Norwegian School of Economics
Bergen, Spring 2019

Empirical Comparison of Time Series Forecasting Strategies

Forecasting The Baltic P1A Spot Price Using Gradient Boosting and Different Strategies for Multi-Step Time Series

Joachim Aae and Anders Dovran

Supervisor: Jonas Andersson

Master thesis, Economics and Business Administration

Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

Acknowledgements

First and foremost, we would like to thank Jonas Andersson for excellent guidance throughout the period we have worked on this thesis. We would also like to thank Peter Lindström and his team at Torvald Klaveness for supplying us with valuable data for our analysis, and knowledge on the dry bulk shipping market. In addition, we would like to thank The Student Association at the Norwegian School of Economics (NHHS) for lending us extra server capacity, and The Norwegian Shippers Association's Fund at NHH for helping us fund this thesis. Last, we want to express our sincere gratitude to our fellow students who took the time to proofread our thesis.

Norwegian School of Economics

Bergen, June 2019

Abstract

This NHH master thesis researches methodologies for forecasting a financial time series, the *Baltic Dry P1A* spot price, one week and one month ahead. The methods researched are four different strategies for time series prediction. The first is by fitting the future timestep directly based on information about today. The second is a recursive strategy, which iterates a one-step ahead prediction model. Third, a rectify implementation that corrects bias from a recursive model, by training on the residuals. Last, a direct recursive approach that fits each timestep directly with previous predictions as an added variable.

Our research finds that the Direct and Direct Recursive (DirRec) strategy is the most accurate for both long and short forecast horizons. This performance is consistent when testing on an independent test set. We cannot exclude that the Direct Recursive strategy could perform better than Direct, as we get differing results when performing experiments with fewer variables. An important trade-off is that the Direct Recursive strategy is substantially more computationally heavy than the Direct strategy.

Keywords – NHH, Master Thesis, XGBoost, Prediction, Time Series, Dry Bulk, Shipping, Multi-Step Forecasting, Recursive Forecasting, GBM, DirRec

Contents

List of Figures

List of Tables

1 Introduction

The dry bulk shipping market is a volatile market where participants take large risks to earn a profit. Taking disadvantageous positions can give a substantial downside, so shipping companies seek to reduce risk by making informed decisions. One way of gaining an information advantage, is by making forecasts. Although domain knowledge is very important in the shipping industry, a reliable and accurate forecast can provide crucial decision support, reduce risk and increase traders' ability to act in the market (Stopford, 2009).

There are many methods available for time series prediction, both classical and modern. In this thesis we are using the same data set used by industry professionals to predict dry bulk spot prices one month in advance. We will look into the feasibility of successfully predicting prices using this data set, as well as research different strategies for prediction. When developing multi-step forecasts, different methods have unique strengths and weaknesses. Classical approaches are often univariate or with a small set of carefully selected features, while modern machine learning approaches can have a large feature space, but require substantial computational time.

For methodology comparisons to be sensible, we must assume that variables in the data set have predictive power. Whether this is true can be answered through how well our results are in general and across methods. Subsequently, we can see how the different implementations of methods affect the results. As a base for our methods, we use a gradient boosting framework called XGBoost (Chen and Guestrin, 2016). Our main methodology hypothesis is that by combining aspects of classical methods, such as autoregressive behaviors, together with a modern machine learning framework such as XGBoost, the resulting forecasts could be more efficient. We will therefore construct and compare various multi-step implementations of the base model. The comparison will evaluate the predictions across several metrics, RMSE, MAPE and MAE, as well as how accurately the models predict the direction of the price development. In addition, we will address the computational time and complexity of implementing the different methods and strategies.

1.1 Structure

Section 2 starts by giving an introduction to the dry bulk shipping market and the forecasting strategies we research in this thesis. Section 3 takes a closer look at the data set provided to us by the shipping company Torvald Klaveness (hereby reffered to as: Klaveness). Section 4 gives a brief description of our benchmark forecasting methods and a more in-depth explanation of more advanced methods. Section 5 presents and compares the final results from applying the different methods to the data set. Section 6 discusses the probable causes to the performance of the different methods. Last, in Section 7 our concluding remarks are presented.

2 Background

2.1 The Dry Bulk Shipping Market

This section introduces the characteristics of global shipping markets and highlights the areas that are relevant for this thesis. If the reader has extensive knowledge about this market, Section 2.1 can be skipped.

The development of shipping markets has had large impact on the economy and world we live in today. Being both a driver and a beneficiary of globalization, seaborne trade has connected countries enabling for transportation of large volumes of goods. This has further enabled economies of scale and global division of labor, shaping the development of international trade and economies throughout history (Stopford, 2009).

Seaborne freight is often divided into a few major categories, depending on the characteristics of the good. Some of the most important categories are: Tank freight, which carries liquids such as crude oil, oil products and liquefied natural gas (LNG). Container ships, carrying large amounts of containers with the 40-foot (2-TEU) unit being the most common. Roll-on/Roll-off (RORO), carrying cars or other wheeled cargo. Refrigerated ships, with perishables requiring a consistent temperature. Lastly, Dry Bulk carriers move goods such as grain, iron ore or coal in large amounts. This last category will be at the center of our analysis, and we will therefore elaborate on this specific market (Stopford, 2009).

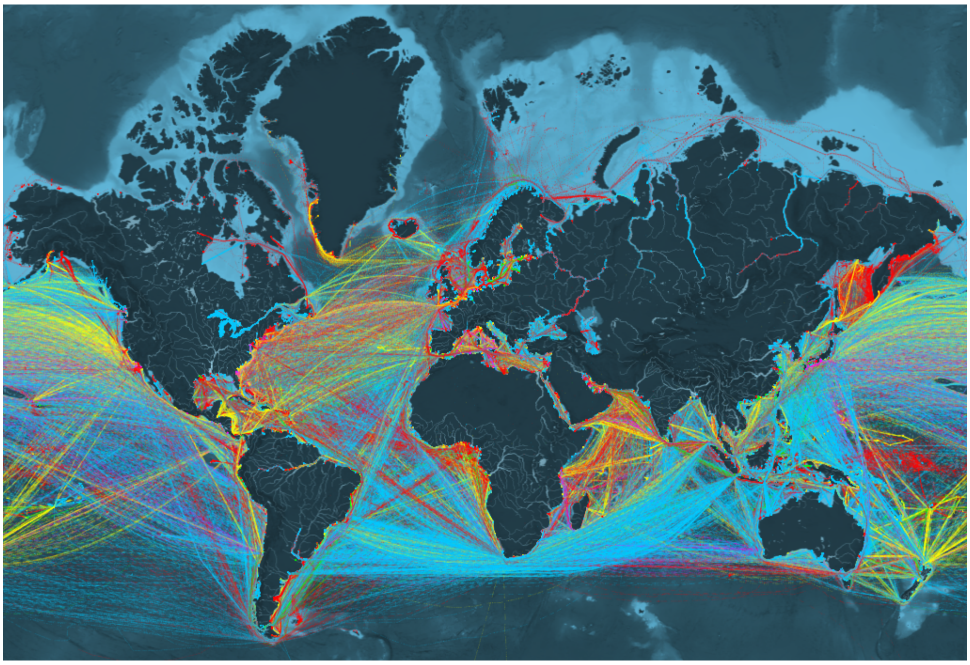


Figure 2.1: Illustration of the world seaborne trade, based on location data from 2012. Dry bulk shipping routes are displayed in light blue. Source: shipmap.org

2.1.1 Characteristics of Dry Bulk Shipping

Dry bulk shipping consists mainly of transporting the three major bulks: iron ore, coal and grain. The United Nations estimates that in 2017, the major dry bulk commodities accounted for 29.9% of the global seaborne shipping trade. The market has been steadily growing, with an annual growth rate of 4.6% between 1980 and 2017 (United Nations, 2018).

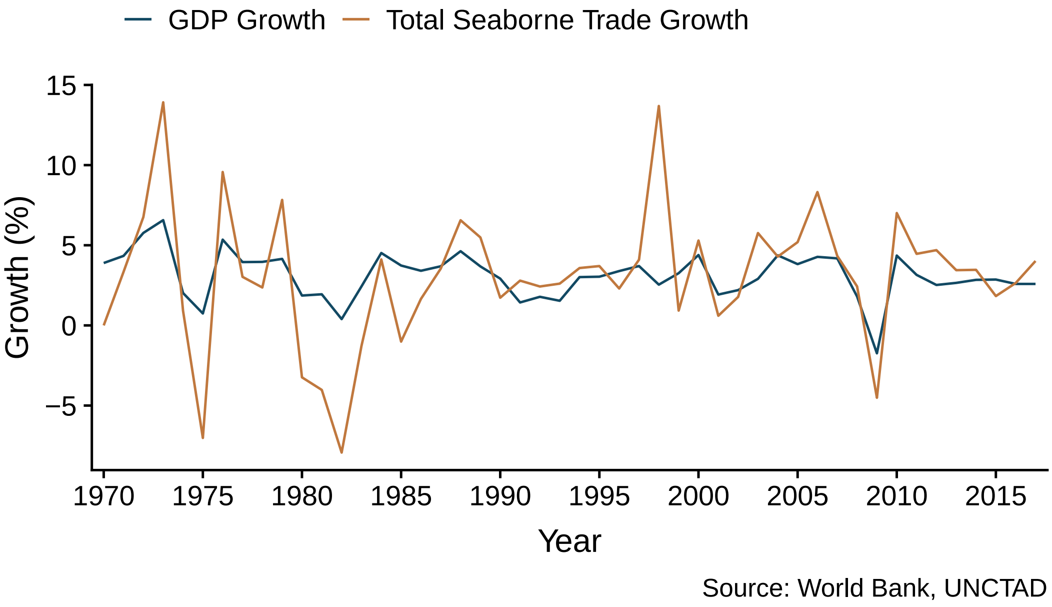
The fleet of dry bulk carriers is divided into groups depending on the capacity of the ship. The four main size segments are, in decreasing order: Capesize, Panamax, Handymax and Handysize, separated by how many Deadweight Tonnes (DWT) they can carry.

In addition to the major bulks, the remaining dry bulk cargo is categorized as minor bulks. Of these, the largest commodities are steel, forest products, metals and minerals. Other large bulk commodities are sugar, fertilizer and *Agribulks*, which is mainly soybeans and soybean meal, but also rice and other oilseeds (Stopford, 2009).

2.1.2 Supply and Demand

As dry bulk shipping is a heterogeneous market, the major exporters and importers of different goods can vary a lot. Compared to e.g. the price for tankers, which will be greatly affected by the price of crude oil, dry bulk is affected by several independent markets each affecting the supply and demand. In general, the movement of goods is determined by the simple questions: Where are things produced? And where are things consumed?

The demand side is heavily dependent on the world economy. A thriving world economy boosts consumption and investment, which increase the demand for production goods such as crude oil, oil products and coal. In Figure 2.2 one can see that the changes in GDP growth correlates with the growth of seaborne trade. With some knowledge of history, major events such as the 1973 oil crisis, 1979 oil crisis and 2008 financial crisis are shown as large negative shocks to the growth of both. Macroeconomic factors surely have a large effect on spot prices over time, but in the scope of predicting a monthly development the effect is probably more long-term than short-term. Seasonality of trades, such as when grains are harvested, has a short-term effect on demand which affects short term pricing. Yearly events like Chinese New Year also affects prices in a short time frame, as Chinese production comes to a halt when the population is on vacation for a few weeks (Stopford, 2009).



Source: World Bank, UNCTAD

Figure 2.2: Development of world GDP growth and the seaborne trade growth, between 1970 and 2017

The supply side of dry bulk shipping is, put simply, set by the amount, size, and behavior of ships in the market (Stopford, 2009). Shipbuilding takes time, and gives a time-lag in the markets' ability to adapt to supply. Some flexibility can be achieved by adapting ships, moving them from one market to another, or by having ships such as combination carriers that can switch between wet and dry cargo. The speed of the ships can also vary, as in a market with low demand and lower prices the shippers would want to reduce costs per trade, while in a market with high demand the shippers increase the speed to complete freights as fast as possible and start the next (Stopford, 2009).

It is important to note that the market is operated by people, and thus, is greatly affected by human behavior. This is an inherent prediction problem, as prices are set based on individual's thoughts and concerns, not by a mathematical equation. Optimistic ship-owners might order too many ships in good times, oversupplying the market when they start arriving. Because of this, the spot price forecast will never be perfect. However, a good forecast will mitigate risk and help decision makers reduce their own behavioral bias (Stopford, 2009).

Another very important factor affecting spot prices are economic shocks and random events. A recent example of this is when a dam collapsed in the Brazilian iron ore mine

The Córrego de Feijão, in the beginning of 2019 (Mandra, 2019). This incident affected the supply for iron ore and therefore the demand for shipping iron ore. Mainly, this affected the market for *Valemax* shipping, which are very large ore carriers specifically used by Vale, the company who owns the mine. These specialty ships are the world's largest bulk carriers, and since the mine closed their capacity opened up to the market, affecting prices of other ship types.

2.1.3 Major Dry Bulk Commodities and Their Markets

Grains

The world's largest exporters of grains are the United States and Russia, with 25% and 23% market share respectively in 2017. This includes wheat, coarse grains and soybeans (United Nations, 2018). Grain is the smallest of the major bulks, but has had steady growth for several decades. A likely reason behind the steady growth, is the increased living standard around the world. With higher incomes, a more bread and meat-based diet is preferred, driving up grain consumption directly but also indirectly as large amounts of the grain trade is animal feed for meat production (Stopford, 2009).

Iron Ore

Iron ore is the largest major bulk, and is mainly used in the worldwide steel production. There are a few countries that dominate the supply and demand side of this commodity. The largest suppliers are Australia and Brazil, at 56% and 26%, while the largest importer is China with a substantial 72% of global demand. Being the largest major bulk, it further exemplifies why events like the Brazilian ore mine collapse and Chinese New Year can affect the global market of all dry bulks (United Nations, 2018).

Coal

The seaborne coal trade is almost as large as iron ore, but has a much more diversified demand. In 2017 the largest importer was China with 18%, closely followed by India, Japan, EU and South Korea, all of which represent 12% of the market (United Nations, 2018). The largest exporters were Indonesia and Australia at 32% and 30%. Coal is divided into two main markets, *coking coal* which is used in steel production, and *thermal coal* which is used to fuel power stations. Coal as an energy source is an alternative to oil and gas, and has historically had growth when oil prices have increased (Stopford, 2009).

2.1.4 Routes and Ships

Shipping is not a fully standardized good, as prices depend on the route, how large the ship is and several other factors. When deciding on market prices, this has to be taken into account, and the result is that there are many indices created from negotiated prices that can be observed. One of the main sources of market information is the Baltic Exchange, which is a global membership organization that gathers information about trades in the market from its members and uses it to create rate indices (Baltic Exchange, 2019a). These indices are one of the main factors when new freight contracts are negotiated.

Panamax

The routes are separated by the earlier mentioned size segments. To limit the scope, we are predicting a Panamax route. Panamax ships are named after being the largest ships that could pass through the *Panama Canal*, with restrictions on width and length. In 2016, the canal opened up a new lane, doubling the capacity and introducing the possibility of larger ships, called Neopanamax to pass through. Regular Panamax ships have a capacity of up to 82,000 DWT, although depth limitation at the canal can limit the capacity to 52,500 DWT.

P1A - Atlantic Round Voyage

In the Panamax size segment, there are four routes which are part of the Baltic Dry Index. These are summarized as the P4TC, which is their average time charter rate equally weighted. To further limit the scope to a single index, we select P1A which is one of these four. The P1A index is rates of transatlantic round voyages between Skaw in northern Europe and Gibraltar in southern Europe, to the US gulf (Baltic Exchange, 2019b).

2.2 Potential Value of Predictions

The spot price market is a physical market, consisting of prices on actual freight contracts being executed. The purely financial alternative is the *Forward Freight Agreement* (FFA) market. For participants in the physical market, Forward Freight Agreements are used as a tool for hedging, for others it opens up purely speculative trades without making physical commitments to ships (Stopford, 2009). Because of this, the FFA market is more liquid and one should expect it to be more efficient. In 2006, the FFA market had an estimated volume of USD 56 billion with more than 300,000 lots traded (Stopford, 2009). The physical market, represented in our case by the spot price, has large barriers to entry and should be less efficient. Therefore, a successful trading strategy for the FFA market would be easier to implement, but less likely to find, while a trading strategy for the spot price market could be easier to discover.

An important decision is for which horizon we are forecasting. Our data set consists of data from days where the market is open, i.e. most weekdays except holidays. Because of this, when building the model, we chose a 5 day horizon as a proxy for a week, and a 22 day horizon as a proxy for a month. The models we are building have no inherent understanding of time, and as it is not the main focus of our research we will not take effort into researching whether 5 or 7 days is the best prediction one week ahead. If one were to implement a trading strategy from our models, the short term prediction should be more accurate in general, but the long term would be easier to act upon as many contracts are negotiated a month in advance.

