAseasonalFIT[1:365])
mse(Price[1:365],ARIMAseasonalFIT[1:365])
rmse(Price[1:365],ARIMAseasonalFIT[1:365])
```

#Rolling forecast with SARIMA:

```
w_size = 365
n_windows =366  #total data - year 19
f1<-0
for (i in 1:n_windows) {training<-Price[(i):(w_size+i-1)]
r e fi t < -
Arima(training,order=c(4,1,1),seasonal=c(2,0,0),include.drift=FALSE,method="ML")
f1<-cbind(f1,as.numeric(forecast(refit,h=1)$mean))}
length(f1)
f1<-f1[2:length(f1)]
ferror<-Price[366:731]-f1
forecastprice=Price[366:731]-ferror
myColors <- c("black","red")
plot(x = Price, xlab = "Time", ylab = "electricity price",main = "Rolling forecast with seasonal Arima for 2020", major.ticks= "months", minor.ticks = TRUE,col="black")
lines(x = forecastprice, col = "red")
addLegend("topleft", legend.names = c("electricity price", "forecast"), lty = 1, col = myColors)
mae(Price[366:731],forecastprice)
 mape(Price[366:731],forecastprice)
 mase(Price[366:731],forecastprice)
 rmse(Price[366:731],forecastprice)
mse(Price[366:731],forecastprice)
```

 # ARCH/GARCH

#Estimation period with ARCH/GARCH:

```
arch.test(residualsseasonal2019)
 squared.res.arima1<-residualsseasonal2019^2
ts_data <- ts(squared.res.arima1, start = c(2019,1,1), frequency = 365)
 par(mfcol=c(3,1));plot(ts_data,main="Squared residuals of SARIMA
residuals",ylab="Squared residuals");acf.squared.res.arima=acf(squared.res.arima,main="ACF of
squared
residuals",lag.max=20,ylim=c(-0.5,1));pacf.squared.res.arima=pacf(squared.res.arima,main="PA
CF of squared residuals",lag.max=20,ylim=c(-0.5,1))
GARCH03<-garch(residualsseasonal2019,order=c(0,3)) ;AIC(GARCH03)
summary(GARCH03)
ht.arch03=GARCH03$fit[,1]^2
 loglik03=logLik(GARCH03)
fit03<-fitted.values(ARIMAseasonal)
archres<-residualsseasonal2019/sqrt(ht.arch03)
fit032019<-fit03[1:365]
fit03PLUS<-fit032019+ archres
 ts.plot((Price[4:365]),ylab="Electricity price")
lines(x = fit03PLUS[4:365], col = "red")
qqnorm(archres)
qqline(archres)
```

#Rolling forecast with ARCH/GARCH:

```
w_size = 365
n_windows =366
 f1<-0
m<-0
for (i in 1:n_windows) { training<-Price[(i):(w_size+i-1)]
 refit<-
Arima(training,order=c(4,1,1),seasonal=c(2,0,0),include.drift=FALSE,method="ML")
 f1<-cbind(f1,as.numeric(forecast(refit, h=1)$mean))
m<-cbind(m,residuals(refit)) }
residuals(refit)
```

```
resref<-residuals(refit)
f1<-f1[2:length(f1)]
ferror<-Price[366:731]-f1
forecastprice=Price[366:731]-ferror
m1<-m[2:length(m)]
 squared.reSARIMA<-resref^2
GARCHFORECAST<-garch(resref,order=c(0,3))
summary(GARCHFORECAST)
ht.arch03f=GARCHFORECAST$fit[,1]^2
fit03f<-fitted.values(refit)
garchresf<-resref/sqrt(ht.arch03f)
fit03final<-forecastprice+ garchresf
plot(fit03final)
 lines(Price[369:731],col="red")
mae(Price[366:731],forecastprice)
mape(Price[366:731],forecastprice)
mse(Price[366:731],forecastprice)
rmse(Price[366:731],forecastprice)
```

#SSA

#Estimation period with SSA:

