



# Machine learning as a decision support system in Capesize route optimization

*Predicting optimal route selection using Recurrent Neural Networks and  
Extreme Gradient Boosting*

**Herman Johan Bomholt & Torsten Stangeland Thune**

**Supervisor: Roar Os Ådland**

Master thesis, Economics and Business Administration

Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

This thesis was written as a part of the Master of Science in Economics and Business Administration at NHH. Please note that neither the institution nor the examiners are responsible – through the approval of this thesis – for the theories and methods used, or results and conclusions drawn in this work.



---

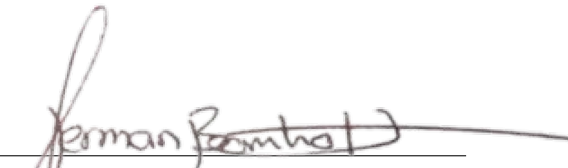
# Acknowledgements

This thesis is written as a part of our master's degree with a specialization in Business Analytics at the Norwegian School of Economics (NHH).

First and foremost, we would like to extend our sincere gratitude for the discussions and mentoring provided by our supervisor, Roar Os Ådland. His extensive knowledge within the field of maritime economics have been essential, providing valuable input and constructive criticism throughout the process. We would also like to thank postdoctoral researcher, Vit Prochazka for vital counseling regarding optimization of vessel allocation through dynamic programming. Lastly, we would like to express our appreciation for the grants provided to us by the Norwegian Shipowners' Association.


Norwegian School of Economics

Bergen, June 2020



---

Herman Johan Bomholt



---

Torsten Stangeland Thune

# Abstract

In this thesis, we investigate the applicability of machine learning to predict optimal route decisions for Capesize vessels. Our approach uses windows of historical macroeconomic and market-specific variables to form a time-series classification problem. We fit a Long-Short-Term-Memory neural network and an Extreme Gradient Boosting model on historical optimal trip choices and predict an out-of-sample routing strategy. By relying on historical input and no knowledge of the future, we can compute possible economic gains, and evaluate the viability of machine learning as a decision support system in route optimization.

In a simplified scenario, with two geographically different roundtrips, we evaluate cumulative earnings throughout a three-year period. Our findings suggest the machine learning methods can outperform an approximation of the average earnings of market participants by almost 11%. Our thesis contributes to the existing literature on spatial efficiency and routing optimization in the dry bulk market and provides insights into a possible method of using machine learning in out-of-sample route prediction.

**Keywords** – Capesize dry bulk market, route optimization, XGBoost, LSTM

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Literature Review</b>	<b>3</b>
<b>3</b>	<b>Data</b>	<b>6</b>
<b>4</b>	<b>Machine Learning Theory</b>	<b>11</b>
4.1	Walk Forward Validation . . . . .	11
4.2	Recurrent Neural Networks . . . . .	11
4.2.1	Long-Short-Term-Memory . . . . .	12
4.3	Extreme Gradient Boosting . . . . .	15
<b>5</b>	<b>Methodology</b>	<b>17</b>
5.1	Chartering Strategy . . . . .	17
5.1.1	Route optimization . . . . .	17
5.1.2	Benchmark Method . . . . .	18
5.2	Machine Learning . . . . .	19
5.2.1	Train-Test Split . . . . .	19
5.2.2	Supervised Learning Data Representation . . . . .	20
5.2.3	Model training . . . . .	21
5.3	Evaluation . . . . .	21
<b>6</b>	<b>Results and Discussion</b>	<b>24</b>
6.1	Perfect Foresight and Benchmark Results . . . . .	24
6.2	Predictive Models and Model Evaluation . . . . .	25
6.3	Expected Cumulative Earnings . . . . .	27
6.4	Limitations and Further Research . . . . .	31
<b>7</b>	<b>Conclusion</b>	<b>33</b>
	<b>References</b>	<b>35</b>
	<b>Appendix</b>	<b>38</b>
A1	Index Volatility . . . . .	38
A2	Tripdetails . . . . .	38
A2.1	Historical Tripcharter Rates . . . . .	39
A3	Confusion Matrix . . . . .	39
A4	T-Test . . . . .	40
A5	Individual Vessel Results . . . . .	40
A6	Variable Importance . . . . .	41

## List of Figures

4.1	<i>Cyclical behaviour of recurrent neural networks (RNNs)</i> Goodfellow et al. (2016). . . . .	12
6.1	Comparison of the perfect foresight and coin strategy cumulative earnings over the entire time horizon, October 2011 to December 2019. . . . .	24
6.2	Comparison of prediction accuracy and Value of Switch . . . . .	26
6.3	Cumulative earnings from LSTM and XGBoost vessel routing, compared to perfect foresight and benchmark. . . . .	28
6.4	Relative cumulative earnings for a fleet of 9 additional ships starting 50 days apart . . . . .	30
A1.1	Historical values of the Baltic Exchange Capesize drybulk indices . . . .	38
A2.1	Historical Tripcharter Rates. . . . .	39
A6.1	XGBoost Variable Importance Plot. . . . .	41

## List of Tables

3.1	Selection of explanatory variables used to predict future trip choices (10 out of 69 total variables) . . . . .	7
6.1	Finalized models; an overview of hyperparameters and out-of-sample prediction accuracy . . . . .	25
A2.1	Abstract of <i>Bulkcarrier Voyage Details (2009 Onwards)</i> , obtained from Clarkson Research . . . . .	38
A3.1	Confusion Matrix with XGBoost predictions. . . . .	39
A3.2	Confusion Matrix with LSTM predictions. . . . .	39
A4.1	Value of switch sorted by predictions. . . . .	40
A4.2	Output from one-sided t-test with unequal variance. . . . .	40
A5.1	Overview of cumulative earnings achieved by each vessel in the fleet. . . . .	40
A5.2	Overview of average daily earnings in USD achieved by each vessel in the fleet. . . . .	40

# 1 Introduction

Historically, the bulk shipping markets have been considered perfect competitive markets, implying co-integration between vessel earnings throughout the global trade routes. Recent studies, however, suggest the dry bulk markets to be spatial inefficient in the short run, creating opportunities to increase economic gains by proper optimization of vessel routing (Adland et al., 2017; Prochazka et al., 2019). The geographical markets are immensely complex and require deep expertise and years of industry experience to understand fully. However, in recent years the availability and quality of data have created new opportunities to model complex relationships without deep industry experience. Hence, proper data analysis could function as a decision-making tool when designing chartering strategies.