2.3 Classification or Regression

There are two general approaches to a prediction problem like this, either regressing the price and predicting a future price $\hat{p}_t$, or classifying the direction of where the price will end up. After discussion with Klaveness, an important consideration is the magnitude of the price change for a specified period. Even though a classifier might be more correct in predicting the direction, it is hard to act upon a binary prediction. The magnitude will serve as a proxy for risk, and as such the traders will behave differently if they know the predicted price is a large increase rather than a small increase. The added value of knowing the magnitude is larger than the potential added accuracy of classifying direction, given that the regression models have sufficient accuracy.

2.4 Methods for Time Series Forecasting

Forecasting or predicting future values is an activity that has been done in some shape or form for a long time. The motivation is simple, as predicting the future could be quite lucrative, either by correctly forecasting the costs or income of a project, or as a financial trade by forecasting future stock prices.

Classical time series forecasting has often been tightly connected with economic theories on the topic. For instance, if assuming that a stock price is a random walk because of the Efficient Market Hypothesis, the prediction of tomorrow's price would then be today's price with a random movement (Adland and Alizadeh, 2018). If the price is correlated with earlier day's prices as well, i.e. a trend, a coefficient will be fitted to earlier values of the price. When a time series is correlated with itself, it is autocorrelated.

This autocorrelation can be accounted for by using a model with an autoregressive term, typical for classical forecasting methods. Such models are either univariate, using only the price itself, or use a small set of carefully selected variables to predict. They function by forecasting incremental steps, using the predicted values to forecast the subsequent value. This is called a recursive approach to multi-step forecasting (Taieb et al., 2012a).

In recent years, methods that fall within the term *machine learning* have become popular and are applied to many topics. Machine learning has proven successful in many fields, but is very data dependent and sometimes computationally challenging. Typically, a multivariate approach will be preferred, with a set of related variables to use as predictors x_t. These methods often fit the future value directly as a function of today, thereby fitting each horizon independently of each other.

In this thesis, we seek to experiment with both direct and recursive approaches, and find out which is more suitable for our forecasting problem. In addition, some researchers suggest variants that combine these two. Taieb et al. (2012b) suggest a *Rectified* strategy, where the iterated predictions from the recursive approach is corrected with a model trained on the errors of the base model. If the model is biased, this strategy could improve predictions (Taieb et al., 2012b). Sorjamaa and Lendasse (2006) suggest a *DirRec* strategy, where for each timestep of the Recursive model, a specific model trained to predict $t + step$ timesteps, rather than the original $t + 1$ model ran iteratively. We will consider the implications of these in the methodology section.

2.4.1 Literature Review

There have been many studies comparing these strategies and more. Some researchers that have frequently discussed the topic are Souhaib Ben Taieb and Gianluca Bontempi. The article 'A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition' gives a comparison and summary of applicable methods with many sources if wanted, and cites many articles if one wants to research this topic (Taieb et al., 2012a). Another extensive source on this topic is the research of Rob J. Hyndman, who is also the main author of the popular R package 'forecast' as well as the textbook 'Forecasting: Principles and Practice' (Hyndman and Athanasopoulos, 2018).

3 Data

The source data of this thesis is a panel data set retrieved from the shipping company Torvald Klaveness, one of the leading service providers to the global dry bulk industry (Klaveness, 2019). Klaveness have gathered data from a variety of sources through their research and daily operations, in order to help the company take reliable positions. With Klaveness' knowledge, variables which there are reason to believe affect prices on the dry bulk shipping market have been selected. In this section we take a closer look at the data set, handling of missing observations, the dependent variable and its relation to other variables in the data set.

This data set is currently being used by Klaveness on a day-to-day basis, consequently Klaveness is in danger of exposing valuable data to their competitors if the variables are explained in detail. To some extent, this sets limitations for the level of detail we will go into in this thesis. However, the description of the dry bulk shipping market in Section 2 describes many of the factors that could be important when constructing a data set for this purpose.

3.1 Descriptive Statistics

If you are forecasting a daily time series, it is often difficult to acquire large amounts of relevant data. Both because a whole year of data would at most be 365 observations, but also because most time series change over time. The 1980's shipping market is not the same as today's, and therefore observations from 40 years ago will likely be of little use to a model predicting future prices. The data set provided by Klaveness contains 1772 daily observations between September 2012 and January 2019, across 759 variables. The data set is limited to trading days, thereby excluding many holidays, and weekends. The explanatory variables are from a variety of sources which can be categorized based on their origin. Examples of such categories are: Securities traded on the Baltic Exchange, stock prices, commodity prices, and foreign exchange rates. In addition, the data set contains variables which are made to account for time specific changes.

3.1.1 Dependent Variable

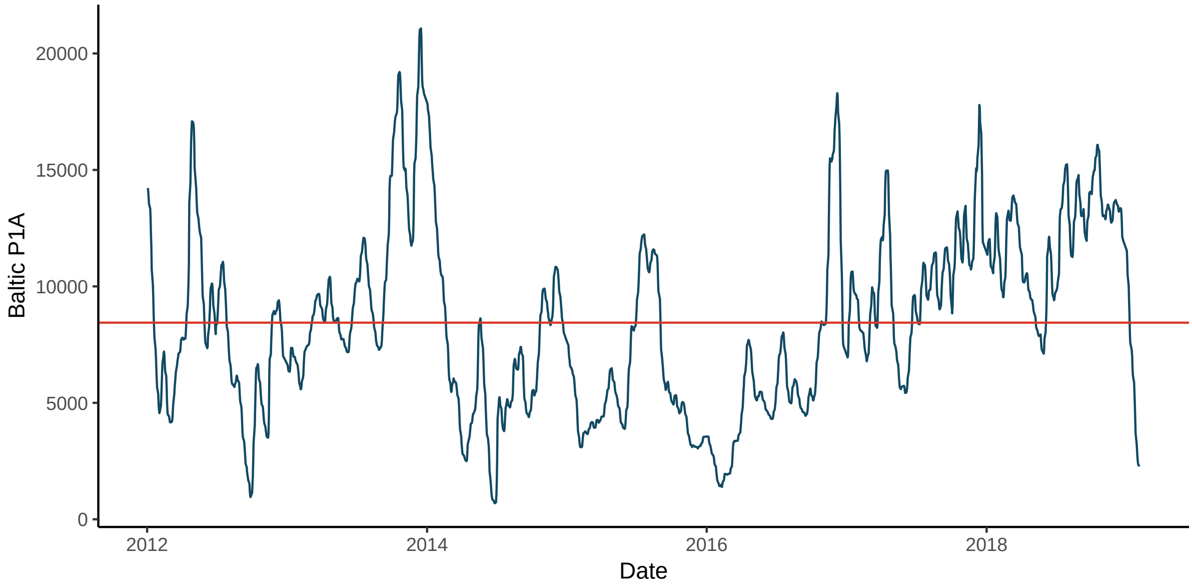


Figure 3.1: Development of the Baltic P1A spot price between 2012 and 2019. The mean is displayed in red.

The dependent variable of the data set is *Baltic P1A*, the spot price index for a Panamax transatlantic round voyage (Explained in Section 2.1.4). Figure 3.1 shows the development of the Baltic P1A price from 2012 until the beginning of 2019. As one can see in Table 3.1 the prices stretches from a minimum value of 690 to a maximum value of 21075. The mean is 8442, not far from the median of 8150, which indicates little skewness.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
690	5346	8150	8442	11018	21075

Table 3.1: Descriptive statistics, Baltic P1A.

The histogram in Figure 3.2 shows how the daily observations of Baltic P1A is bell-shaped with a slight deviation in the number of observations close to 5000. The bell-shape supports the empirical results of Koekebakker et al. (2006) which finds that freight rates have stationary mean reverting characteristics.

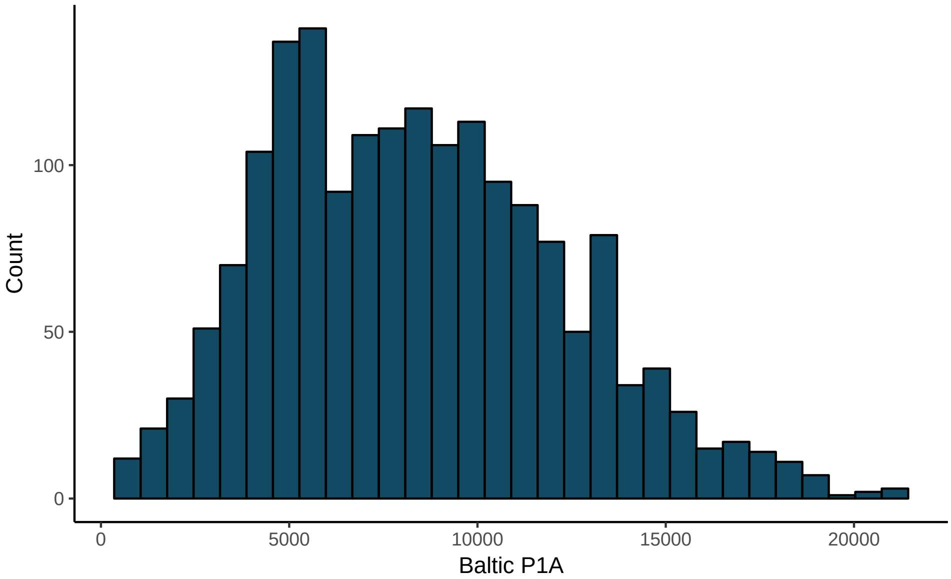


Figure 3.2: Histogram illustrating the distribution of the Baltic P1A spot price.

Stationarity

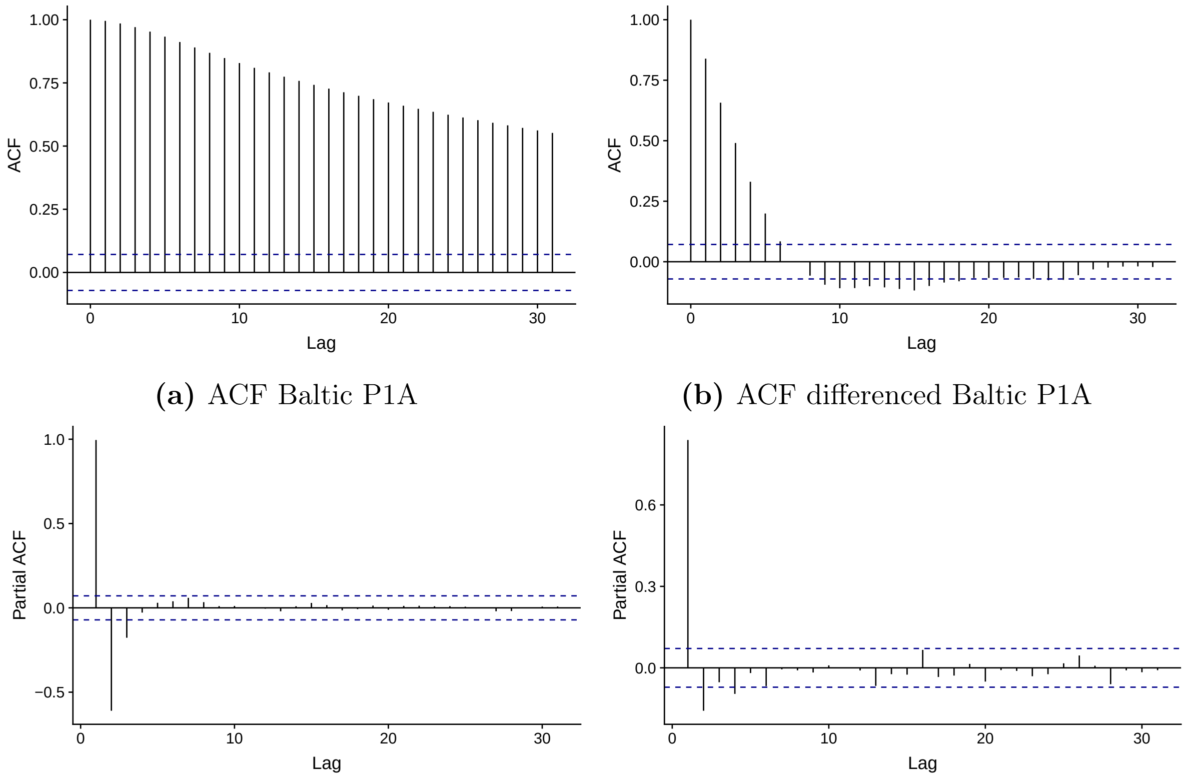
A stationary data set will have all observations independent of the time series in which they were observed, without seasonality or trends (Hyndman and Athanasopoulos, 2018). A non-stationary time series can become more stationary by differencing the values, such that each value is the change from the previous day: $y'_t = y_t - y_{t-1}$. Certain forecasting methods may perform better with a differenced data set, while other methods have built in abilities to account for trends and seasonality e.g. multivariate models with dummy variables accounting for time specific events, seasons or weekdays. Knowledge about the data set in terms of stationary properties can help develop a better forecasting model. Thus, it is relevant to address whether the data set holds such features, before constructing models.

ACF and PACF plots

The autocorrelation functions (ACF) in Figure 3.3 show the correlation between a Baltic P1A and its lags. These plots show that y_t is correlated with y_{t-1}, which also means that y_{t-1} and y_{t-2} must be correlated. This correlation may be the result of both lags being connected with y_t, and not necessarily because y_{t-2} holds any new information. This effect of correlation between lags is neutralized in the *partial* autocorrelation function

(PACF) plot, which shows the correlation adjusted for prior lags. Such plots can be a useful tool to find appropriate number of lags for the autoreggressive and moving average terms in an ARIMA model (Hyndman and Athanasopoulos, 2018).

By inspecting the plots in Figure 3.3, one can see how Baltic P1A shows clear evidence of autocorrelation. Figure 3.3a shows the autocorrelation function for different lags of Baltic P1A, with a slowly decreasing autocorrelation. The effect of differencing is evident when comparing the plots in Figure 3.3a and 3.3b, where the autocorrelation clearly is reduced. The partial correlation of each lag can be found in Figure 3.3c and Figure 3.3d.



(a) ACF Baltic P1A

(b) ACF differenced Baltic P1A

(c) PACF Baltic P1A

(d) PACF differenced Baltic P1A

Figure 3.3: ACF and PACF plots for Baltic P1A, showing the effect of differencing

Unit Root Test

There are a variety of tests available to determine whether Baltic P1A spot price is stationary, e.g. Kwiatkowski–Phillips–Schmidt–Shin (KPSS), Augmented Dickey-Fuller (ADF) and Phillips-Perron (PP). We have chosen to conduct a KPSS test, as it has proven to lead to better ARIMA models on average (Hyndman, 2014). Generally, these tests struggle with size distortion, and can possibly reject the null hypothesis even though it is true (Zivot and Wang, 2003). The null hypothesis (H_0) of the KPSS test is that the data is stationary, and it is rejected for a test statistic higher than the 5 percent critical value of 0.463. The KPPS test with seven lags gives a test statistic of 2.98, indicating rejection of the null hypothesis at a 1% level. This supports the autocorrelation displayed in the ACF plots in Figure 3.3. However, it contradicts Koekebakker et al. (2006) in terms of freight rates having stationary characteristics.

The ACF and PACF plots indicate that the rates should be differenced to become more stationary. Therefore, if the goal of this analysis was inference, we would have likely differenced the data. In our case, we want to achieve as accurate predictions as possible, and whether or not the data should be differenced to facilitate accurate forecasting is still arguable. Therefore, when stationarity is an important concern in the model, we will test out both with and without differencing, and select the best performing model.

3.1.2 Correlation with Baltic P1A

Many of the available predictors are highly correlated with our dependent variable. Table 3.2 shows the ten variables with the highest correlation with the future spot price, on a five-day horizon. We can see that all variables on this list are indices or securities listed at the Baltic Exchange with a positive correlation to the Baltic P1A. Not surprisingly, the highest correlating variable is Baltic P1A itself, indicating autocorrelation on a five-day horizon. Second is the P4TC, which contains 25% of the P1A rate.[1] In addition, the one month FFA for P4TC is present with a correlation of 0.861, illustrating the relationship between futures and spot prices. We also find other routes such as P2A, P5 and Baltic S4A which explain that rates are correlated across different routes and ship types.

Calculation of correlation was also performed on a monthly horizon, showing some other variables among the top ten. These calculations are unfortunately excluded from the thesis, due to the fact that they would expose too much of the data set.

[1] The P4TC rate is explained in more detail in Section 2.1.4

#	Variable	Correlation
1	BALTIC_P1A	0.939
2	BALTIC_P4TC	0.896
3	FFA_P4TC_M0	0.861
4	BALTIC_P2A	0.849
5	BALTIC_Smax_Atlantic_Tess_58	0.796
6	BALTIC_P5	0.777
7	FFA_P4TC_M1	0.776
8	BALTIC_P3A	0.763
9	BALTIC_SM10TC	0.759
10	BALTIC_Ratio_P4TC_S6TC	0.756

Table 3.2: Top ten correlated five-day lagged variables in relation to Baltic P1A

In Figure 3.4 the price development of Baltic P1A and Baltic P2A are illustrated together. The similar development indicates that many of the same underlying factors affect the two prices. This is reasonable considering that P2A is a route from the Skaw-Gibraltar range through the US Gulf to the Taiwan-Japan area, and thereby partially operating the same region. Routes operating in parts of another route's region, shipping the same type of goods, should intuitively be affected by many of the same market factors.