```
Price7<-ts(Price,frequency=7)
s1<-ssa(Price7[1:365],L=182)
plot(s1,type="vectors",groups=as.list(1:8))
 plot(s1,type="series",groups=as.list(1:8))
plot(s1,type="series",groups=as.list(1:182))
ws1<-wcor(s1,groups=as.list(1:182))
plot(ws1)
ws1_<-wcor(s1,groups=as.list(1:50))
plot(ws1_)
res1<-reconstruct(s1,groups=list(1:22))
```

```
trend<-res1$F1
 res.trend<-residuals(res1)
 spec.pgram(res.trend,detrend=FALSE,log="no")
plot(res1$trend)
plot(res1,add.residuals=FALSE,plot.type="single",col=c("black","red"),lwd=c(1,2))
s2<-ssa(res.trend,L=182)
plot(s2)
 plot(s2,type="paired",idx=1:10,plot.contrib=FALSE)
plot(s2,type="series",groups=as.list(1:182))
w<-wcor(s2,groups=as.list(1:50))
plot(w)
w1<-wcor(s2,groups=as.list(1:182))
plot(w1)
res2<-reconstruct(s2,groups=list(1:8))
 seasonality<-res2$F1
  res<-residuals(res2)
 plot(res2,add.residuals=FALSE)
summary(s2)
plot(Price7[1:365])
  lines(Price7[1:365]-seasonality,type="l",col="red")
mae(Price7[1:365],Price7[1:365]-seasonality)
 mape(Price7[1:365],Price7[1:365]-seasonality)
 mse(Price7[1:365],Price7[1:365]-seasonality)
rmse(Price7[1:365],Price7[1:365]-seasonality)
```

#Rolling forecast with SSA:

```
w_size = 365
n_windows =366  #total data - year 19
f1<-0
m<-0
for (i in 1:n_windows) {training<-Price[(i):(w_size+i-1)]
res1<-reconstruct(s1,groups=list(1:22))
```

```
 res.trend<-residuals(res1)
s2<-ssa(res.trend,L=182)
res2<-reconstruct(s2,groups=list(1:8))
 seasonality<-res2$F1}
 plot(price[366:731],main="Out-of-sample forecast”,ylab=“Electricity price”)
lines(price[366:731]-seasonality ,col=“red")
mae(price[366:731]-seasonality,price[366:731])
mape(price[366:731]-seasonality,price[366:731])
rmse(price[366:731]-seasonality,price[366:731])
mse(price[366:731]-seasonality,price[366:731])
```

REFERENCES

Carlo Fezzi & Luca Mosetti, (2018). "Size matters: Estimation sample length and electricity price forecasting accuracy," DEM Working Papers 2018/10, Department of Economics and Management.

R. Weron (2014). “Electricity price forecasting: A review of the state-of-the-art with a look into the future,” International Journal of Forecasting, 30(4): 1030-1081.

Arash Miranian, Majid Abdollahzade, Hossein Hassani (2013),” Day-ahead electricity price analysis and forecasting by singular spectrum analysis,” IET Generation, Transmission and Distribution 7(4):337-346

Nader Alharbi and Hossein Hassani (2016). “A new approach for selecting the number of the eigenvalues in singular spectrum analysis,” Journal of the Franklin Institute, 353(1): 1-16.

Tim Bollerslev, Robert F. Engle and Daniel B. Nelson (1994). “ARCH models,” In Handbook of Econometrics, Vol IV, Robert F. Engle and Dan McFadden (eds.), Elsevier, Amsterdam.

Shahidehpour, M., Yamin, H., & Li, Z. (2002). Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Management. John Wiley & Sons, Inc. https://doi.org/10.1002/047122412X.ch2

Hyndman, R. J., & Athanasopoulos, G. (2014). “Forecasting: principles and practice,” OTexts.

Zareipour, H. R. (2012). Price-based Energy Management in Competitive Electricity Markets: Price Forecasting and Optimal Operation of Wholesale Customers. AV Akademikerverlag.

James D Hamilton (1994). “Time Series Analysis,” Princeton University Press, Princeton,

NJ.

Weron, R. (2006). Modeling and Forecasting Electricity Prices. John Wiley & Sons Ltd. https://doi.org/10.1002/9781118673362.ch4

Nord Pool data portal (2022). Day-ahead electricity price data. https://www.nordpoolgroup.com/en/trading/Operational-Message-List/2022/01/new-data-portal-for-2022-20220114124600/.