In this thesis, we propose a machine learning approach for optimal vessel routing through space and time to maximize expected earnings. The method is formulated as a time-series classification (TSC) problem using historically optimal routes and related explanatory features. Time-series classification is a technique used to predict different classes based on sequential behavior throughout time. Unlike traditional classification, TSC takes into account the ordering of the data, thus interpreting data points as dependent on each other. Consequently, the proposed method is implementable for real-world applications as we rely solely on information from the past without knowledge of the future.

We implement the model in a simplified world scenario, with two different roundtrip estimations based on the China-Brazil and China-Australia Capesize routes. Consider a Capesize vessel positioned in China has the choice to either sail to Australia and back, or take the longer trip to Brazil and back. As the vessel returns, the same process of trip choice is repeated. If Brazil is chosen, they might miss out on high future earnings because of the prolonged employment time. In contrast, the shorter Australia trip could yield opportunities to capitalize on next week's increased rates. If freight rates were to fall in the next couple of months, it could be beneficial with prolonged employment time and earning fixed rates. Decision-makers need to evaluate the movements of the market to appropriately design routing strategies, taking into account the entire time-horizon and not only optimal local decisions.

Capesize is used as the vessel of choice for our implementation because of the freight



rate volatility, a limited number of possible routes, and homogeneous cargo (Kavussanos et al., 2010). The market volatility (See Figure A1.1) could make it possible to profit from understanding prices correctly, especially if there are predictable regional differences. The Capesize vessels are too large to use the Panama and Suez Canal, with a typical ship size of around 175,000 deadweight tonnes (DWT). The vast size also makes the market dominated by two resources, iron ore and coal. Capesize routes are therefore fixed between the major exporters and importers of these resources. Because the market is dominated by a few key resources and trading partners, it is possible to model a simplified two-trip scenario that still covers a significant part of the market (Stopford, 2009).

This thesis aims to evaluate the feasibility of historical information for future route optimization. Previous relevant studies, like Prochazka et al. (2019), have focused on the possibility of optimizing chartering through space and time using perfect information on some of, or the entirety of, the future time-horizon. Whereas these studies provide deep insight into the dynamics and regional differences of freight markets and define the *theoretical framework*, we wish to use these findings to suggest a real-world, applicable methodology.

Subsequently, we are evaluating future market movements and how to utilize them based on previous cycles and explanatory industry observations. Machine learning models are able to model historical inter-relationships too complex and time-consuming for humans to calculate, providing valuable insight that decision-makers can use to evaluate their future outlook. Naturally, machine learning cannot directly replace intuition and deep experience in the field. Still, it could serve as a decision support system in operational planning by extracting signals and relationships useful when designing chartering strategies.

The remainder of the thesis is structured as follows. First, we will review relevant literature regarding route optimization and freight rate forecasting. Second, we will discuss the freight rates and explanatory data that are selected based on traditional market drivers. Next, the theoretical framework will focus on the machine learning techniques applied. The methodology section explains the implementation of the optimization and how these results are processed to represent a supervised learning problem. Ultimately, the results section highlights the prediction accuracy and possible economic gains relative to the market average estimation.

---

## 2 Literature Review

The literature review will first cover relevant past work on route optimization. Further, we will focus on relevant literature covering spatial efficiency and freight rate forecasting. These can be considered prerequisites to achieve economic gains by applying the method suggested.

The optimization problem in this thesis will focus on tramp shipping. Tramp shipping involves accepting contracts for spot cargoes between port pairs in a transport network. The traditional literature on shipping route optimization is usually focused on liner shipping networks (Christiansen et al., 2013). Unlike tramp shipping, liner shipping networks operate with fixed routes, time windows, and specific cargo contracts (Hemmati et al., 2014). The tramp shipping market on the other hand, is defined by a large number of ships and cargoes that are constantly matched by shipbrokers in a perfectly competitive market (Prochazka et al., 2019).

Prochazka et al. (2019) present a method to use knowledge of future prices to solve the optimization problem of route selection in tramp shipping. They apply a dynamic assignment problem with stochastic travel times and known freight rates for the entire planning period (hereby known as perfect foresight). To highlight the effect of geographic location and changes in freight rates over time, they also assume the ship operator as a price taker, where decisions made do not affect the future market, and constant cargo availability.<sup>1</sup> The results obtained from the perfect foresight method reveal that the economic gains relative to the market average were the largest in the Capesize sector. With perfect knowledge of the future, they found one could outperform the benchmark strategy with 23% in the period 2009–2016. The assumption of perfect foresight makes the results unrealistic to achieve. However, they serve as an upper bound to any realistically achievable earnings and imply that there is a potential for exploiting spatial inefficiencies to optimize route selection and increase profits.

Prochazka et al. (2019) also introduces a new solution to the typical *end of planning horizon problem* by using policy approximation. The policy approximation is performed

---

<sup>1</sup>The last assumption is justified by assuming the model is made for a relatively modern ship that due to its energy efficiency and relatively lower marginal cost will remain employed even in times with low freight rate and limited availability of cargo.

by using limited foresight of future prices to predict the optimal decisions.<sup>2</sup> They apply a neural network algorithm trained on the optimal decisions found by the perfect foresight method, using limited foresight of future prices as input. The results obtained show that with limited foresight, they still outperform the market average. However, the results are not as robust as with perfect foresight. The limited foresight method is not directly implementable in real life, as it demands knowledge of future prices to make predictions.

There have been two main ways of investigating spatial efficiency in the shipping sector, co-integration analysis of freight rates, and finding profitable trading strategies. The co-integration analysis has been performed on the movement of freight rates in different geographical regions. Glen and Rogers (1997) and Berg-Andreassen (1997) found freight rates within a sector to be non-stationary and co-integrated following the Engle and Granger (1987) definition. These findings imply that the freight rates share a long-run equilibrium and will only deviate from the equilibrium in short time spans (Engle and Granger, 1987). However, Koekebakker et al. (2006) argues that the findings of non-stationarity in regional freight rates are questionable due to the weak power of the statistical tests to reject non-stationarity. By using appropriate statistical tests, they find regional freight rates to be stationary. Consequently, questioning the validity of co-integration as the Engle and Granger (1987) definition assumes non-stationary time series, integrated of order one.