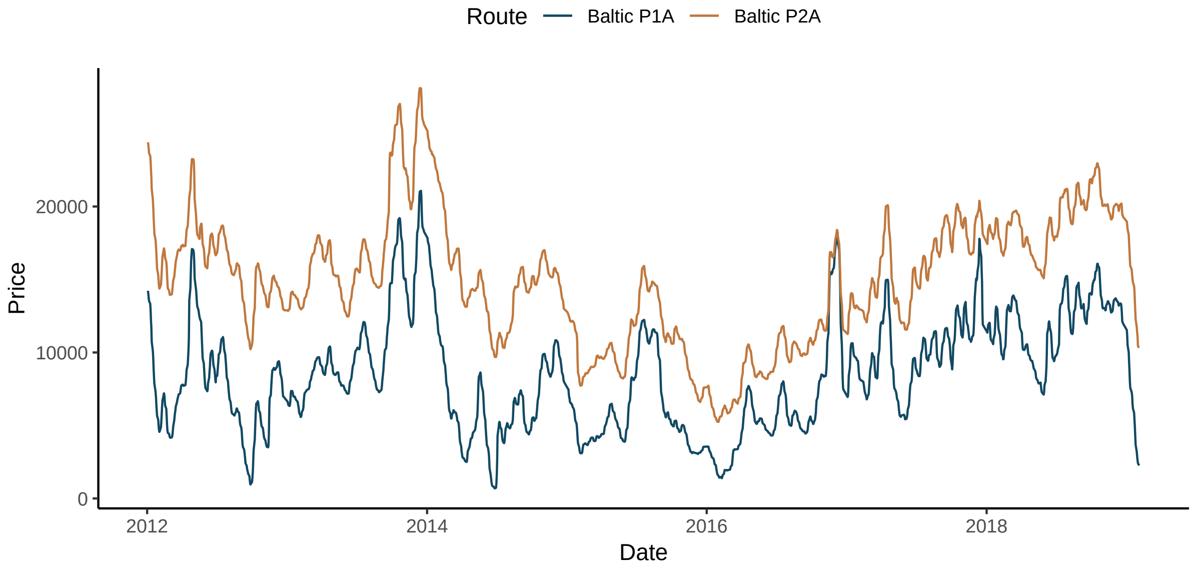


Figure 3.4: Development of Baltic P1A and Baltic P2A between 2012 and 2019

3.2 Missing Values

Some variables in the data set consist of a continuous time series, i.e. a time series without gaps. However, most variables are registered inconsistently, resulting in a total of 18% missing values in the data set. The occurrence of missing values is visualized in Figure 3.5, ordered by most to least occurrences of missing values, from left to right. These missing values are handled in a systematic way, removing columns and then rows, in order to balance the trade-off between data quality and number of observations. Our approach of handling missing values will be explained in more detail in Section 4.1.1.

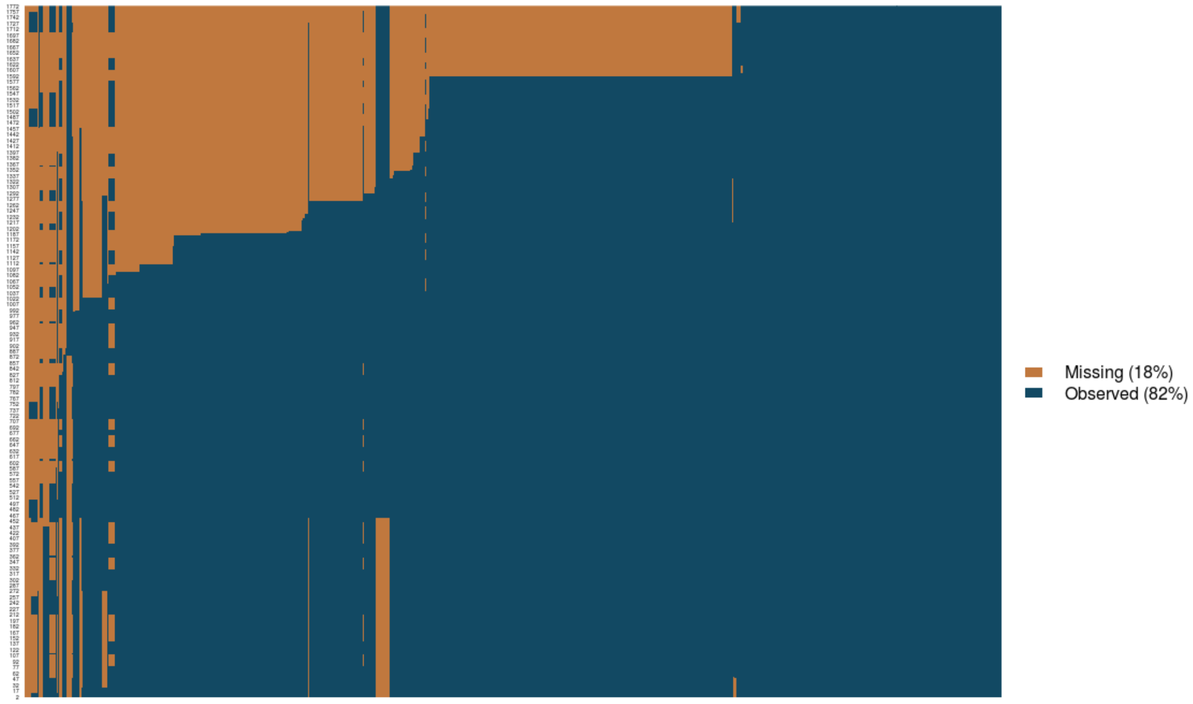


Figure 3.5: Missingness map showing variables on the x-axis, and observations on the y-axis.

4 Methodology

This section focuses on different methods to solve aspects of the forecasting problem. First, the data preparation method is presented. Afterwards, the method of cross-validation used for model selection and assessment of performance. Later, the different benchmark models are introduced. Last, the characteristics of the different strategies are explained.

4.1 Data Preparation

4.1.1 Handling of Missing Values

The objective of handling missing values is to enable the different methods to make use of the data available, and at the same time manage the trade-off between data quality and the number of observations. Handling missing observations can influence the results in different ways depending on the type of data, how it is distributed, the objective, and the model in use. In order to mitigate the issues of having missing values, we have applied an intuitively sensible rule in order to remove missing values in a consistent way. A summary of this approach is explained in Table 4.1, while the following section explains the rationale behind it.

Step	Action	Columns(Δ)	Rows(Δ)
	Status before data preparation	759	1772
1	Remove columns >50% NA	(-32)	
2	Remove row 1-180 (>50% NA)		(-181)
3	Remove columns with recent NA	(-19)	
4	Remove columns with NA in the middle of the data set	(-5)	
5	Remove columns without variation	(-1)	
6	Median impute columns		
	Final data set	702	1591

Table 4.1: Our stepwise approach of handling missing values (NA)

First, our approach is to remove columns with more than 50% missing observations. This threshold was decided after inspecting each decile of missing value percentage, and counting the number of columns within each. This showed that an effective removal of missing values would be at a threshold of 50%. Another reason for choosing 50% as a threshold is because we consider it to be difficult to impute variables with more than half of its values missing, and at the same time ensure a realistic representation of the reality. This process reduced the number of columns from 759 to 727.

Thereafter, row 1 to 180 was removed because they contained more than 50% missing values. These rows represent the oldest part of the data set, and is therefore considered to be less relevant. Next, columns with missing values in the most recent observations were removed. This is because these rows represent data we may not receive in the future. A variable which always has to be imputed when receiving new data, will have limited ability to increase the forecast accuracy of a model.

Certain variables with substantial amount of missing values in the middle of the time series were removed without imputation. In the same way, a variable only containing the value 0 was removed, as it cannot explain any variation.

After removing columns and rows, there were still variables containing missing values. These variables were median imputed, which is the process of replacing missing values with the median value of the specific variable. As it does not make sense to median impute certain types of variables, they were assessed with basis in the type of data and its relationship to the dependent variable. Imputation was performed in a stepwise manner in order to maintain the order of the time series. This ensures that no future information enter the training set through the imputed values. By using this approach, we ensure to maintain a continuous time series, while not adding new characteristics to the data set.

4.1.2 Feature Selection

Methods for high dimensional data usually require some type of feature selection before method implementation. Since the models used in this thesis either are univariate or have built in feature selection, and that a large part of the thesis focuses on comparison of methods, feature selection will not be a focus. Despite this, the variable importance will be addressed in section 5 in order to understand which variables add predictive power to the models.

4.1.3 Aggregation

An important consideration to make is whether to aggregate the data set, which changes the granularity of the data. For instance, if a prediction about next week is wanted, both historical daily data and weekly data could be utilized. There are two main ways of aggregating a data set. One is taking the mean of the observations in the aggregated period, while another is summarizing the change of all the observations. The two different methods will give different meaning to the forecasts. It is therefore important to find the most suitable method for the specific forecast.

Aggregating the data set from daily observations to weekly or monthly can have both positive and negative effects on the forecasts. On the one hand, aggregating will result in less computational time, since the data set will have reduced size. In addition, aggregation can give the models access to data which is registered with a different level of granularity. This can help the forecast make use of data it would not have access to otherwise. On the other hand, aggregating means that one will lose information that might be important to make a precise forecast.

When forecasting dry bulk time charters, aggregation by taking the mean seem to be the most suitable method, as it would both account for variation within the period, as well as smooth out the high variability of daily values. After experimenting with aggregation, we decided not to aggregate, as the test results were significantly worse and the number of observations were reduced to an undesirable level.

4.2 Cross-Validation

Forecasting accuracy can only be determined by evaluating how a model performs on new data, which has not been used for fitting the model (Hyndman and Athanasopoulos, 2018). K-fold cross validation would regularly be a go-to approach for validating machine learning methods. However, when performing cross-validation on time series data, it makes little sense to mix the order of the observations. In addition, standard K-fold cross-validation will suffer from data leakage, mixing future data into the training set (De Prado, 2018). Forecasting the future will essentially be easier with knowledge on the actual future, making the results unrealistically good and causing the model to overfit. To ensure out-of-sample testing and prevent data leakage, we therefore use the time series cross-validation approach.

Time series cross-validation, also known as walk forward validation, is one of the most widely used forms of time series evaluation together with Akaike's Information Criterion (AIC) and Schwarz's Bayesian Information Criterion (BIC). Hyndman and Athanasopoulos (2018) suggest using AIC, corrected AIC or time series cross-validation as they will lead to the same result when a large number of observations are used for estimation. We have selected time series cross-validation of model performance in order to ensure comparable results independent of the number of observations and method used.

4.2.1 Time Series Cross-Validation

Time series cross-validation involves training on all prior observations and validation on a single future observation H steps ahead (Hyndman and Athanasopoulos, 2018). This procedure is then repeated for all future observations. The incremental construction of the training set is illustrated in Figure 4.1, where the blue squares make out the training observations and the red square is the observation in which validation is performed.

The main benefit of the method, is that the tested validation error is always a future value, thereby reducing the possibility of data leakage. The method is still not without drawback. De Prado (2018) asserts that since the walk forward always tests the same scenario, repeated tests such as through parameter optimization can lead to overfitting. Another drawback is that the starting calculations are made on a small part of the data set, and as such, many of the errors will be calculated without sufficient training data.

Training and Test Set

In order to mitigate the drawbacks of time series cross-validation, and to reduce the computational time needed, we have chosen to split the data into an initial training set from observation *t=1* to *t=1200.* The walk-forward is then ran throughout the remaining data, validating on all future observations. This should make the training set sufficiently large for all observations, reducing the possible drawback of walk-forward. The downside is that the overall amount of test data is reduced significantly, making our test estimates less robust. Last, the evaluation metric for the errors is calculated in order to find which model, and with which parameters, that on average results in the lowest forecast error. Figure 4.1 and Equation 4.1 gives a visual and mathematical illustration of our cross-validation set up, respectively.

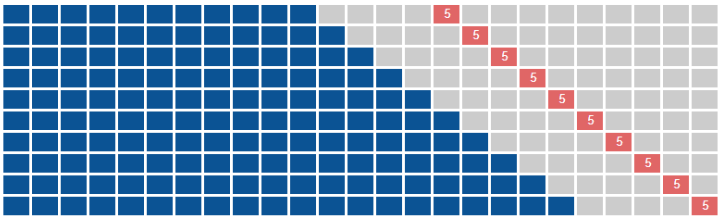


Figure 4.1: Example of cross-validation with validation at $H = t + 5$

$$CV_{(t)} = \frac{1}{T - t} \sum_{t=1200}^{T} (y - \hat{y})_{t+H} \tag{4.1}$$

4.3 Method Evaluation

To evaluate how the models perform, several metrics are available. The models will be evaluated on their forecasting performance, not on their ability to explain the relationship between variables.

4.3.1 Evaluation Metric

Root Mean Square Error

The root mean square error (RMSE) is used as the main evaluation metric in this thesis. For every method, each prediction error for all timesteps is stored. The errors are then squared to make them absolute, since errors in either direction are of same importance. By squaring the errors, RMSE also incorporates a measure of risk, as the value increases exponentially with the error, thereby punishing high variance. Last, the errors are averaged to get the Mean Squared Error (MSE), and then taking the square root to get RMSE. The objective is to find the method and model which corresponds to the lowest RMSE.

The RMSE is a common evaluation metric used for a wide range of problems (Hastie et al., 2013).

Mean Absolute Error

The mean absolute error (MAE) is used as a secondary evaluation metric, after RMSE. MAE works by taking the mean of the absolute value of the error. This metric also measures errors in either direction similar to RMSE, but does not punish high variance as much. In other words, MAE is different from RMSE in that it punishes large errors less (Hastie et al., 2013). The MAE is however easier to interpret as it is directly transferable to the unit of the dependent variable.

Mean Absolute Percentage Error

The mean absolute percentage error (MAPE) calculates the absolute error relative to the actual value. This makes it easy to interpret the magnitude of the errors. Another benefit of using MAPE is that it is unit-free and can therefore be used across data sets (Hyndman and Athanasopoulos, 2018). Although not used for model selection, we list the MAPE because of its interpretability.

Hit Rate

As explained in Section 2 the implications of forecasting the wrong direction of the price is considerable when taking positions in the dry bulk market. It is therefore interesting to see how the method has performed in terms of forecasting the correct direction compared to today's price. As a result, the hit rate (the percentage of correctly forecasted directions) is included in Table 5.1. As mentioned earlier, the hit rate could have been evaluated as a classification problem where the objective is to predict whether the price will go up or down. The importance of direction is addressed, but will not be the main focus of the analysis.

Trading Strategy

Another method of measuring performance is by calculating the returns of a trading strategy using the forecasts. This would be an easily interpretable method, as the net profit would be a relatable metric. There are two reasons we are not using this approach. The main reason is that we do not have available data to calculate the actual profits of many trading strategies. The spot price can be either a revenue or cost component depending on which side of the trade you are on. In both cases, additional information is needed to get a trustworthy estimate of profits. An arbitrage strategy using FFA prices and spot prices could be constructed, but this requires more exploration of the technical financial details of these markets than we deem necessary for this thesis. Another important reason is that compared to RMSE, a *net profit* metric would not be affected by variability and therefore *risk*. Because of these reasons, a trading strategy metric is not utilized.

4.4 Benchmark Models

In order to evaluate and compare the performance of more advanced methods, we have chosen a group of benchmark models differing in complexity. By having suitable benchmarks, it is easier to evaluate the actual gain of using more advanced methods. The gain is evaluated against its complexity, interpretation, and computational feasibility. The benchmarks also function as introductions to some key concepts used in the more advanced models. Next, we introduce the *Random Walk* model, *Moving Average*, and *ARIMA* as benchmark models.

4.4.1 Random Walk Model

A common benchmark in forecasting is to use a random walk model, also called the *naïve method*. The random walk model will predict the last observed value for any future horizon, as shown in Equation 4.2. Random walk models are often used if the data is non-stationary, which is often the case in financial data such as stock prices. The rationale behind the model is that the underlying time series is a *random walk*, i.e. that all future movement is unpredictable but equally likely. This makes the expected value of a future observation the same as the last observed, and if true this model should perform well (Hyndman and Athanasopoulos, 2018).

$$\hat{y}_{t+H} = y_t \tag{4.2}$$

4.4.2 Moving Average Model

Another univariate benchmark is to calculate the moving average in a range of prior observations. On a data set with mean reverting characteristics, a moving average is intuitively a good forecast, depending on the frequency of the mean reversion. The moving average is not always suitable, e.g. in a time series with a constant growth. When it comes to the hit rate, it is intuitive that such a model would be correct about 50% for a normal distributed time series. For this benchmark, the period to average over needs to be selected. Our selection is somewhat arbitrary, with a 100-day period, made mainly to demonstrate the concept of a moving average model.

$$\hat{y}_{t+H} = \frac{1}{100} \sum_{i=t-100}^{t} y_i \tag{4.3}$$

4.4.3 The ARIMA Framework

The Autoregressive Integrated Moving Average (ARIMA) model holds the characteristics of both the moving average and random walk model. Deriving from the Box–Jenkins methodology, the univariate ARIMA model is commonly used in time series analysis. ARIMA consists of three components: Autoregressive (AR), Integrated (I), and Moving average (MA). This entails that ARIMA accounts for effects of prior values of the dependent variable, as well as the element of differencing. As ARIMA is a simple, but yet a powerful forecasting tool, the model is well suited as a benchmark model.

$$y'_t = c + \phi_1 y'_t + ... + \phi_p y'_{t-p} + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} + \epsilon_t \tag{4.4}$$

Equation 4.4 shows the integrated, autoregressive and moving average term of the ARIMA(p, d, q) model, for different values of p (order of the autoregressive term), d (degree of differencing) and q (order of the moving average term) (Hyndman and Athanasopoulos, 2018).

Model Selection

The optimal ARIMA model is selected through a combination of visual inspection of the ACF and PACF plots described in Section 3 and cross-validation of different p, d, and q models.

First, the visual determination of p and q is performed by identifying lags which corresponds to the positive significant spikes in the plot, before it cuts off or turns negative. The last lag of significant spikes in the partial autocorrelation function plot is used to identify the number of AR terms included in the model. In the same way, the number of MA terms is identified by looking at significant lags in the autocorrelation plot. Last, a range of values for p, d and q, in reasonable proximity to the visual determination, is tested with cross-validation in order to find the best performing ARIMA model.

4.5 Machine Learning Framework

4.5.1 Tree-based Methods

A simple method that can be used both for regression and classification is *decision trees*. The method works by splitting the predictor space into regions, and attributing a prediction to each region. Depending on the use case, the method can be called *regression trees* or *classification trees* as well. Decision trees are highly interpretable and intuitive to understand, and are named for the visual rendition of them which resembles a tree structure, as seen in Figure 4.2 (James et al., 2013).

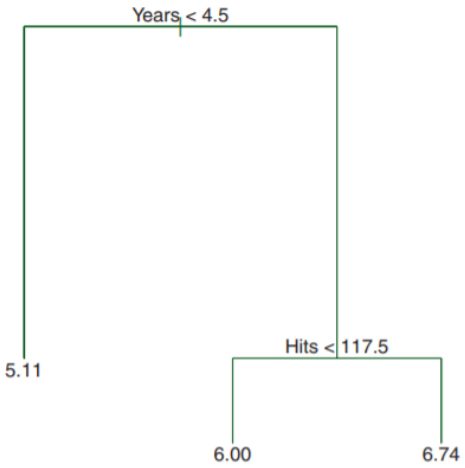


Figure 4.2: Illustration of a simple regression tree. Source: (James et al., 2013)

Standard decision trees often perform worse compared to other methods, and is therefore rarely used if prediction accuracy is the goal. Still, the method can be used as a base for three common models: *Bagging*, *Boosting* and *Random Forest*. All of these are *ensemble models*, which instead of relying on a single decision tree create several slightly varying models, and use the combination of them to output a final predicted value. In a regression setting, this means averaging the predictions of the models. In a classification setting this means taking a majority vote of which class the models predicted. Ensembles of decision trees can create highly accurate models, although at the cost of interpretability (James et al., 2013).

The main difference between the three variants, is how the additive models are fitted. Bagging uses random sampling with replacement, also known as *bootstrap sampling*. The observations are sampled when each model is fit, to reduce variance of the models and improve ensemble accuracy. Random forest instead samples the predictors, and uses a random subset of predictors for each fit, decorrelating the models and improving ensemble performance. In our thesis, the boosting method is used extensively and therefore requires a more in-depth description in the sections below (James et al., 2013).

4.5.2 Boosted Trees

Boosting is a sequential implementation of prediction models, which fits multiple models on previous models' residuals, to produce the final ensemble model for prediction. The fitting sequence initially fits a selected model that minimizes loss, for instance a regression tree. The residuals of this fit, is then used to fit a new model f_m that best explains the remaining variance. To prevent overfitting, the models could also be shrunk by a shrinkage parameter η. The process is described simplified in the algorithm below. Usage of boosting together with tree-based models is often called *Boosted Trees* (James et al., 2013; Hastie et al., 2013).

Boosting process for Regression Trees

```
Set $\hat{f}(x) = 0$ and residuals $r_i = y_i$ for each observation i in training set.
for all $m = 1, 2, ....M$ do
    Fit model $\hat{f}_m(x)$ to data $(X, r)$
    Update model $\hat{f}(x) \leftarrow \hat{f}(x) + \eta \hat{f}_m(x)$
    Update residuals $r_i \leftarrow r_i - \eta \hat{f}_m(x, r)$
end for
```

This process is repeated a desired M times. Giving a final model

$$\hat{f}(x) = \sum_{m=1}^{M} \hat{f}_m(x)$$

XGboost

There are many ways to implement *Boosted Trees*, and they have some distinct differences. XGBoost has proven to be a highly effective method in many machine learning cases, demonstrated by the amount of *Kaggle* competitions won by using this algorithm.[2] A 2016 NTNU master thesis researched why the algorithm performs so well, concluding that tree boosting is highly adaptable by using differing levels of flexibility for different areas of feature space (Nielsen, 2016). This implies that the model takes into account the bias-variance trade-off, fitting simpler representations where the relationships between variables are simpler, but still being able to fit complex interactions and functions where they are occurring. This also weighs features, providing an implicit feature selection in the algorithm (Nielsen, 2016).

To highlight the main differences of XGBoost, we compare it to the common

[2] *Kaggle* is an online competition platform for Data Science, and XGBoost is used extensively by top-performing teams (Chen and Guestrin, 2016).

implementation of 'Gradient Boosted Machine' suggested by Friedman (2001), used in the R *gbm* package. The original author of XGBoost, Tianqi Chen, answers this question himself by stating that:

> "Both xgboost and gbm follows the principle of gradient boosting. There are however, the difference in modeling details. Specifically, xgboost used a more regularized model formalization to control over-fitting, which gives it better performance." (Chen, 2018).

This means that the difference lies in how the model itself is fit, the general boosting framework of algorithm 4.5.2 is still used. *gbm* offers regularization through the shrinkage parameter or learning rate η, the depth of each tree which decides how many variables each tree will split on growing depth-wise, and a bagging fraction which combines bagging and boosting. XGBoost adds many parameters to this list, but some of the most important ones are: The L1 regularization term *alpha*, which adds a shrinkage parameter similar to *lasso regression*, punishing the size of the variable weights. The L2 regularization term *lambda*, similar to *ridge regression* which punishes the squared size of variables weights. The *gamma* is the minimum loss required for a tree to split. Last, the column subsampling fraction enables the fit to behave more similar to a random forest regression, where a specific share of the predictors are selected randomly for each iteration (Chen and Guestrin, 2016; Nielsen, 2016).

By utilizing column sampling, the algorithm will select more varying predictors to explain the dependent variables, aiding the implicit feature selection. In our case, with a feature rich data set, having an algorithm that automatically selects and weighs features is highly practical. The simplicity of implementing XGBoost is also beneficial to test out a controlled experiment on creating autoregressive implementations of regression models. XGBoost is also sparsity aware, which means that it can work around missing values in the data set. The algorithm learns which direction that minimizes loss for each parameter, which allows for further growth of the trees even though a missing value has been encountered. Since we remove the missing values beforehand, this feature is not used, but it can be very practical if a quick experimental test run is needed (Chen and Guestrin, 2016).

Parameter Optimization

The mentioned regularization parameters can dramatically change the performance of the model, and must be tuned and selected to perform optimally. A thorough way of doing this is through using a parameter grid, where for n combinations of p parameters the $n * p$ possible models are tested. Naturally, this takes a long time if the grids are large. A common approach is tuning a few parameters at a time beginning at the parameters that should have the largest impact on the model. We follow this approach, by at first tuning the number of boosting rounds, learning rate and depth of the models. The number of boosting iterations and the learning rate have a high degree of interaction, and therefore should be trained jointly (Nielsen, 2016). After these main variables are predicted, row-wise and column-wise sampling parameters are tested, as well as regularization parameters lambda (L2 regularization) and gamma (split regularization) (XGBoost Documentation, 2019).

4.6 Strategies for Multi-Step Time Series Forecasting

This section introduces the four strategies for time series forecasting studied in this thesis. With the exception of the Direct strategy, all strategies contain different activities between t and $t + H$ which have the intention of increasing the forecasting accuracy of $\hat{y}_{t+H}$, illustrated by the yellow areas in Figure 4.3.

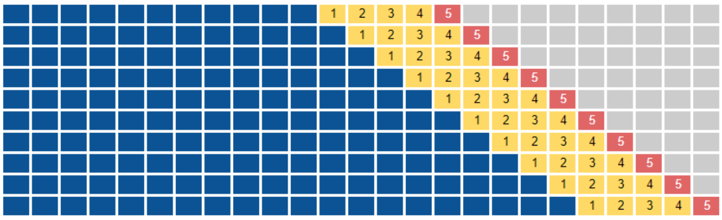


Figure 4.3: Example of model expansion in step 1-4, on a five-day horizon

Direct Approach

As mentioned in Section 2, there are several ways of expanding a Boosted Trees model to have autoregressive components. In our Direct approach, this is achieved to some degree by creating new variables that use the previous days prices as predictors. Specifically, new predictor columns are created, with $y_{t-1}, y_{t-2}...y_{t-x}$. A method of expanding this would also be creating similar lagged variables of several of the predictors, which would especially be relevant if the model has a few specific predictors with large predictive power. If many predictors are present, this method would quickly increase the width of the data set. In our implementation we have limited it to only the price with a four timestep lag, making our model for a given horizon H the equation below:

$$\hat{y}_{t+H} = f(y_t....y_{t-4}, x_t) + \epsilon \tag{4.5}$$

Because of our choice to only predict y_{t+5} and y_{t+22}, we are only required to fit two models. If a complete forecast of the horizon $h = 1, 2, ..., H$ is wanted, the Direct method requires H models to be fitted. Since the H models should ideally be individually tuned, this approach becomes very computationally intensive (Taieb et al., 2012a). In addition, the model for each consecutive day $f_1, f_2, ...f_H$ is estimated independently of each other. This opens up the possibility that each horizon $\hat{y}_{t+h}$ can be estimated in with highly differing coefficients, creating high variance between predictions of consecutive days (Taieb et al., 2012b).

Recursive Approach

If H forecasts are required, a faster solution would be to only fit one model for $\hat{y}_{t+1}$ and use the predicted values of this model to predict each $t+1, ...t+H$ model iteratively. The same lagged variables from the Direct approach are still used, but are now updated with the predictions from the previous time step as it walks forward.

$$\hat{y}_{t+1} = f(y_t....y_{t-4}, x_t) + \epsilon \tag{4.6}$$

$$\hat{y}_{t+2} = f(\hat{y}_{t+1}....y_{t-3}, x_t) + \epsilon \tag{4.7}$$

$$\vdots$$

$$y_{t+5} = f(\hat{y}_{t+4}, .., \hat{y}_{t+1}, y_t, x_t) + \epsilon \tag{4.8}$$

Compared to the Direct strategy, the Recursive approach will be computationally easy, as only one model is fitted. The largest potential issue with using a recursive strategy, is that for each forward step the error of the predictions will accumulate and become more biased for each iteration (Taieb et al., 2012a,b). This is intuitive, as consecutive predictions use an estimate of $\hat{y}_{t-1}$ as a predictor.

A good explanation of the trade-off between the Direct and Recursive approach can be found in Marcellino et al. (2006). Stating that

> "The iterated method produces more efficient parameter estimates than the Direct method, but it is prone to bias if the one-step-ahead model is misspecified. (...) Because it seems implausible that typically low-order autoregressive models are correctly specified, in the sense of estimating the best linear predictor, the theoretical literature tends to conclude that the robustness of the Direct forecast to model misspecification makes it a more attractive procedure than the bias-prone iterated forecast." (Marcellino et al., 2006)

This implies that the Direct strategy should give better results, but requires increased computational time as stated by Taieb et al. (2012a).

More recently, several researchers are suggesting that combinations of these approaches can give improved forecasts, suggesting some different variations (Taieb et al., 2012a,b) which we will address below.

Rectified approach

The working paper by Taieb et al. (2012b) suggests training a Recursive model and an additional *rectifier*. The rectifier is a flexible model fit on the errors of the base model $\hat{\epsilon} = f_{rec}(\hat{y}_{base})$. The reasoning behind this implementation is that if the base model has a bias, such that the mean of the residuals of this model is different from 0, the rectifier should learn this bias and be able to correct it.

Taieb et al. (2012b) suggests using a linear model as a base, with low variance but high bias, and a flexible KNN model as the rectifier. If these cooperate optimally, they should create an ideal compromise of the bias-variance trade-off.

$$\hat{y}_{t+1}^{base} = f_{base}(y_t ... y_{t-4}) + \epsilon \tag{4.9}$$

$$\hat{y}_{rectified} = \hat{y}_{base} + f_{rec}(\hat{y}_{base}) \tag{4.10}$$

In our implementation we test out this method with linear models as the base, and XGBoost models as the rectifier.

Direct Recursive (DirRec)

Another variant suggested by Sorjamaa and Lendasse (2006), is a strategy called *DirRec*. It is similar to the Direct model, in that you construct H models to predict. The element from the Recursive method is that each model f_h is a function of the predictions from previous models $f_{h-1} ... f_{t+1}$, as well as the variables y_t, x_t. An important aspect of the DirRec strategy is that the predictions are added, not replaced, in the subsequent model. The DirRec model f_{t+2} would be equivalent to a Direct fit with an extra predictor $\hat{y}_{t+1}$, the model still accumulates errors from previous forecasts, but keeps all the data from a Direct fit (Sorjamaa and Lendasse, 2006). This is repeated until the desired forecast $\hat{y}_{t+H}$, as described in Equation 4.11 below.

$$\hat{y}_{t+1} = f(y_t ... y_{t-4}, x_t) + \epsilon \tag{4.11}$$

$$\hat{y}_{t+2} = f(\hat{y}_{t+1}, y_t ... y_{t-4}, x_t) + \epsilon \tag{4.12}$$

$$\vdots$$

$$y_{t+5} = f(\hat{y}_{t+1} ... \hat{y}_{t+4}, y_t ... y_{t-4}, x_t) + \epsilon \tag{4.13}$$

The trade-off in this strategy is that the iterative use of former predictions should accumulate error, but accurate predictions of a former timestep can assist the model in predicting the final horizon. To further elaborate this point, we can first compare to the recursive strategy. The accumulated error should always be less than recursive, because each DirRec $t+h$ model still keeps all the original y_t, x_t variables, removing the possibility of predictive power being removed from the model (Sorjamaa and Lendasse, 2006). In addition, given that some explanatory variables have predictive power at different horizons, the $t+h$ model should be better at predicting $t+h$ than an iterated $t+1$ model, thereby having less error per step.

The reason the model can also outperform Direct, is because the XGBoost algorithm selects variables based on their loss reduction, which if functioning optimally should only select the a former $\hat{y}_{t+h}$ as a feature if it adds predictive power. If the model does this efficiently, it will act as an additive model, which together increases performance. One limitation for this method in our experiment, is that DirRec always requires H models, even when you only are interested in only the $\hat{y}_H$ predictions. The computational time is therefore approximately H times the Direct model, if only the horizon H is of interest.

4.7 Tools

The models in this thesis are mostly constructed using the R programming language. All data manipulation is done in R. The libraries used are mostly those part of the *tidyverse* environment, as well as model specific libraries such as XGBoost. To mitigate the computational limitations of ordinary desktop computers, virtual machines on Microsoft Azure was utilized to provide on-demand capacity.

5 Results

5.1 Forecast Results

The models were able to forecast prices with a high degree of accuracy on both horizons. This indicates that the data set compiled by Klaveness contains variables well suited for explaining future development of the spot price. The hit rates are in general high, and the accuracy of the best models precise enough that a trading strategy likely could be utilized. As expected, the predictions were better on the weekly horizon than the monthly, but the difference in accuracy was surprisingly low. In our analysis we will comment on the performance on each model, and try to diagnose why each performed the way they did.