Adland and Strandenes (2006) initiated the idea of finding profitable trading strategies to uncover spatial inefficiencies in the shipping market. Their findings suggested a tanker operator could achieve improved earnings by applying kernel smoothing to identify peaks and troughs in the market. Tsioumas and Papadimitriou (2015) followed with an investigation on trading rules based on technical analysis of tripcharter rates and found it was possible to outperform the benchmark strategy. Adland et al. (2017) argued there is evidence that the Capesize bulk market is not spatially efficient, as they found a significant premium in the Atlantic basin over time. Furthermore, they comment that this premium would, if common knowledge, be corrected by the decision-makers in the market, unlike previous assumptions made by Laulajainen (2007). Ultimately, Adland et al. (2018) highlights how the co-integration of freight rates does not necessarily make the shipping markets spatial efficient, seeing as the short-term regional difference might

---

<sup>2</sup>They test with 20, 50 and 80 days.

still allow for well-informed chartering strategies to take advantage of spatial inefficiency. Even though we will not forecast freight rate directly, investigating the relevant literature on freight rate forecasting will indicate the ability of machine learning to pick up meaningful information in historical data. The topic is not new and is thoroughly covered in the existing literature. Batchelor et al. (2007) tested the performance of traditional time series techniques on both spot rates and forward rates.<sup>3</sup> Benth and Koekebakker (2016) used univariate stochastic modeling on the Supramax market and found that short-term movements in spot rates were, to some degree, predictable. Lyridis et al. (2004) applied a univariate neural net on the tanker spot price, while Fan et al. (2013) obtained better results using a multivariate neural network. Kanamoto et al. (2019) forecasted the Baltic Capesize Index with a multivariate long-short-term-memory network and got promising results. Overall, a multivariate approach to capture relevant market information has shown the most effective and has given decent accuracy, at least in the short-term.

This thesis contributes to the literature by using the method developed by Prochazka et al. (2019) as a foundation for a real-life implementable chartering strategy based on machine learning. Therefore, the thesis serves as a continuation of the work done by Prochazka et al. (2019) and also enriches the already existing literature on spatial efficiency in the Capesize sector. Our findings are relevant for both vessel operators and researchers alike, as it gives an indication of spatial efficiency and assesses whether further work should be done to test the effectiveness of implementing a similar approach in real life.

---

<sup>3</sup>ARIMA, VAR and VECM

### 3 Data

The sample data is derived from multiple time series spanning from January 2011 to December 2019. All observations are processed to be represented on a daily frequency, aiming to reflect the market situation on a given day throughout the time horizon. The choice of time span was motivated by two factors. Firstly, more explanatory variables were available when starting in 2011, instead of an earlier point in time. Secondly, by going further back than 2011, we would include the shipping boom prior to, and the crash after, the 2008 financial crisis. Similarly to Prochazka et al. (2019), we assume these freight rates not to be repeated, and that the future will be more like what has been observed after the crises.

The dataset serves two purposes. Firstly, we establish optimal route selections throughout time using Capesize earnings for the respective routes, forming a historical optimal chartering strategy. Secondly, explanatory variables are used in addition to past earnings to estimate a future chartering strategy based on prior, optimal decisions and the related market situation. The data is mainly derived from the Clarkson Shipping Intelligence Network, complemented with macroeconomic variables from Thomson Reuters Datastream.

To evaluate regional price differences between the Capesize routes, we use tripcharter earnings as reported by Clarkson Research. The earnings are based on 2010 Capesize vessels, from Tubarao (Brazil) and Western Australia to Quingdao (China), respectively.<sup>4</sup> These earnings functions as an indicator of the estimated daily earnings for a vessel operating on the specific routes, implied by the current spot freight rates (Research, 2015). Note that the vessel is assumed to return to the original point of origin as the trip is completed. Therefore, the reported earnings also take into account the ballast distance from Quingdao.

The dataset, including the route earnings, contains 69 variables in total, with various sampling frequencies. The additional explanatory variables used in the predictive analysis, are carefully selected to adequately explain the supply/demand dynamics for Capesize vessels, and the shipping industry in general. Table 3.1 depicts an abstract of ten of the variables, their relevancy in the prediction of route choices, as well as their sampling

---

<sup>4</sup>See A2.1 for further trip details

frequency and source.

**Table 3.1:** Selection of explanatory variables used to predict future trip choices (10 out of 69 total variables)

Name	Description	Relevancy	Frequency	Source
Iron_Ore_Price	Iron Ore 62% Fe, CFR China (TSI) Futures Settlements	Demand, Seaborn Commodity Trade	Daily	Thompson Reuters Datastream
USYield_10yrs	US 10-year Treasury Yield	Demand, World Economy Indicator	Daily	Thompson Reuters Datastream
Rebar_Future	SHFE Rebar Commodity Future Continuation 1	Demand, Seaborn Commodity Trade	Daily	Thompson Reuters Datastream
Vale	Vale SA Stock Index (Global Mining)	Demand, Seaborn Commodity Trade	Daily	Thompson Reuters Datastream
USYUAN_FX_Spot	US Dollar/Chinese Yuan Offshore FX Spot Rate	Demand & Supply, World Economy Indicator/ Local Costs	Daily	Thompson Reuters Datastream
Newbuild_Price	176-180k Capesize Bulkcarrier Newbuilding Prices, \$m	Supply, Shipbuilding Production	Weekly	Clarkson Research
MGO_Singapore	MGO Bunker Prices, Singapore \$/Tonnes	Supply, Transportation Costs	Weekly	Clarkson Research
India_Scrap	India Scrap Prices (Capesize) \$/td	Supply, Scrapping and Losses	Weekly	Clarkson Research
Port_Congestion	Capesize Port Congestion as % of Capesize Fleet	Supply, Fleet Productivity	Monthly	Clarkson Research
Avg_Scrap_Age	Capesize Demolition - Average Age, Years	Supply, Scrapping and Losses	Monthly	Clarkson Research

As seen from Table 3.1, the relevancy of the variables are considered according to their effect on the market dynamics. Because of the significant number of variables, this discussion will focus on the different groups of variables and what they aim to explain, rather than going into detail on each individually.

Firstly, we have included variables to capture the seaborne commodity trade dynamics, as explained by Stopford (2009), such as iron ore and coal prices. Long-term commodity price trends give an indication on economic activity, thus affecting the volume of vessels employed to carry such commodities. Including raw materials, like `Iron_Ore_Price` and `Coal_Price`, aims to capture the commodity trade specifics for Capesize vessels, as well as the lagged relationship between commodity and freight rates. The lagged relationship is especially important; according to Stopford (2009), seaborne trade has a significant lag between decisions and implementation. For Capesize commodities, this can be seen in how China tends to stockpile iron and coal and withhold imports until prices decrease. Furthermore, Tsioumas and Papadimitriou (2018) found a bidirectional lead-lag

relationship between the BCI index, and iron ore and coal prices, providing statistical evidence of this relationship. In addition to the raw materials, metallurgical products such as rebar and coil futures are included to see if more product-specific derivatives can provide insight into the seaborne trade dynamics.