Horizon	Method	RMSE	MAE	Hit Rate	MAPE
Weekly (5)	MA(100)	2312	1866	53%	16.0%
	Random Walk	1470	1173	-	9.9%
	ARIMA	1303	1023	70%	8.6%
	Direct	**729**	**570**	**85%**	**4.8%**
	Recursive	1350	1058	67%	9.0%
	Rectify	1883	1445	64%	12.1%
	DirRec	**697**	**548**	**86%**	**4.6%**
Monthly (22)	MA(100)	2580	2092	56%	18.0%
	Random Walk	2323	1818	-	15.3%
	ARIMA (1,0,2)	2323	1825	54%	15.0%
	Direct	**817**	**616**	**86%**	**5.1%**
	Recursive	2461	1894	58%	15.8%
	Rectify	2271	1735	66%	14.6%
	DirRec	**767**	**583**	**87%**	**4.8%**

Table 5.1: Main results, with time series cross-validation starting at observation 1200

5.1.1 Prediction Performance across Horizons

In Figure 5.1 we can see that the increase in residuals of the different methods starts flattening after a while, indicating that there is more predictive power in some variables for a long horizon rather than the earlier. An important factor here, is that when tuning the models the hyperparameters which minimize loss at $H = 22$ were chosen for all models. The DirRec and Direct model should ideally be tuned for each iteration, but this would require such high amounts of computing power and/or time that it is not feasible for our research. The Recursive model however, is tuned both for a weekly and monthly horizon, resulting in similar performance to ARIMA.

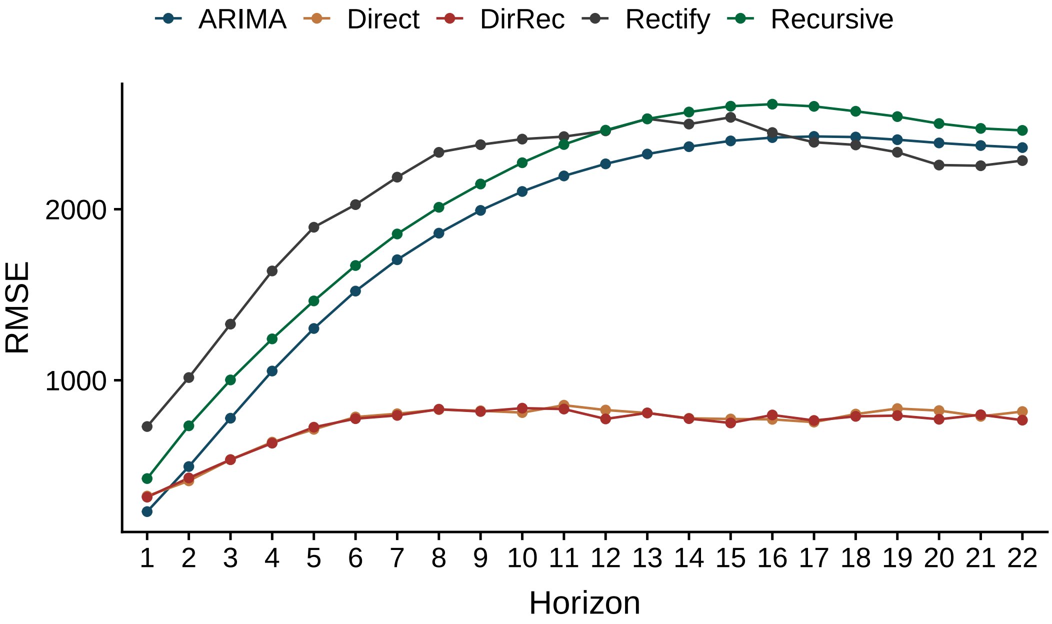


Figure 5.1: RMSE of our main models for different horizons

5.2 Benchmark Models

All results should be seen in comparison to the benchmark models, these are all relatively simple and should be thought of as the bare minimum of performance for a forecasting model. Our analysis will emphasize on the best performing benchmark, ARIMA.

5.2.1 Moving Average

As mentioned in Section 4 the moving average model should perform well on mean reverting data sets. In our case, the 100-day moving average model seems to not be flexible enough to predict well, shown in Figure 5.2. This could be because the spot price likely is not mean reverting on a 100-day horizon.

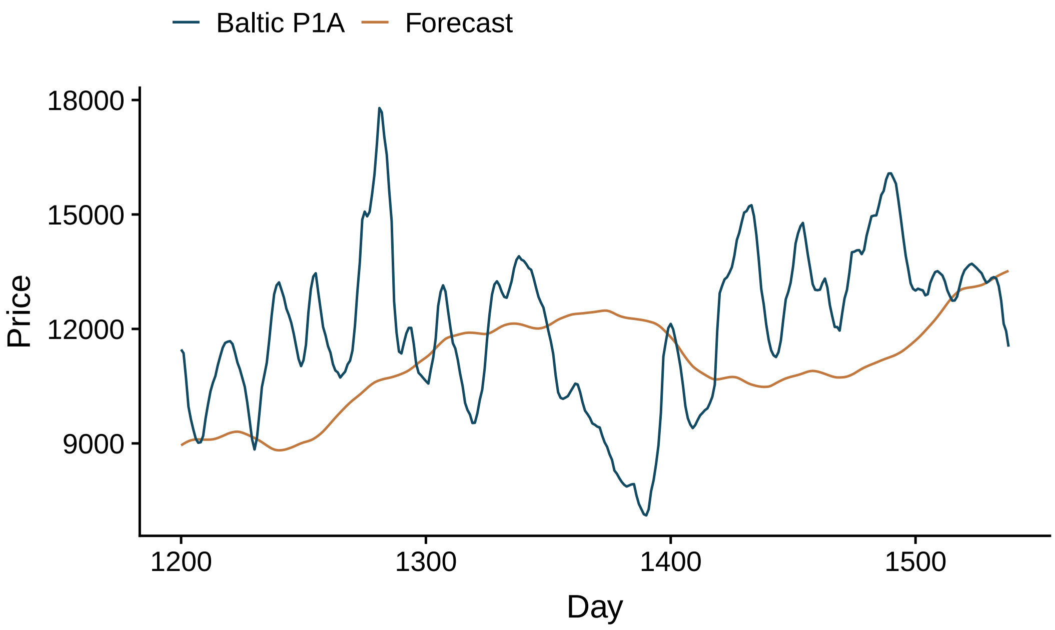
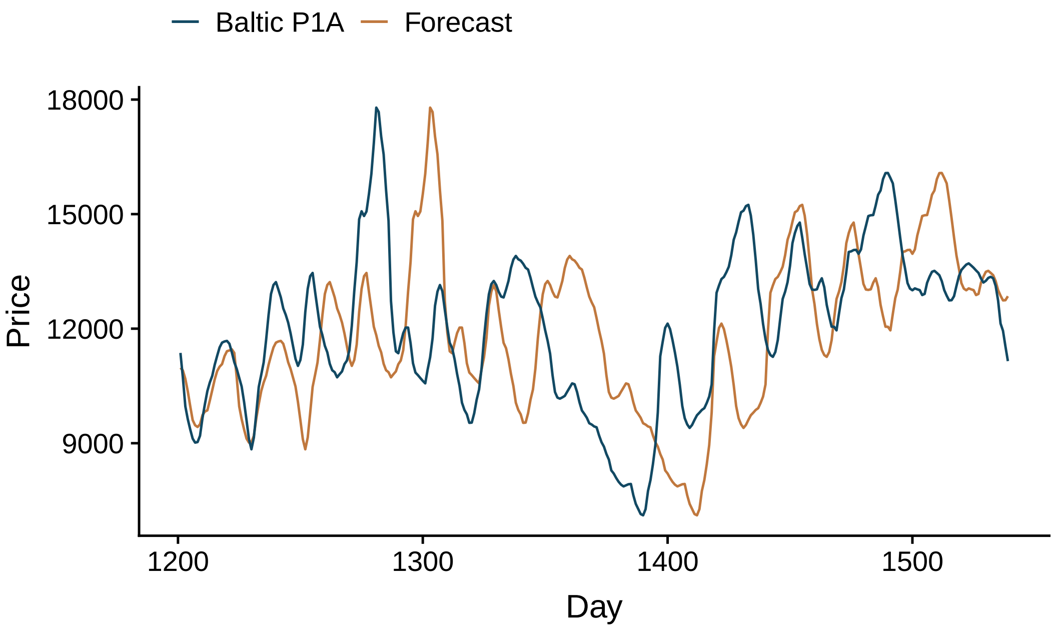


Figure 5.2: Forecast using the 100-day moving average model on a 22 day horizon

5.2.2 Random Walk

The random walk model performs relatively well, and serves as a good benchmark model, with performance better than moving average and almost as good as the ARIMA model. Figure 5.3 displays the behavior of the random walk model. We see that the model forecasts the exact value of Baltic P1A, with a 22 day delay.



S

Figure 5.3: Forecast using the Random walk model on a 22 day horizon

5.2.3 ARIMA

Our best empirically tested ARIMA model appears to be a (1,0,3) model for the 5-day horizon and a (1,0,2) model for the 22-day horizon. These models support the findings of Koekebakker et al. (2006), but contradicts our interpretation of the plots in Figure 3.3 and the residual ACF and PACF plots of the selected ARIMA model in Appendix A2. None of our best performing models contain an integrated term (d), indicating that differencing does not increase the forecasting accuracy. The effect of differencing is evident when comparing the performance of two different ARIMA models with and without an integrated term, seen in Figure 5.4.

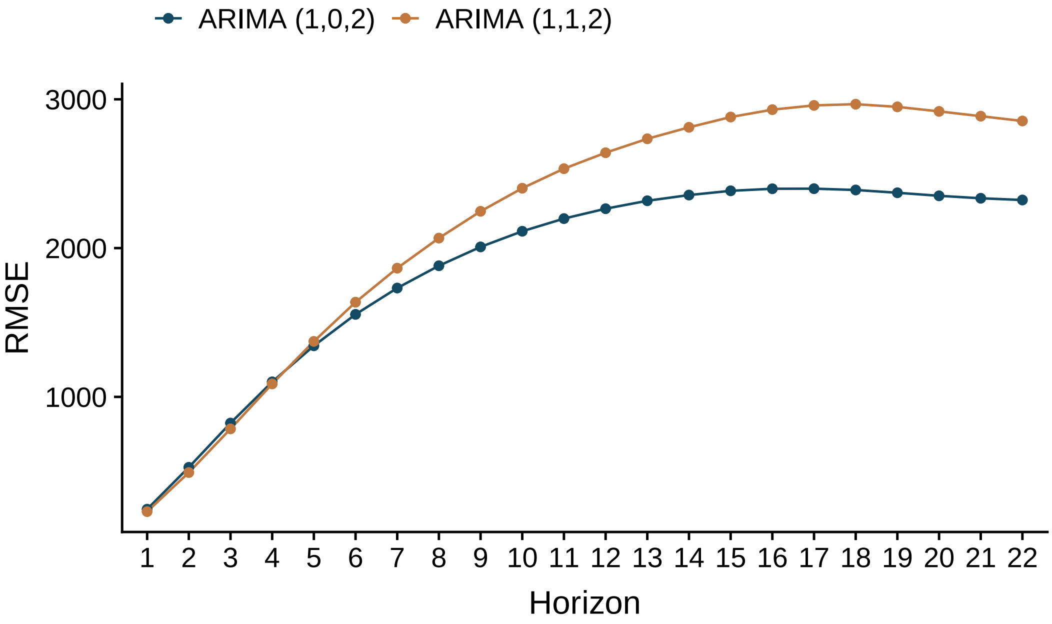


Figure 5.4: Performance comparison between ARIMA(1,0,2) and ARIMA(1,1,2) on a 22 day horizon

In Table 5.1 we see that ARIMA performs very close to the random walk model. An obvious reason is that ARIMA holds much of the same characteristics as the random walk method through the autoregressive term (AR=1). This is apparent in Figure 5.3 and Figure 5.5. This, in addition to the inclusion of moving average term of the prior three days (MA=2), makes ARIMA the best performing benchmark model.

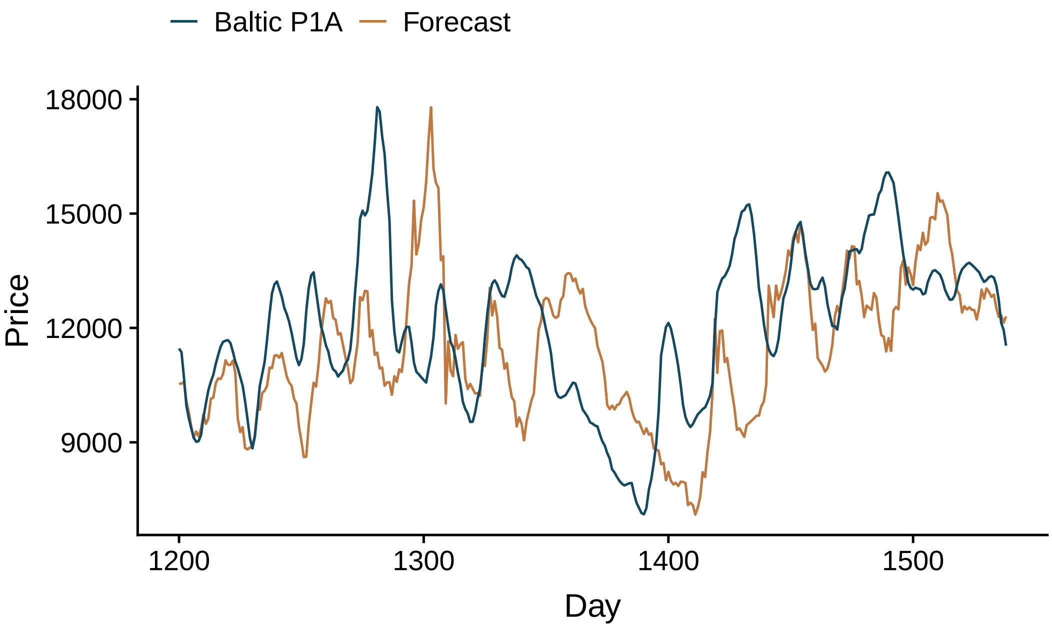


Figure 5.5: Forecast using ARIMA(1,0,2) on a 22 day horizon

5.3 Strategy Evaluation

The Direct strategy proved to be very efficient, with the overall best results on a weekly horizon and very good results on the monthly, outperforming all benchmarks considerably. The Recursive model performs better than the Random Walk model in a weekly horizon, but not better than the ARIMA model. In a monthly horizon it does not perform better than any benchmark.

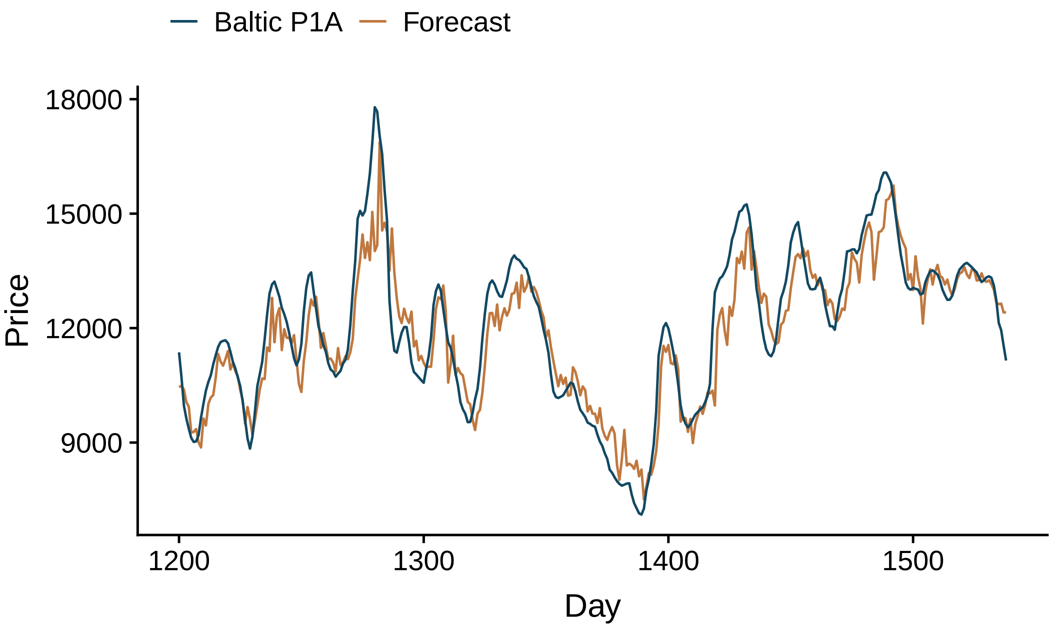


Figure 5.6: Forecast using the Direct model on a 22 day horizon

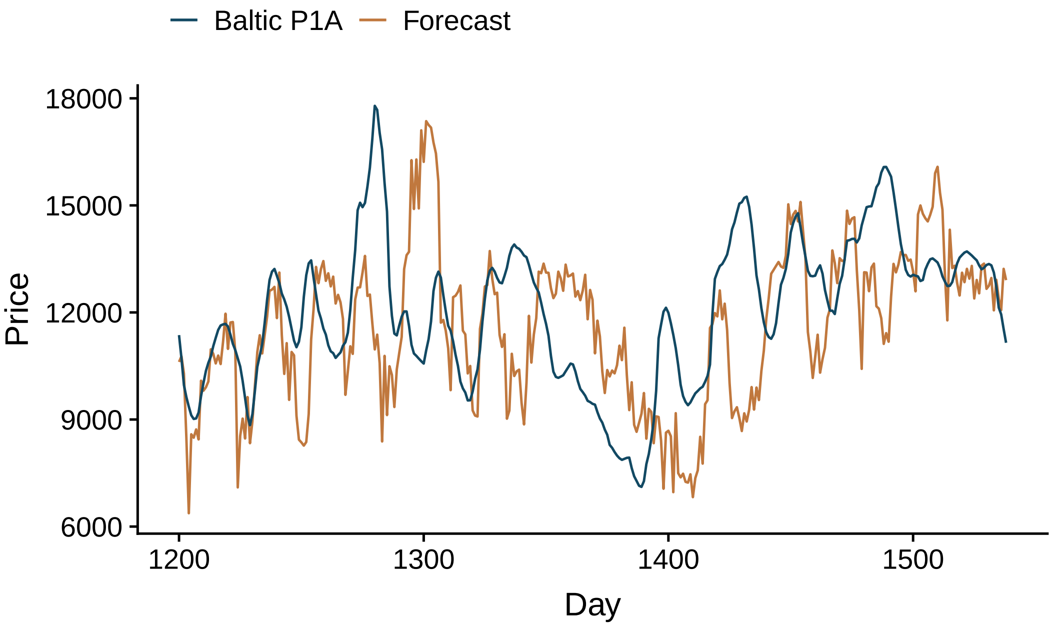


Figure 5.7: Forecast using the Recursive model on a 22 day horizon

Both the Direct and Recursive strategy have comparable computational time, but have a very differing performance. The Recursive strategy performs relatively worse on a longer horizon than a short, in line with Marcellino et al. (2006) that the error of a misspecified model accumulates as the predictions walk forward.

By investigating the Direct models at different horizons, we see that the selected variables slightly differ each time. Our data set is constructed to forecast at a monthly horizon, which is of high interest to shipping companies. Still, predictions are better at a short term than in the long term, but some results point towards many of the features getting increased predictive power on a longer horizon. This will be investigated further in Section 6.

The ideal parameters of the Direct strategy are shown in Table 5.2. The best performing tree depth is quite high, as well as a high degree of column sampling. This can be a consequence of the high number of features, and possibly the presence of interaction effects. Lambda is also present, indicating that the model had a benefit from adding a regularization term.

Hyperparameter	**Value**
Boosting iterations	600
Learning Rate	0.05
Tree Depth	12
Column Sampling	0.25
Row Sampling	1
Lambda	1
Gamma	0

Table 5.2: Hyperparameters for XGBoost model using Direct method

5.3.1 DirRec Strategy

As mentioned in Section 4, the DirRec strategy should perform comparable to the Direct. This is confirmed by our results, as they have very similar scores on our performance metrics. Studying this further, we look at the plots of RMSE at each timestep for DirRec and Direct. In figure 5.8, as well as a density plot in Appendix A4, we can see that the two strategies have similar performance across all horizons. A likely explanation for this is that the feature selection in XGBoost did not choose the previous $\hat{y}_h$ as predictors, thereby neither accumulating errors nor increasing performance. Still, the model proves to be just as good as the Direct strategy, although considerably more computationally heavy if only one horizon H is needed. In fact, the Direct strategy is approximately 18 times faster than the DirRec strategy, on a 22-day horizon in our empirical tests.

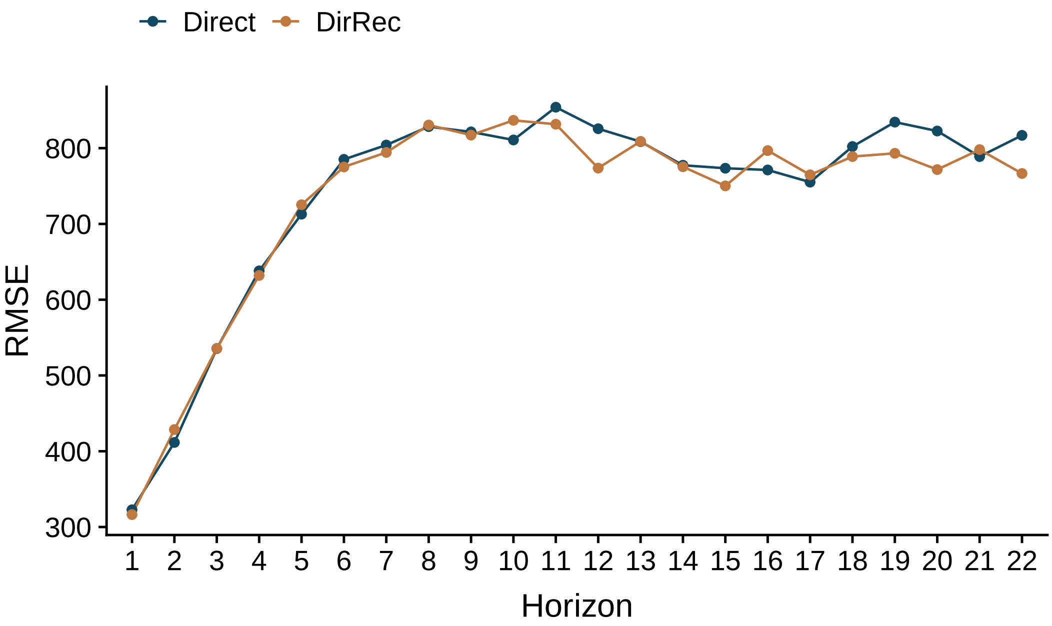


Figure 5.8: RMSE of Direct and DirRec models for each horizon

Parameter-wise, the best performing DirRec model used the same values as the Direct. While this is not necessarily true if more parameters had been tested, it is also no surprise considering that the models are quite similar and can be identical depending on the feature selection.

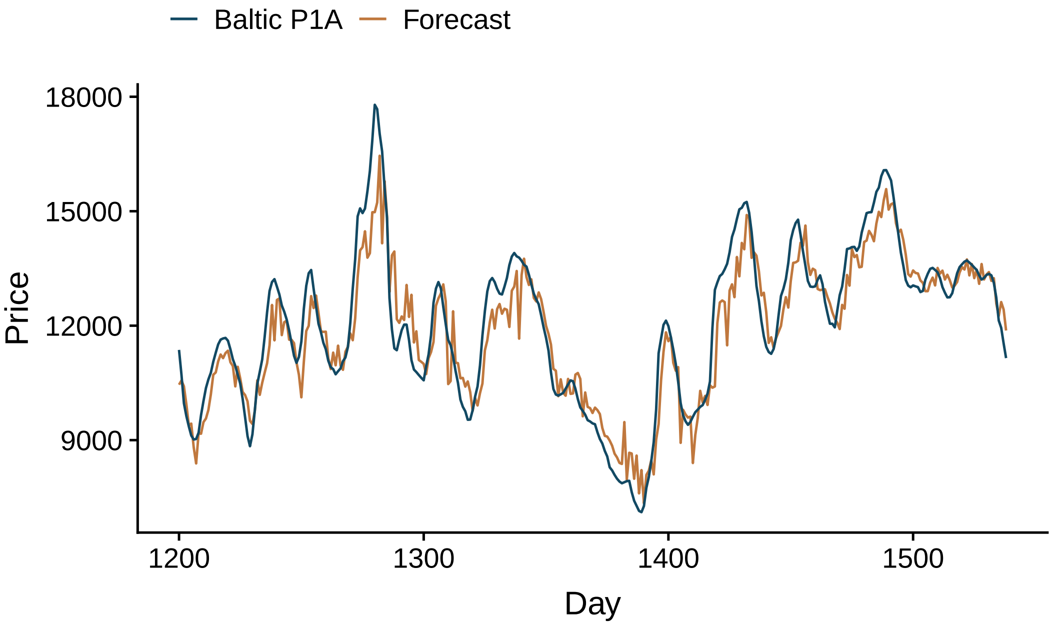


Figure 5.9: Forecasted values using the DirRec model on a 22-day horizon

5.3.2 Rectify Strategy

Finding the best performing implementation of the Rectify strategy proved difficult. We found the intuition from (Taieb et al., 2012b), that a linear model with a high-variance rectifier could produce good results to be somewhat correct. Our best performing base model was therefore an ARIMA(1,0,8), which is considerably smoother than the previously used ARIMA(1,0,2) model. Figure 5.10 shows the rectified predictions, where high variance in the predictions is apparent. If flexible models such as XGBoost were used on both, the models would seem to work against each other rather than together, resulting in inaccurate predictions, and even more variance.

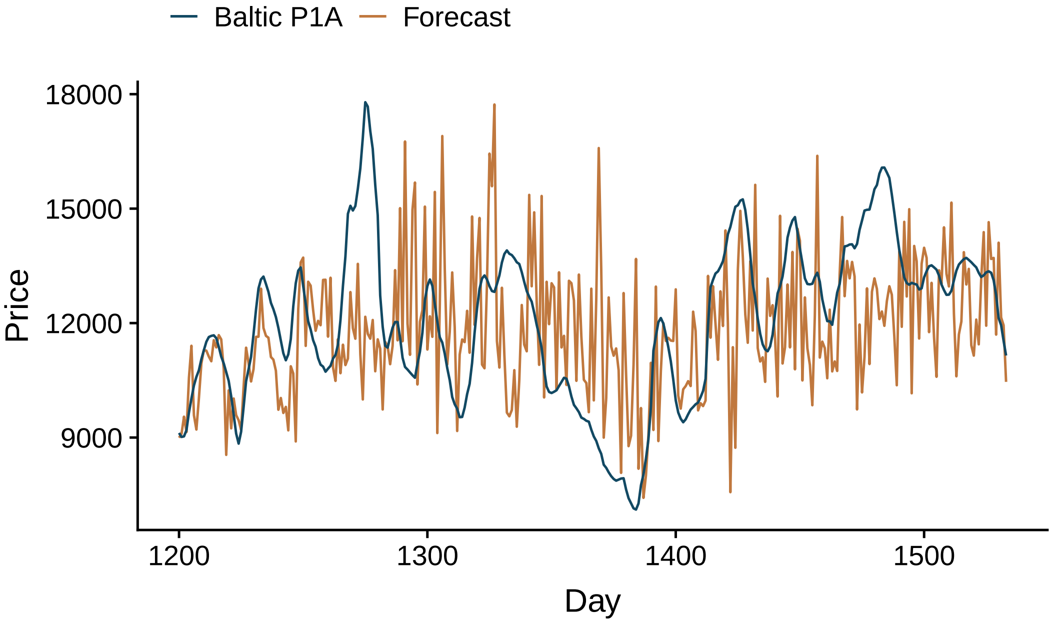


Figure 5.10: Forecasted values using the Rectify model on a 22-day horizon

Repeating the rationale behind this strategy, the linear base model should have low variance and a constant high bias. The flexible rectifier should then correct this by having high variance and low bias, approximating the function of the bias given the predicted values. To research how plausible this scenario is, we can look at the residuals from the underlying ARIMA(1,0,8) model In Figure 5.11. A high bias could manifest as a pattern in the residuals, which the rectifier can use as training data to correct the bias. We see that in the residuals of the base model, there is a downward sloping trend, illustrated by the blue linear regression line. If the rectifier is successful, the final residuals of the forecast should be irreducible error, i.e. randomly distributed around zero. The red regression line illustrates the rectified predictions, clearly with a lower bias. The model becomes a very clear illustration of the bias-variance trade off, where the decreased bias of the rectifier comes with a large increase in variance.

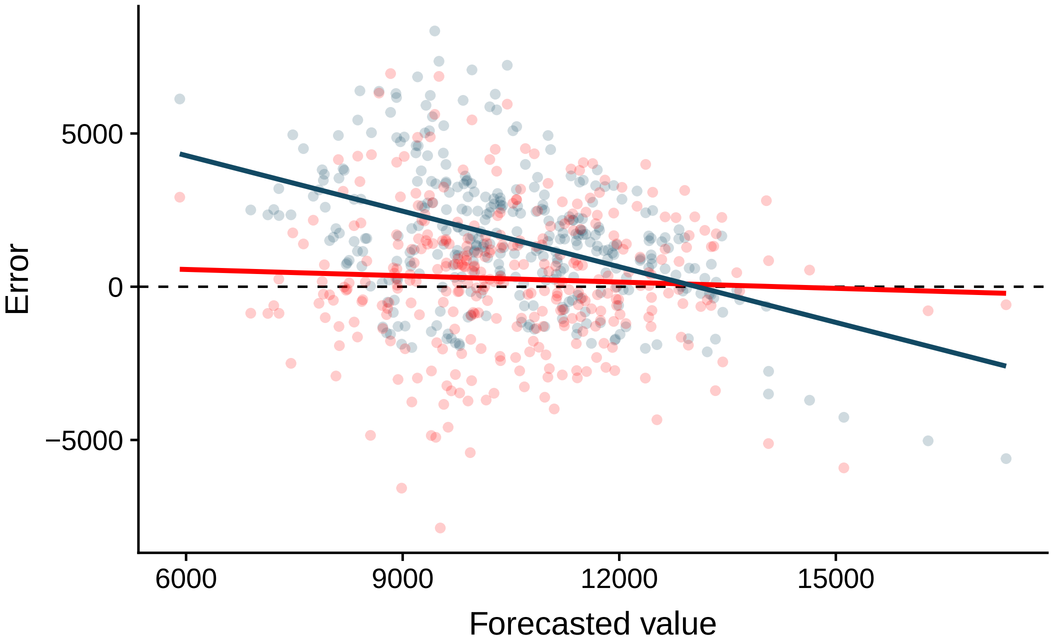


Figure 5.11: Plot of residuals of the base model (blue) and the rectifier (red). A linear regression line is provided for each to illustrate the change in bias by adding the rectifier.

By studying individual observations, we see that the rectifier cannot consistently improve predictions, as many are rectified to increased error, but extreme values at the lower and higher range of predictions are often correctly rectified. Another way of illustrating the rectifier effect, is through a density plot of the residuals shown in Appendix A3.

6 Discussion

6.1 Performance of the Direct and DirRec Strategies

Since the actual fitted models of the two strategies became very similar in our main predictions, we construct some experiments to see if this always is true. Will the Direct and DirRec strategies always perform equally?

Our data set has a high number of features, and mostly time series, with two main assumptions that have been made about the features: (1) They can affect the future spot price, (2) the effect is on a specific horizon (H).

The second is a major assumption, as it cannot possibly be true for all variables of our data set. A simple comparison could be made with two variables, e.g. FFA prices for the next month and the ECB interest rate. Both have the possibility of affecting spot prices, but should intuitively have an effect at a completely different horizon. Since each model that predicts $t + H$ directly, only have access to variables y_t, x_t, they cannot utilize the variables with a lag unequal to H. They can get some degree of predictive power, if these variables are persistent, but the optimal lag relationship between the variables would require the respective x_t to be lagged to the ideal time frame.

The former Direct predictions available to the last DirRec model, represent $H - 1$ models trained to predict their own horizon. As long as these models choose different variables x_t when training the model, they can become proxies for the variables with a specific lag, giving increased predictive power to the final model, if selected by the algorithm.

By calculating the relative importance of each variable in the model, we study some of the selected variables by the DirRec and Direct models at different timesteps. From $H = 10$ and forward, the models start using a future price of a common dry bulk commodity as a predictor, indicating that the stabilization in RMSE in Figure 5.1 could be the effect of this feature's predictive power on $H > 10$. Still, the former predictions of DirRec are not used as predictors in $\hat{y}_H$ and therefore the models will behave similarly, and neither benefit nor be punished from using previous predictions.

6.2 Control Experiments

6.2.1 Univariate Comparison

To research whether this behavior is consistent, we conduct a control experiment by implementing a univariate version of Direct and DirRec, using only the Baltic P1A variable. The DirRec strategy can still use the predictions of previous timesteps. This variant equals the setup that Sorjamaa and Lendasse (2006) used for their analysis. The experiment would remove the effects of x_t, and isolate the possible benefit or downside from the DirRec compared to Direct strategy.

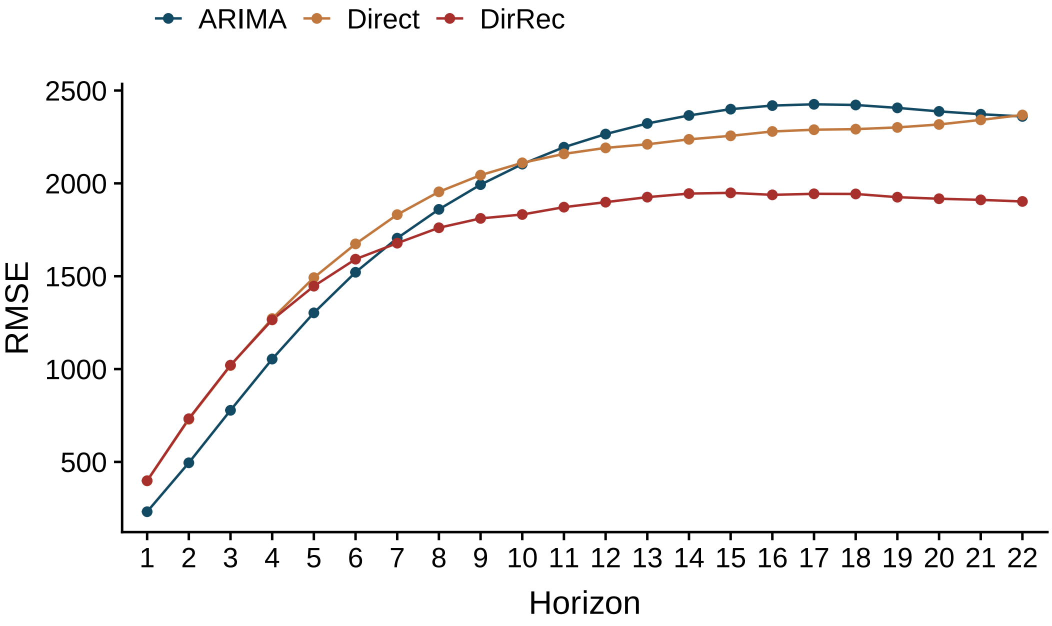


Figure 6.1: Comparison of RMSE for each horizon of ARIMA and univariate models of the DirRec and Direct strategy.