Exchange rates are included to capture the relative fluctuations between economies, while also functioning as a proxy for local costs. Consequently, exchange rates for Brazil, China and Australia are included, such as `USDAUS_Fx_Spot`, `USDYUAN_Fx_Spot` and `USDBRL_Fx_Spot`. Stopford (2009) argues that unit costs vary proportionately with exchange rates as the main currency of the shipping industry is USD. Shipyards are especially vulnerable to exchange rate fluctuations, affecting the world fleet dynamics through the newbuilding market. Short-term shipping cycle peaks are identified by a higher volume of newbuilding orders, where cash flows out of the shipping industry because shipyards pay for materials and labor. Besides providing information on the liquidity in the Capesize market, shipyard specific exchange rates also aims to capture some of the supply dynamics of the world fleet. Consequently, `USD/JPY` and `USD/KRW` are included, as Japan and South Korea produce about two-thirds of the world's ships (Stopford, 2009).

To complement the supply characteristics captured by the newbuilding market, we introduce variables defining the demolition dynamics. An active demolition market could reflect a *trough* period, as minimal freight rates cause financial pressure, where old ships fall to scrap price. Scrap prices, such as `India_Scrap`, in addition to `Avg_Scrap_Age` aims to reflect the scrapping decisions done by decision-makers.<sup>5</sup> While low freight rates may trigger the demolition of older ships, shipowners are still reluctant to scrap vessels considering future expectations. Therefore, scrapping dynamics is considered more of a strategical process relative to expected future earnings (Tvedt, 2003). In addition, to reflect market expectations, scrap prices especially are included as world fleet capacity proxies. Higher rates in the demolition market functions as an incentive for carriers to scrap vessels, adjusting the capacity of the world fleet.

Transportation costs are mainly accounted for by including bunker cost variables. Besides influencing the vessel earnings directly, bunker costs also indirectly affect the market supply of ships. Higher fuel cost arguably increases supply costs, in turn causing higher

---

<sup>5</sup>Other scrap prices include; Pakistan, and Bangladesh, defining the majority of the demolition market.

freight rates due to high demand relative to market supply. As shipowners and operators aim to maximize profit, fuel optimization is essential, especially when demand exceeds supply. To create additional supply to cover the demand, operators could speed up their fleet to increase capacity, increasing fuel consumption. Consequently, fuel prices would determine the point where increasing the speed is no longer profitable.

Additionally, `Port_Congestion` is also included as a variable reflecting operational efficiency and to account for supply fluctuations. In general, congestion causes vessels to operate less efficiently, where schedule unreliability could affect fuel consumption. Furthermore, Stopford (2009) argues port congestion to give indications of the market cycle through the supply/demand relationship and consequently affecting the freight rates.

As time-lag plays a significant role in understanding the dynamics of the shipping market, we need to consider the lead-lag relationship between the variables. The methodology suggested, time-series classification, solves this issue to an extent, by using explanatory variables lagged through time. Consequently, the dataset is constructed with lagged versions of the explanatory variables to capture how historical decisions influence the current market situation. The representation of lagged relationships in a time-series classification will be further explained throughout the methodology.

Lastly, some assumptions are made to the raw explanatory data as they are sampled with multiple frequencies. Firstly, due to the problem framing in this thesis, each variable should be represented on a daily basis to reflect daily optimal decisions. To account for the weekly reported earnings, we are assuming the earnings for each week to represent a mean approximation for each day. The interval between each data point needed to fill the remaining days is substantial, meaning interpolation, in general, causes increased bias. Furthermore, an assumption of a linear process between data points, such as with linear interpolation, removes the effect of random shocks and could result in incorrect earnings estimates and unreliable market expectations. Therefore, we consider a mean approximation to be the most realistic approach. The same method is used for variables reported monthly, treating these observations as integers rather than trying to fit a continuous pattern with potential bias.

Regarding variables sampled on business days, however, we use a simple linear interpolation. Accordingly, we assume the missing data, especially over the weekends, to be close to



the bordering data points. Most of the business day data is related to financial fixtures, where volatility is reduced throughout the weekends, due to no trading (Sutherland et al., 2013). Therefore, the assumption of similarity in bordering values seems logical. On the other hand, this might lead to an underestimation of volatility within the series (Gençay et al., 2001). Alternatively, the series could be considered using no interpolation at all, reducing the uncertainty, but decreasing the number of observations significantly, as well as the possible explanatory variables. Ultimately, the treatment of missing values is based on a trade-off between bias and the problem framing, where the approach used follows common principles within time series analysis and machine learning practices (Hyndman and Athanasopoulos, 2018).

## 4 Machine Learning Theory

### 4.1 Walk Forward Validation

The models will be validated using walk-forward validation. This methodology involves incorporating new information as it becomes available, one time-step at a time. The flow of information for each time step is decided by the *window size*, which can either be static (sliding) or dynamic (expanding). The procedure can be explained as a *rolling forecast*, where a model is trained at the beginning of the time series until time step  $t$ . The model creates predictions for  $t + 1$  and is validated against the known value. For time step  $t + 2$ , the known value from  $t + 1$  is included in the window, and the process is repeated.

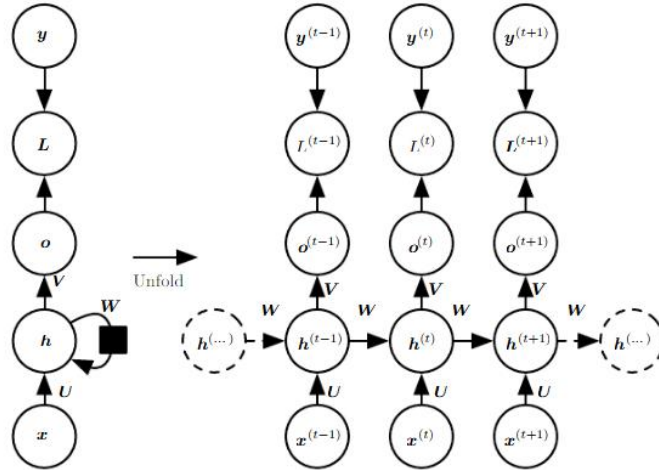
### 4.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are types of networks well suited for sequential-data processing such as time series. Different from traditional feed-forward networks, RNNs have the ability to interpret observations as dependent on each other, by storing information throughout the process. Whereas feed-forward networks have a one-way flow of information throughout the neurons, RNNs are extended to include feedback connections where information cycles back into the network. These cycles enable the network to have memory, where the neurons have different *states*. Mathematically, the state of the neurons can be defined as:

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta), \quad (4.1)$$

where  $h^{(t)}$  represents the state. Equation 4.1 depicts how the system is recurrent as  $h^{(t)}$  is defined by the previous version of itself,  $h^{(t-1)}$ , thus obtaining information from the prior state (Goodfellow et al., 2016). This cyclical behaviour of RNNs can also be illustrated by unfolding each iteration of the network, as seen in Figure 4.1.

One key aspect of RNNs, which is easily illustrated in Figure 4.1, is how the weight matrix,  $W$ , is used repeatedly throughout time. This eventually results in one of the implications of RNN architecture, the *exploding and vanishing gradient problem*. Exploding gradients



**Figure 4.1:** Output ( $o$ ) is produced given input  $x$  at time  $t$ . Loss  $L$  measures the error of  $o$  when compared to target  $y$ . The input-to-hidden connection weights,  $U$ , hidden-to-hidden recurrent connection weights,  $W$  and the hidden-to-output connection weights,  $V$ . Reprinted from "Deep Learning", Goodfellow et al., 2016, Ch.10, MIT Press.

make learning unstable, while the vanishing gradients make it almost impossible to know the direction of the parameters to improve the cost function (Goodfellow et al., 2016). To overcome these difficulties, this thesis will explore special, *gated*, RNNs which address the gradient problem, namely Long-Short-Term-Memory (LSTM).

### 4.2.1 Long-Short-Term-Memory

Proposed by Hochreiter and Schmidhuber (1997), LSTM overcomes the vanishing and exploding gradient problem by introducing three gated cells,  $f_t, i_t, o_t$ , controlling the information flow for each state unit.

$$\begin{pmatrix} f_t \\ i_t \\ g_t \\ o_t \end{pmatrix} = \begin{pmatrix} \sigma(W_f[x_t, h_{t-1}] + b_f) \\ \sigma(W_i[x_t, h_{t-1}] + b_i) \\ \tanh(W_g[x_t, h_{t-1}] + b_g) \\ \sigma(W_o[x_t, h_{t-1}] + b_o) \end{pmatrix} \quad (4.2)$$

The forget gate,  $f_t$ , sets the old states to zero once all information from a feature has been used, thus deciding what information to remove from the state. This is controlled via a sigmoid unit, setting the weights between 0 and 1. The input gate,  $i_t$ , decides which values in the cell state should be update in a similar fashion. Combining the forget and

input gate, the internal state of the cell,  $c_t$ , is given:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \quad (4.3)$$

where  $g_t$  represents the vector of cell updates, seen in Equation 4.2

Lastly, the output  $o_t$  is based on the information encoded in the cell state, but will be filtered through the output gate using the same sigmoid activation as well as a *tanh* layer to control the output values from the network:

$$h_t = o_t \cdot \tanh(c_t) \quad (4.4)$$

When training RNNs, such as LSTM, the aim is to minimize the loss,  $L$ , as depicted in Figure 4.1. This is achieved using optimization algorithms that are used to adjust the network weights to get the optimal performance. The weight tuning in this thesis will utilize the *Adaptive Moment Estimation (Adam) optimizer*, introduced by Kingma and Ba (2014), which has been shown to be a robust stochastic optimization algorithm (Kingma and Ba, 2014). Adam uses separate learning rates for all weights in the model, meaning the learning rate is adapted separately as learning progresses within the network.

Overfitting to training data is a major concern when training neural network models. One approach to help avoid overfitting is through feature selection. By applying Lasso Regularization (L1), a penalty term,  $\lambda$ , is introduced to both shrink coefficients and provide feature selection.

$$L = \frac{1}{2N} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 + \lambda \sum_{j=1}^N |w_j| \quad (4.5)$$

As seen from Equation 4.5, L1 regularization uses the sum of the absolute value of the coefficients as a penalty parameter, forcing some of the network's weights to be zero.

Dropout regularization is a classic technique used to reduce overfitting by simply dropping random units in the network while training. However, the use of dropout in recurrent neural network architectures has been proven to reduce the memorization ability of RNNs (Jozefowicz et al., 2015; Gal and Ghahramani, 2016). Gal and Ghahramani (2015) propose

a way to apply appropriate dropout to RNNs by applying the same dropout mask to every time steps, instead of a dropout mask that varies from time step to time step. Recalling the computation of the LSTM gates from Equation 4.2, Equation 4.6 exemplifies how Gal and Ghahramani (2015) proposes dropout for RNNs.

$$\begin{pmatrix} f_t \\ i_t \\ g_t \\ o_t \end{pmatrix} = \begin{pmatrix} \sigma(W_f[x_t, d(h_{t-1})] + b_f) \\ \sigma(W_i[x_t, d(h_{t-1})] + b_i) \\ \tanh(W_g[x_t, d(h_{t-1})] + b_g) \\ \sigma(W_o[x_t, d(h_{t-1})] + b_o) \end{pmatrix} \quad (4.6)$$

Recurrent dropout as described by Gal and Ghahramani (2015) is available through Keras (for documentation, see Chollet et al. (2015)). This thesis will explore recurrent dropout rates of 25% and 50% in accordance with Semeniuta et al. (2016), in order to identify and finalize a suitable model to predict optimal route decisions.

To formulate the output to suit a classification problem, *softmax* activation is used in the output layer. Softmax takes the numeric output from the network and turns it into probabilities for each class.

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (4.7)$$

Equation 4.7 illustrates this by taking the exponent of each output and normalizing the output summing the exponents. The function further ensures that the sum of probabilities add up to 1, which is preferable for multi-class-classification problems, and enables the network to be expanded to include multiple routes for further research (Dunne and Campbell, 1997).

LSTMs have been used to great effect, learning long-term dependencies, and how information from previously observed data is relevant to future sequential observations, obtaining state-of-the-art performance on sequence processing tasks. These capabilities have been used to solve complex natural language processing, such as neural machine translation and speech recognition, learning complex interrelationships between words.