We can see in Figure 6.1 that the DirRec outperforms the Direct strategy when all other predictors are removed. This suggests that the value of having $t + 1....t + H - 1$ forecasts as predictors increase prediction performance. XGBoost can be a very flexible model, and by using the intuition that a model trained for e.g. $t + 15$ outperforms an equal model trained for $t + 22$ on a $t + 15$ horizon, it follows that the $t + 22$ model can be improved by having the former prediction available as a predictor. Sorjamaa and Lendasse (2006), Taieb et al. (2012a), and Bontempi et al. (2013) all refer to this model as DirRec because of its similarities with both the Direct and Recursive strategy. If the underlying fitting method is flexible, a different way of describing it would be an additive model, where the effects of each previous model is taken into account to optimize $t + H$ accuracy.

On a short horizon, ARIMA has lower RMSE than both XGBoost models. As the previous predictions are added to the DirRec model, they deviate, while ARIMA and Direct show similar performance for longer horizons.

6.2.2 Effect of Adding Variables

To isolate the effects of the specific variables x_t, we run another control experiment by training a DirRec model which adds only the noted x_t that the model selected in the later timesteps of the full model. This could isolate the effect of this single commodity, and be compared to the univariate DirRec.

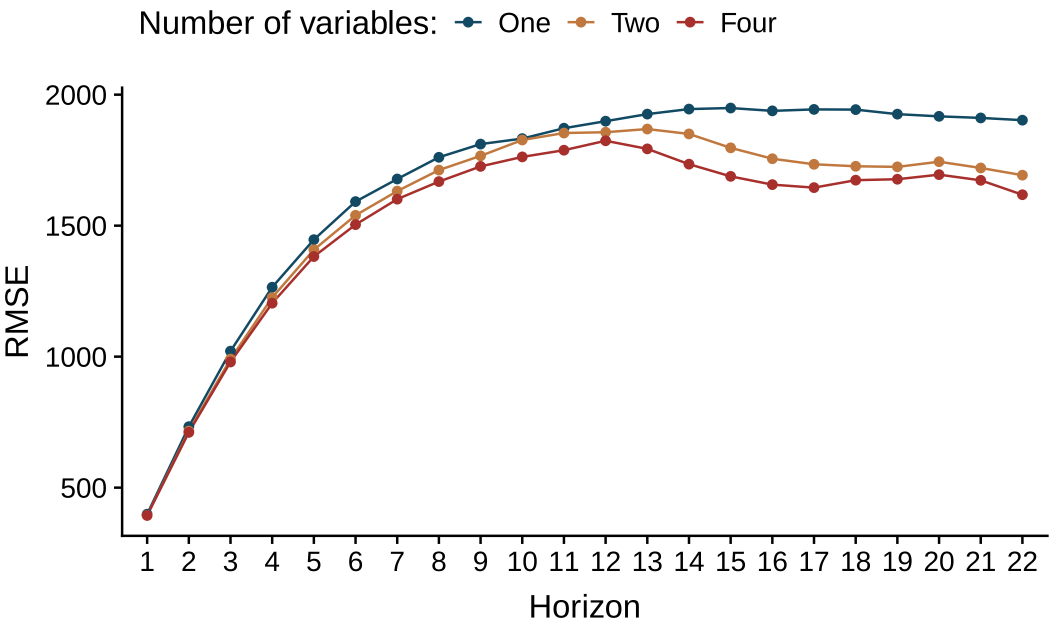


Figure 6.2: Change in RMSE of the DirRec model by adding a few selected variables

As we can see in Figure 6.2 the added variables improve predictions in the long term, indicating a lag relationship between the variables. This effect also increases with more added variables containing information about the same commodity.

In our original model containing all available features, there was no noticeable difference between Direct and DirRec. As mentioned, this could be because the XGBoost algorithm always selected the x_t variables instead of using former predictions $\hat{y}_h$. This would indicate that the possible benefit of using DirRec is overshadowed by the addition of better explanatory variables. To investigate this, we use the four previously used commodity variables on both a Direct and DirRec model, and compare them in Figure 6.3 below.

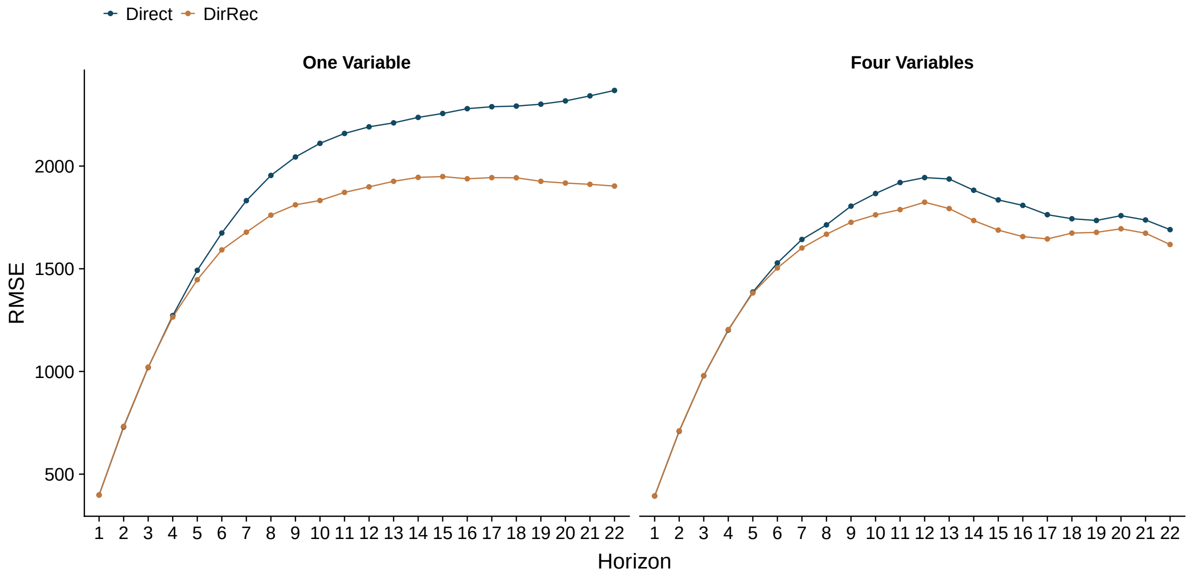


Figure 6.3: Comparison in RMSE per horizon of Direct and DirRec with one and four variables

Even though the DirRec strategy consistently outperforms Direct, the difference between them is gradually reduced as more variables are added. This supports our hypothesis, and suggests that there is some potential in the DirRec strategy which we were not able to fully utilize in our main model.

6.2.3 Handling of Missing Values

Instead of limiting the data set and imputing values, we could have left the missing values handling to the XGBoost algorithm itself. The main issue with this would be the difficulty of testing out other types of models on the same data set, as many require no missing values in order to be fitted. We also performed a test of our models using the raw data set. Even though there was a slight decrease in performance, the difference was small. If speed of implementation is a large concern, using XGBoost without handling missing values would have worked well in our case.

6.3 Robustness and Critique

Our results indicate that we can predict future spot prices to a large extent. This would let us deduce that the data set accurately describes future developments of the spot price. An important indicator for this statement is the varying selection of features, where in a short horizon the Baltic prices is used as predictors, while commodity-related variables are used on a longer horizon. These results coincide with the intuition that movements in commodity markets affect dry bulk shipping, but not as instantaneous as movements of related markets.

Looking at the forecast values, such as in Figure 5.9, we can see that the models predict with an extraordinary accuracy. Despite the whole plotted forecast being at some point a held-out test observation, we cannot exclude the possibility that the data has been overfit. This is one of the more challenging aspects of a prediction problem like ours. As mentioned in Section 4.2.1, the cross-validation approach we have used is good but not perfect. Once a high degree of parameter tuning is done, there is little one can do other than gather new data to further validate the results. In an operational setting, the daily retrieval of new observations would over time give a more precise measure of the true test error.

In addition, a real-life implementation would meet some distinct challenges. Every day, events that simply cannot be predicted, occur. There is no data set which in practice could capture all information required to predict perfectly, as this information is acted upon by humans in the final step. That being said, predictive power is definitely achievable, and can be used to develop methodical hypotheses as well as for taking calculated risks in the market. There is considerable value in reducing risk, but participating in a market will always carry a potential downside.

6.3.1 Lack of Feature Selection

One could argue that feature selection should have been performed despite the built-in abilities available in XGBoost. However, in order to evaluate the performance of the complete method, without the risk of evaluating the feature reduction methods, we have chosen not to perform any feature reduction. We therefore heavily rely on the assumption that the data set by Klaveness in large part are variables which can affect the dry bulk market. If implementing the methods described in this thesis, we recommend thorough feature analysis to avoid spurious relationships.

6.3.2 Limitations of Parameter Tuning

Although a substantial amount of time has been spent tuning hyperparameters, the process does not cover all possible combinations of parameters. The truly optimal parameters are likely not found by our approach, and the chosen parameters should be regarded as an estimate. Some of the parameters may also have negligible impact on the final model, and therefore end up with arbitrary values as the chosen parameter.

6.3.3 Which Method is Preferred When Forecasting the P1A Price?

Using the DirRec strategy gives the best results according to our tests, but it is considerably more computationally heavy than the Direct method. Likely, the DirRec strategy can be used with higher efficiency if careful feature selection is implemented. This being said, it is important to notice that the computational time is significantly reduced when the best performing model is used on a specific forecast, rather than being run through the whole cross-validation process.

6.3.4 Starting Time Series Cross-Validation at *t=1200*

XGBoost does not necessarily require 1200 observations to be fit. The decision to start at 1200 was a practical choice to reduce computation time considerably. The main argument for this decision is that since markets can change greatly over time, the performance several years ago is of less relevance than the most recent years.

To control the robustness of our models, we also ran the best performing models on an increased test set, by running from $t = 800$ and forward. The results did not change dramatically, and can be seen in Appendix A1. Even though this increases the trust in our results slightly, it is only a marginal improvement. To further substantiate our results we use a hold-out test set.

6.3.5 Hold-Out Test Set

As our original data set was from February 2019, we acquired an updated version towards the end of writing this thesis. This served as a hold-out test set, where we could get a test estimate with a low possibility of overfitting. The updated data set added 103 observations, shown in Figure 6.4 below.

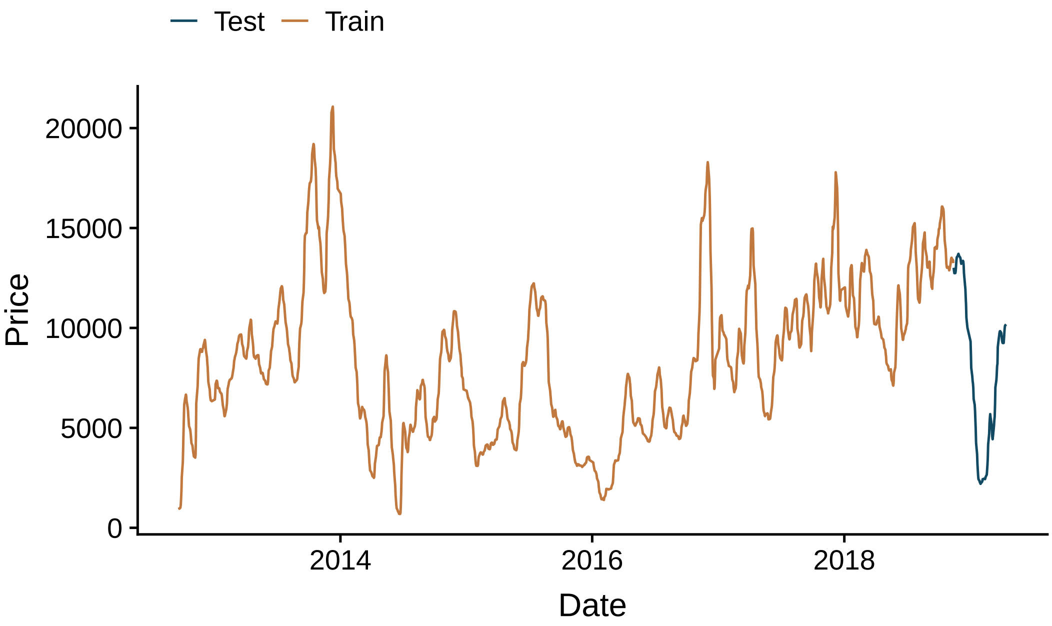


Figure 6.4: Baltic P1A with added test observations

The testing period should give an unbiased test error, but is limited by its small size. As previously stated, random events can affect the results dramatically. Notably, our test period contains the mentioned *Vale Dam* accident in Brazil, contributing to a sharp decline in prices. The results can be viewed in Table 6.1 below. The models and benchmarks performed significantly worse in this period, indicating a less predictable time period. Still, compared to the benchmarks, the models performed consistently with previous results. The Direct and DirRec models continues to have the best performance, with comparable results. A particular detail is the remarkably high hit rates, likely due to the particular movement of the spot price in the short period of the test set. A larger test set would be preferable, but the additional observations still provide considerable confirmation of our findings.

Horizon	Method	RMSE	MAE	Hit Rate	MAPE
Weekly (5)	MA(100)	6150	5012	48%	146.8%
	Random Walk	1515	1190	-	22.2%
	ARIMA (1,0,3)	1233	962	79%	17.9%
	Direct	**814**	**646**	**84%**	**11,6%**
	Recursive	1086	876	70%	15.9%
	Rectify	1434	1011	69%	16.9%
	DirRec	**768**	**619**	**88%**	**11,4%**
Monthly (22)	MA(100)	7110	5938	50%	171.9%
	Random Walk	5358	4794	-	1.1%
	ARIMA (1,0,2)	4876	4267	93%	102.7%
	Direct	**1436**	**1161**	**99%**	**27.6%**
	Recursive	5088	4451	73%	104.3%
	Rectify	3742	2416	86%	45.4%
	DirRec	**1452**	**1158**	**99%**	**28.6%**

Table 6.1: Results table on hold-out test set

The results across horizons were very similar, shown in Figure 6.5. ARIMA and Recursive still has comparable results and DirRec behaves the same as Direct. A notable change is that the Rectify model performs better compared to the previous results, but still not competing with Direct and DirRec. In Figure 6.6 the forecast values of the period are shown.