## 4.3 Extreme Gradient Boosting

The extreme gradient boosting (XGboost) algorithm has become one of the most used machine learning algorithms and consistently perform competitively in machine learning competitions. XGBoost was developed by Cho et al. (2014) and builds on the concept of traditional boosting. Traditional boosting involves combining several weak learners to ensemble a robust predictive model (Friedman, 2001). The most common learner to boost is the tree algorithm and boosting involves fitting them sequentially on the residuals of the previous trees. Therefore, each new tree is fit to pick up variance in the data not already picked up by previous trees. This process is repeated until it reaches the predetermined maximum number of trees. According to Cho et al. (2014), the main aspect of XGBoost is how additional regularization parameters are included to prevent overfitting. Thus, the model formalization is more regularized and gives better results. Nielsen (2016) concluded that the effectiveness of the algorithm stems from the ability to operate with varying flexibility, depending on the location in feature space. Hence, incorporating the bias-variance trade-off as a core aspect in the model fitting. The increased flexibility compared to traditional gradient boosting comes from the fact that XGBoost deploys a Newton method in function space, while traditional gradient boosting has been based on gradient decent algorithms (Nielsen, 2016).

The XGBoost method uses  $K$  additive functions to make predictions:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^K f_k(x_i) \quad f_k \in F \quad (4.8)$$

where  $F$  is the space of regression trees and each  $f_k$  corresponds to an individual tree (Cho et al., 2014). Predictions are made using the individual trees and weighting the output from each output leaf in the  $K$  amount of function by the leaf weight,  $w$ . The following regularized objective function is minimized to obtain the functions used by the model  $t$ :

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (4.9)$$

Where,  $l$  represents the loss function, measuring the difference between the actual and

predicted value, with *Omega* included to penalize complex models.

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (4.10)$$

Where,  $T$  represents the number of leaves, and  $\gamma$  and  $\lambda$  are regularization parameters included to prevent overfitting (Cho et al., 2014).  $\gamma$  is a minimum loss required for a tree to split; therefore, a high gamma will lead to more shallow trees and less overfitting at the cost of not being able to pick up patterns in the data.  $\lambda$  is the L2 regularization parameter, similar to ridge regression. In addition to these parameters, XGBoost offers multiple other hyperparameters that must be tuned for each dataset. The learning rate decides the shrinkage that is done at every sequential step and is closely related to the other boosting parameter. In general, complex relationships will demand a higher number of trees and lower learning rate to compensate and prevent overfitting. The *min\_child\_weight* is like  $\gamma$ , a tree parameter, and determines the minimum amount of observations that must be in a terminal node for it to be included. Column sampling and row sampling are also available randomization techniques that can enhance the overall model by decorrelating trees, thus reducing the overall variance (Nielsen, 2016; Cho et al., 2014; James et al., 2013).

The values of the hyperparameters can drastically change the effectiveness of the XGBoost algorithm (Cho et al., 2014). Too much regularization and the fit will not pick up trends and patterns from the training data. On the other hand, not enough regularization will cause overfitting and poor generalization (James et al., 2013). To find the optimal parameter values, one can use validation and a grid search of potential values. Ideally, one would use a large grid to test as many combinations as possible. However, the high number of parameters quickly becomes computationally demanding when there are many variables and observations. Therefore, we have decided to follow an alternative approach by tuning the learning rate, the number of trees and tree dept first, as parameters should have the most significant effect on the result (Cho et al., 2014). We will then tune the remaining parameters before we ultimately perform a recheck of the learning rate and the number of trees.

## 5 Methodology

This chapter contains three major parts, outlining the methodology used in the thesis. Firstly, the route optimization method is presented. In this section, we will explain how the upper bound is calculated with perfect foresight and how the benchmark is calculated using a coin strategy. Secondly, data processing and model training is explained. This part focuses on how the optimization output is transformed and formatted to predict route decisions using machine learning algorithms. Ultimately, the evaluation section will explain how the predictions are evaluated and used to calculate expected earnings.

### 5.1 Chartering Strategy

#### 5.1.1 Route optimization

Prochazka et al. (2019) proposed a version of the minimum-cost flow problem to optimize vessel allocation within a limited planning horizon. The version presented by Prochazka et al. (2019) assumes perfect knowledge about prices to establish an upper bound for a realistic routing strategy.<sup>6</sup> The optimization model in this thesis is largely based on the work of Prochazka et al. (2019) adopted to suite a two-route scenario, where,  $r_{jt}$ , denotes the tripcharter rates for a roundtrip,  $j$ , with duration from  $\tau_j^{\min}$  to  $\tau_j^{\max}$ , within the planning horizon,  $T$ .<sup>7</sup>

The following optimization model is used to compute the optimal vessel allocations by maximizing expected earnings within the specified planning horizon:

$$\max_f \sum_{jt} e_{jt} f_{jt} \quad (5.1)$$

s.t

$$R_t = \sum_j f_{jt} \quad \forall t \quad (5.2)$$

---

<sup>6</sup>For a more in-depth explanation of this version of the minimum-cost flow problem see Prochazka et al. (2019)

<sup>7</sup>When comparing to the original paper, note that the version presented in this thesis assumes only one origin,  $i$ , and two destinations,  $j$ , thus  $i$  is omitted in Equation 5.1 - 5.6



$$R_{jt} = R_{jt}^0 + \sum_{s=t-\tau_j^{max}}^{t-\tau_j^{min}} \pi_j f_{js} \quad \forall j, t \quad (5.3)$$

where the flow,  $f_{jt}$ , of a ship is computed as a function of the optimal decision,  $x_j$  and the capacity  $R_t$ :

$$f_{jt} = x_{jt} R_t \quad (5.4)$$

and  $\pi_j$  defines the probability of a trip duration:

$$\pi_j = \frac{1}{\tau_j^{max} - \tau_j^{min} + 1} \quad (5.5)$$

while  $e_{jt}$  defines the expected earnings for a trip to destination  $j$  at time,  $t$ , given all possible trip durations  $s$ :

$$e_{jt} = \sum_{s=\tau_j^{min}}^{\min\{\tau_j^{max}, T-t\}} \pi_j r_{jt} s \quad (5.6)$$

Solving the optimization model outputs the optimal chartering strategy stored as a binary solution in  $x_{jt}$ . For each time period,  $t$ , one trip could yield higher expected earnings than the other; thus,  $x_{jt}$  can be interpreted as a signal as to which route is the most profitable based on perfect foresight of the future.

The optimal chartering strategy can be applied to recreate a trading flow and calculate earnings achieved by taking the optimal route. The calculations of the earnings in each time period is performed in similar fashion as in Prochazka et al. (2019), and the cumulative earnings will be calculated for evaluation.

### 5.1.2 Benchmark Method

In addition to the upper bound of possible earnings, represented by the perfect foresight results, we wish to compare our predicted routes with a benchmark strategy. Optimally, the benchmark would be based on the real trade flows on each route, as used by Adland et al. (2017), or the earnings achieved by an actual vessel sailing the selected routes in the same period. However, because we do not have data available to find any of these

two possible benchmarks, we will apply a *coin strategy* similar to the approach taken by Prochazka et al. (2019). The coin strategy will be based solely on preset probabilities of taking each route, not using any foresight of rates or knowledge of historical values. Adland et al. (2017) found that by using probabilities proportional to real trade flows, one could simulate the average earnings of the market participants. Prochazka et al. (2019) further found that using probabilities, such that the expected utilization of each route is equal, gave earnings that were not significantly different. Therefore, we can apply the coin strategy as a proxy for average earnings achieved by the market participants. Hence, the benchmark strategy represents a lower bound to the earnings one should, at a minimum, meet by predicting optimal route selection. This implies that our models must outperform the coin strategy to be considered adequate.

To simulate the earnings achieved by the coin method, the optimal destination,  $x_j$ , is set to the probability of going to each destination  $j$ . The calculations of the resulting expected earnings can then be performed by taking the aggregate of all alternatives with a non-zero probability such that a route is taken in time  $t$ .<sup>8</sup>

## 5.2 Machine Learning

### 5.2.1 Train-Test Split

To fit and evaluate performance on out-of-sample data, the dataset is divided into a training and test set. The sample ratio between the train and test set is considered a trade-off evaluation, as less data in the training set results in a greater variance in the parameter estimates, and vice-versa. The trade-off evaluation, therefore, comes down to the nature of the problem. In addition to giving proper destination estimates, this thesis also aims to show how routing optimization affects cumulative earnings over time. To properly identify the effect of an optimal chartering strategy, the testing period needs to be of a certain size, as each roundtrip last a significant number of days. For this reason, a test set containing 1,095 days is used, representing three years of chartering. Keeping the last 1,095 days as out-of-sample gives a training set of 2,178 days.

To compute the optimal destinations for the training set, we apply the optimization

---

<sup>8</sup>For a more in-depth explanation of the difference in coin strategy calculations problem see Prochazka et al. (2019)

algorithm described in Section 5.1.1 on the training subset of data. Optimal destinations are calculated as a function of expected earnings; thus, we do not include the testing data in the training computation to ensure no data leakage from future observations. Furthermore, we treat the testing set as a continuation of the training set, where the optimization is conducted on the entirety of the data. We further extract the last 1,095 optimal destinations, for out-of-sample testing.<sup>9</sup>

## 5.2.2 Supervised Learning Data Representation

The nature of the problem requires us to formulate the data as a supervised learning problem, where each window of feature data represent one *class*.<sup>10</sup> To properly process the data, the sliding window approach is applied, where the input data is divided into separate fixed window containing lagged features. The shape of the data can be described in terms of the number of samples, the number of time steps and the number of explanatory features. For the training set, this approach returns a three-dimensional array,  $[2128, 50, 69]$ , containing 50 time steps of the 69 explanatory variables, with 2128 samples.<sup>11</sup> Furthermore, the test set would be summarized as the remaining samples;  $[1095, 50, 69]$ . Essentially, meaning that for each sample, we have one window containing  $(50 \cdot 69)$  3450 explanatory variables, with the 69 features for the 50 previous days.

Consequently, each window of features is associated with a specific optimal trip according to the routing optimization described in Section 5.1.1. Following this approach, we have a binary response variable explained by a window of historical features. This approach ensures that there is no data leakage. The models are only trained and validated on historical observations, whereas the actual response variable from the test set is not made available to the model before evaluating performance.

The different scales of the explanatory variables may cause slow training and cause large, unstable weights, especially for the LSTM model. To account for the different scales, we normalize the explanatory variables when pre-processing data, thus presenting all

---

<sup>9</sup>Three year testing horizon,  $(x_{jt} \quad \forall t \geq 1095)$

<sup>10</sup>"Class" in classification term simply defines the different outcomes of the classification. In this thesis we have two different trips, thus two different classes.

<sup>11</sup>Due to 50 lags used as explanatory variables, the 50 first optimal trips are removed, explaining the reduction from 2,178 to 2,128 samples.

variables on the same scale (Brownlee, 2018).<sup>12</sup>

### 5.2.3 Model training

The lagged data will be used to train, fit, and make predictions with both machine learning methods. Hyperparameter tuning will be performed using walk-forward-validation solely on the training data to prevent data leakage. After the optimal parameter values are found, the models will be refit on the entire training set to include all available information throughout the training period. After predicting the initial year, we will refit the models yearly without retraining due to a lack of computing power. The route optimization will be re-done and included in the refitting to prevent data leakage. Thus, our method assumes that the decision-maker trains the models before each three-year period and then refits it every year to capture variable levels.

To make predictions, the explanatory variables in the test set is used as input. This means that for each day, we make predictions using historical information from the previous 50 days. The intuition behind this method is that if a decision-maker, at day  $t$ , is to make a trip decision, all information prior to that day is available to base the said decision on. The output of the predictions will be a vector of integer values representing the predicted optimal destination each day.

## 5.3 Evaluation

To evaluate the predictions, we will first compute the out-of-sample accuracy and Cohen's Kappa. Accuracy is found by dividing the number of correct predictions by the total number of predictions. The accuracy generally gives a good measurement of how many observations that are predicted correctly but can be artificially high in the case of imbalanced classes (James et al., 2013). Therefore, to account for imbalanced classes, we will also look at Cohen's Kappa statistic. The Kappa statistic takes into account the possibility of the agreement between predictions and reference occurring by chance (Cohen, 1960), comparing the observed accuracy,  $P_o$ , and the expected accuracy,  $P_e$ :

---

<sup>12</sup>Z-score normalization is used for data scaling

$$k = \frac{P_o + P_e}{1 - P_e} \quad (5.7)$$

Where  $P_e$ , is defined as the accuracy achievable from any random classifier based on the confusion matrix. Ultimately, Cohen's Kappa simply measures the accuracy of our classifier relative to a classifier randomly guessing based on the frequency of each class.

In addition to the aforementioned accuracy measures, the predictions will be evaluated based on their ability to capture differences in expected earnings between the two trips. To quantify the relative difference, we calculate the value of trip switching,  $S_{jt}$ , based on the results from the optimization model presented in Section 5.1.

$$S_{jt} = Z_{it} - Z_{jt} \quad i \neq j \quad (5.8)$$

where  $Z_{jt}$  is calculated using the expected earnings,  $V_t$ , from time  $t + s$ , to the end of the planning horizon.  $Z_{jt}$  will represent the expected earnings for a trip to destination  $j$ , at time  $t$  to the end of the planning horizon:

$$Z_{jt} = \sum_{s=\tau_j^{\min}}^{\tau_j^{\max}} \pi_j (r_{jt}s + V_{t+s}) \quad (5.9)$$

where  $s$  represents all possible trip durations to destination  $j$ .