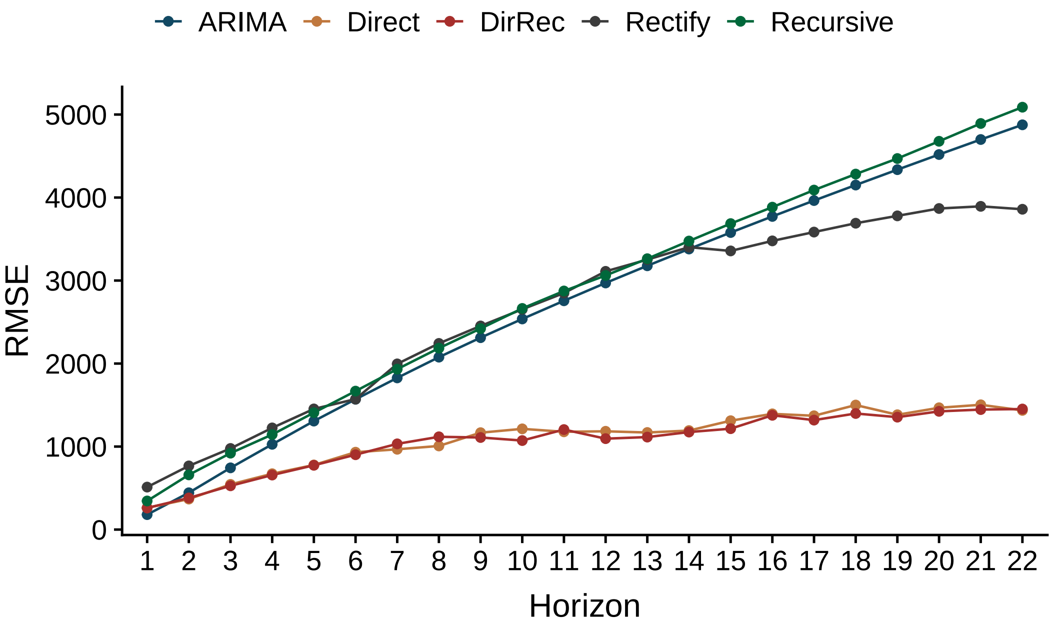


Figure 6.5: Performance comparison across horizons on test-set

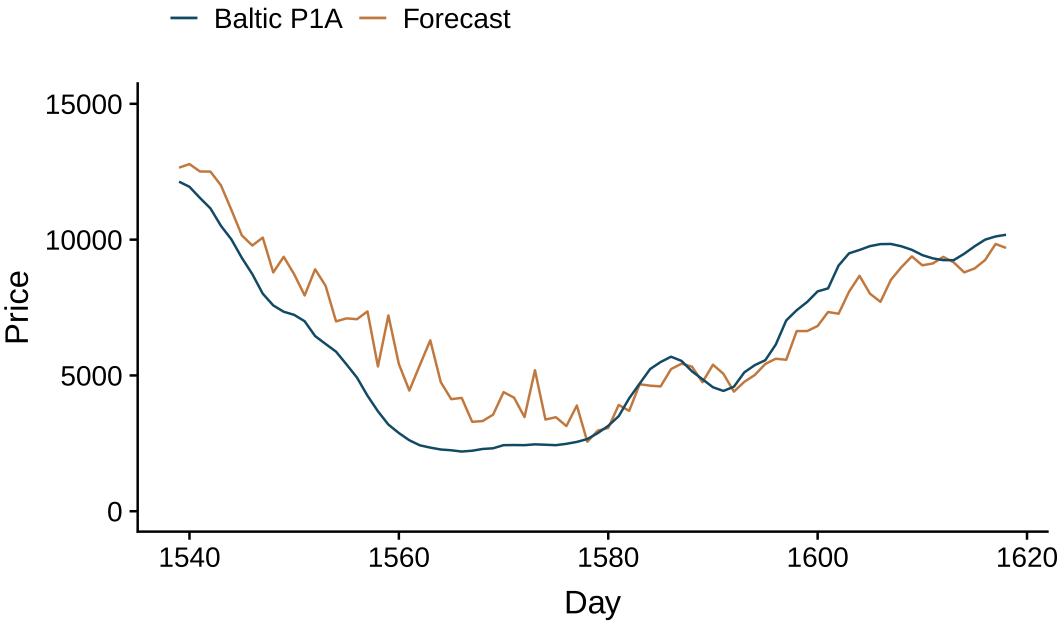


Figure 6.6: Forecast using the DirRec model on test set

Last, the behavior of the univariate models is tested. As shown in Figure 6.7, the results are consistent with our previous discoveries, as the DirRec strategy still outperforms ARIMA and Direct. This further strengthens the possibility that DirRec has potential to consistently outperform Direct.

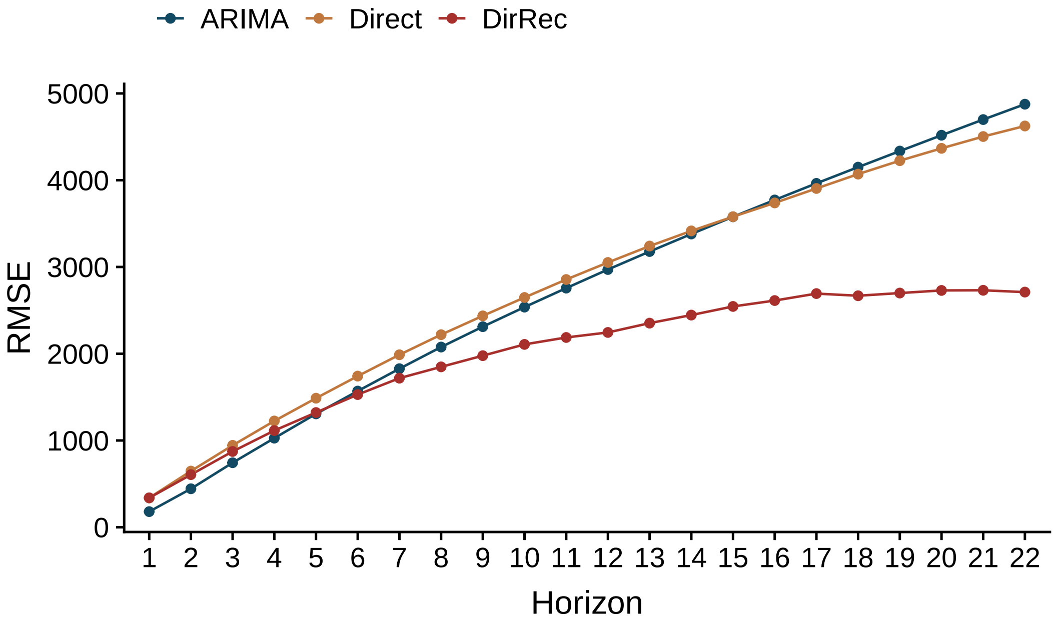


Figure 6.7: Performance across horizons on test-set

6.4 Further Research

This thesis focuses on forecasting prices in the Atlantic area through the Baltic P1A price. This opens for future studies to look into whether similar models and type of data can be applied to dry bulk prices in other geographical regions. In this context, questions one may want to answer is whether prices in different regions demands specific data, or whether it is possible to make a generic set of data which can be used to provide sufficient predictions in all regions. Different regions may have specific factors affecting the price, but are there predictors which can work on all? This thesis can hopefully make the foundation for future predictions of other prices in the dry bulk shipping market.

7 Conclusion

According to our results, the Direct and DirRec strategy greatly outperform our benchmark models when predicting dry bulk spot prices. The drawback of DirRec is a considerable increase in computational time.

Our most interesting discovery, is that our experiments on feature sparse data indicate that there could be some added predictive power by using the DirRec strategy. In the univariate model comparison, the DirRec strategy performed significantly better than both ARIMA and Direct, but the difference in performance was diminishing as more explanatory variables were added to Direct and DirRec. Our supposition is that the individual prediction models for each timestep perform better at their specified horizon, enabling an additive model to perform better at the final forecast horizon compared to a direct prediction.

References

Adland, R. and Alizadeh, A. H. (2018). Explaining price differences between physical and derivative freight contracts. *Transportation Research Part E: Logistics and Transportation Review*, 118:20–33.

Baltic Exchange (2019a). Baltic exchange shipping market information derivative freight about. Retrieved from https://www.balticexchange.com/about-us/. (Accessed on 05/01/2019).

Baltic Exchange (2019b). Indices (routes). Retrieved from https://www.balticexchange.com/market-information/product-overview/indics/. (Accessed on 05/01/2019).

Bontempi, G., Ben Taieb, S., and Le Borgne, Y.-A. (2013). *Machine Learning Strategies for Time Series Forecasting*, volume 138.

Chen, T. (2018). What is the difference between the r gbm and xgboost? - quora. Retrieved from https://www.quora.com/What-is-the-difference-between-the-R-gbm-gradient-boosting-machine-and-xgboost-extreme-gradient-boosting. (Accessed on 05/23/2019).

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*.

De Prado, M. L. (2018). *Advances in financial machine learning.* John Wiley & Sons, New Jersey.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Hastie, T., Tibshirani, R., and Friedman, J. (2013). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series in Statistics. Springer New York.

Hyndman, R. J. (2014). Unit root tests and arima models | rob j hyndman. https://robjhyndman.com/hyndsight/unit-root-tests/?fbclid=IwAR12-v25Mp71_6TIabtjARr1TZ_CNgXGF6ouflnUddKb7RS7X-15o1SX6fU. (Accessed on 05/04/2019).

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice.* OTexts.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer, New York.

Klaveness (2019). About us » torvald klaveness. Retrieved from https://klaveness.com/about-us/. (Accessed on 04/03/2019).

Koekebakker, S., Adland, R., and Sødal, S. (2006). Are spot freight rates stationary? *Journal of Transport Economics and Policy*, 40(3):449–472.

Mandra, J. O. (2019). Vale dam collapse and impact on shipping | world maritime news. Retrieved from https://worldmaritimenews.com/archives/269886/vale-dam-collapse-and-impact-on-shipping/. (Accessed on 04/03/2019).

Marcellino, M., Stock, J. H., and Watson, M. W. (2006). A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series. *Journal of Econometrics*, 135(1):499 – 526.

Nielsen, D. (2016). Tree boosting with xgboost-why does xgboost win" every" machine learning competition? Master's thesis, NTNU.

Sorjamaa, A. and Lendasse, A. (2006). Time series prediction using dirrec strategy. In *Esann*, volume 6, pages 143–148.

Stopford, M. (2009). *Maritime economics 3e.* Routledge.

Taieb, S. B., Bontempi, G., Atiya, A. F., and Sorjamaa, A. (2012a). A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. *Expert Systems with Applications*, 39(8):7067 – 7083.

Taieb, S. B., Hyndman, R. J., et al. (2012b). *Recursive and direct multi-step forecasting: the best of both worlds*, volume 19. Citeseer.

United Nations (2018). *Review of Maritime Transport 2018.* United Nations, New York.

XGBoost Documentation (2019). Xgboost parameters — xgboost 0.83.dev0 documentation. Retrieved from https://xgboost.readthedocs.io/en/latest/parameter.html. (Accessed on 05/04/2019).

Zivot, E. and Wang, J. (2003). *Unit Root Tests*, pages 105–127. Springer New York, New York, NY.

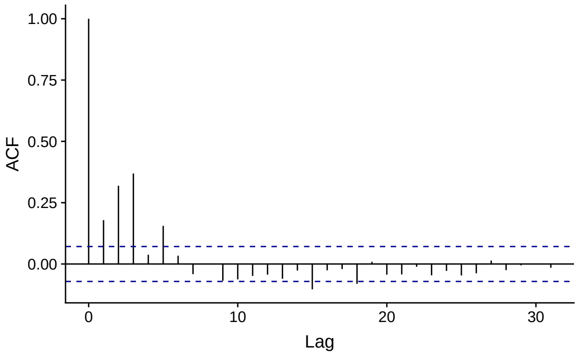
Appendix

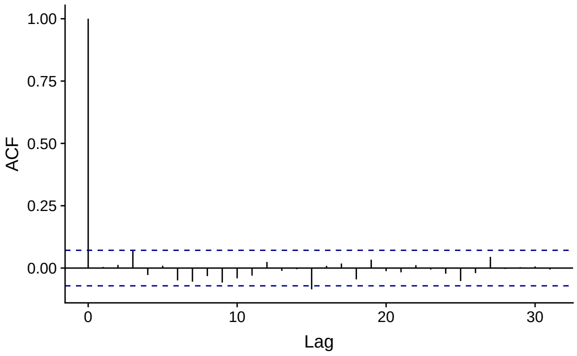
A1 Results Table with Predictions Starting at 800

Horizon	Method	RMSE	MAE	Hit Rate	MAPE	Comp.Time
Weekly (5)	MA(100)	2724	2035	52%	27.3%	20 seconds
	Random Walk	1484	1086	-	12.1%	-
	ARIMA	1268	918	72%	10.1%	5.1 minutes
	Direct	**664**	**495**	**84%**	5.7%	12.4 minutes
	Recursive	1309	943	66%	10.8%	24.9 minutes
	Rectify	1839	1300	64%	13.6%	20.5 minutes
	DirRec	**663**	**486**	**87%**	**5.5%**	3.5 hours
Monthly (22)	MA(100)	3080	2301	61%	31.4%	17 seconds
	Random Walk	3053	2308	-	28.0%	-
	ARIMA (1,0,2)	2939	2231	58%	27.0%	6.7 minutes
	Direct	**843**	**621**	**92%**	**7.0%**	42.0 minutes
	Recursive	3067	2306	57%	27.6%	4.8 minutes
	Rectify	3142	2213	67%	24.6%	5.8 minutes
	DirRec	**786**	**578**	**93%**	6.7%	15.7 hours

Table A1.1: Results table with predictions starting at 800

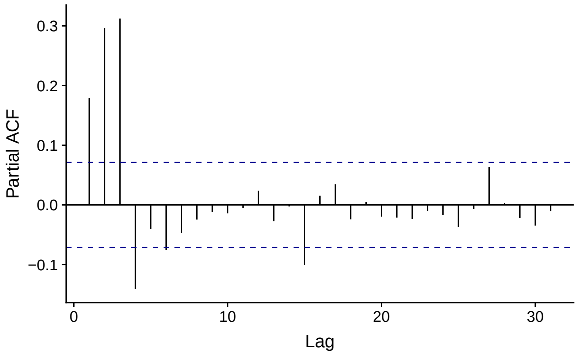
A2 Comparison of Residual ACF and PACF Plots of ARIMA Models

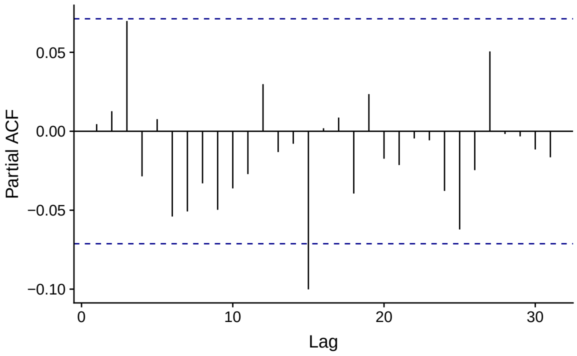




(a) Residual ACF ARIMA(1,0,2)

(b) Residual ACF ARIMA(1,1,2)





(c) Residual PACF ARIMA(1,0,2)

(d) Residual PACF ARIMA(1,1,2)

A3 Errors of Base Prediction Compared to Rectified Prediction

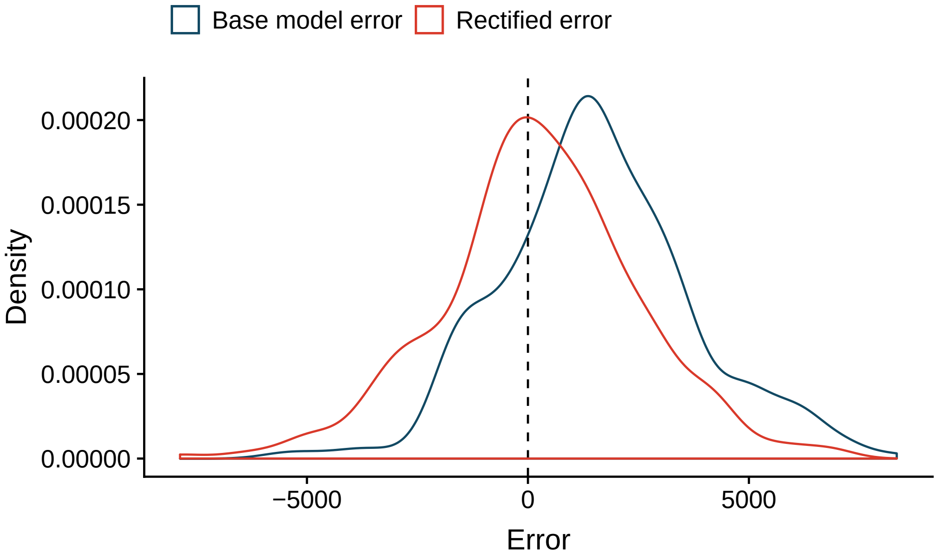


Figure A3.1: Density plot of errors, Rectify

A4 Errors of Direct Compared to DirRec

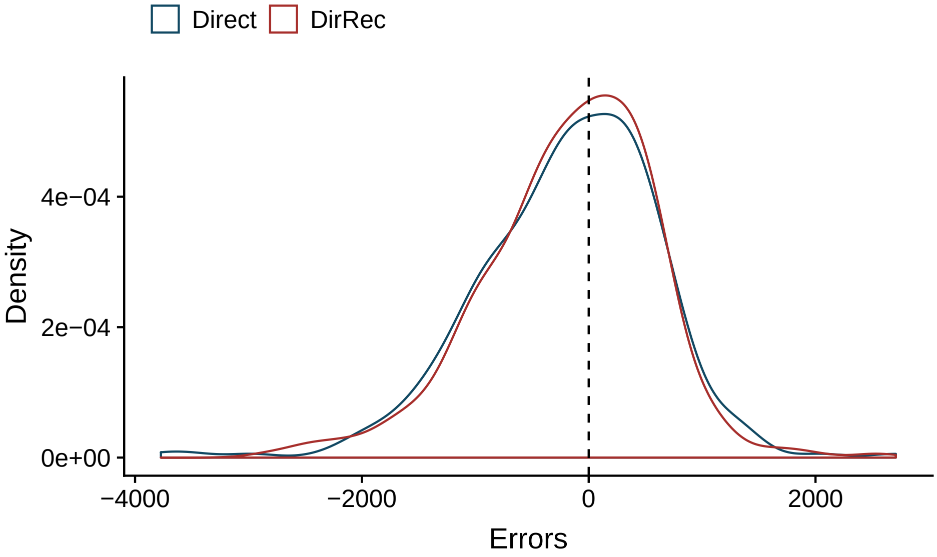


Figure A4.1: Density plot of errors, Direct and DirRec