$S_{Brazil,t}$ , will, in our two-trip case, give the difference in earnings if the Australia roundtrip is taken instead of the Brazil roundtrip. Therefore, a positive  $S_{Brazil,t}$  will indicate that a higher expected earning is achieved by taking the Australia trip, while a negative  $S_{Brazil,t}$  suggests the opposite. The magnitude of the absolute value of  $S_{jt}$  functions as an indicator describing the importance of taking the optimal decision each day.

The prediction output will be used to calculate expected earnings by using the destinations in a three-year chartering simulation. Cumulative earnings achieved with the machine learning prediction methods will then be compared to the earnings achieved with perfect foresight and the benchmark strategy. The cumulative earnings are used instead of daily earnings because the optimal route is found by maximizing the earnings at the end of

the planning horizon.<sup>13</sup> Therefore, the optimal trip for a given day might not yield the highest short-term earnings, as the earnings from one period could be sacrificed for better future positioning. Consequentially, the current daily earnings are not as important as the total cumulative earnings and average daily earnings over the entire period.

To further evaluate the methods, a fleet of 9 additional ships will be included. Each vessel will start at a different point in time, but ending on the same day. Therefore, the level off cumulative profit cannot be used to compare the results between the different vessels. Thus, we will evaluate the performance relative to the benchmark. Expanding the analysis to include different starting points will, the results are more robust as it reduces the likelihood of the results being due to chance.

---

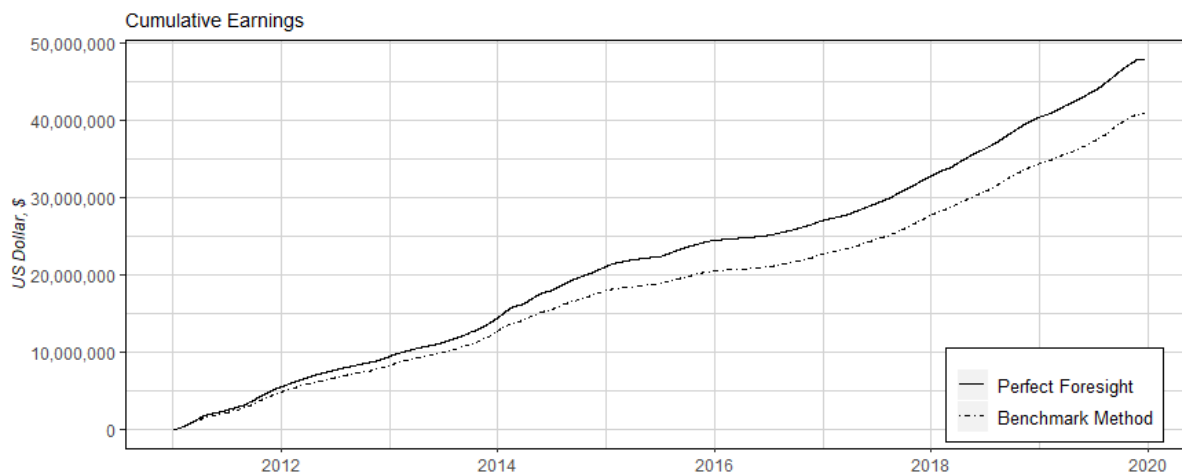
<sup>13</sup>An alternative would be to use average daily earnings over the entire period. However, it would give similar results to cumulative earnings.

## 6 Results and Discussion

This chapter describes the research findings of the thesis. The first section presents the result from applying the perfect foresight method and benchmark method for the entire period. Secondly, the LSTM neural networks and XGBoost models are presented, providing a brief discussion on the hyperparameters used. Next, the prediction models are evaluated using both accuracy measures and the value of switch. Lastly, the earnings achieved by using the predicted optimal routes are compared to the upper and lower bounds to evaluate the effectiveness of the methods as a decision-support system.

### 6.1 Perfect Foresight and Benchmark Results

The first step in evaluating the results is to assess if it is possible to outperform the market average by estimating the earnings for the entire time horizon. In accordance with Prochazka et al. (2019), the upper bound is established using perfect foresight and reflects the earnings obtained by optimal vessel positioning. Furthermore, a simulation using the benchmark strategy is conducted to represent average earnings for market participants. The cumulative earnings for the respective chartering strategies can be seen in Figure 6.1.



**Figure 6.1:** Comparison of the perfect foresight and coin strategy cumulative earnings over the entire time horizon, October 2011 to December 2019.

As expected, based on the results presented by Prochazka et al. (2019), the perfect foresight simulation outperforms the benchmark strategy achieving 17.33% higher cumulative earnings at the end of the period. The results show that it is possible to take advantage of

relative differences in freight rates to extract economic gains, at least under the assumption of perfect foresight. Therefore, the machine learning methods can also outperform the assumed market average if they are able to predict the optimal decisions correctly.

## 6.2 Predictive Models and Model Evaluation

The following subsection highlights the finalized models, their specifications, and the hyperparameters. As described in the methodology section, we conduct a grid search to tune the hyperparameters for both models. The results of the grid search, as well as the achieved out-of-sample accuracy, can be seen in Table 6.1.

**Table 6.1:** Finalized models; an overview of hyperparameters and out-of-sample prediction accuracy

LSTM		XGBoost	
Hyperparameter	Value/type	Hyperparameter	Value/type
Number of hidden neurons layer 1	200	Number of trees	1300
Number of hidden neurons layer 2	200	Maximum Tree Depth	13
Classifier	Softmax	Learning Rate	0.01
Optimizer	Adam	Column subsampling	0.2
Learning rate	0.0005	Gamma	5
Number of epochs	2000	Minimum Child Weight	3
Recurrent Dropout	0.50	Row subsampling	1
L1 Regularization	0.08		
<b>Out-of-sample accuracy:</b>	71.49%		68.07%
<b>Cohen's kappa:</b>	0.2975		0.3032

As depicted in Table 6.1, the network configuration for the LSTM architecture is quite complex, as it is a two-layered LSTM network with a significant number of iterations. To account for the complexity of the architecture and avoid overfitting, *recurrent dropout* is used, dropping 50% of the connections between the layers with each iteration. *L1 Regularization* reduces overfitting by penalizing coefficients and provides feature selection, which is well suited for this problem with a large number of features.

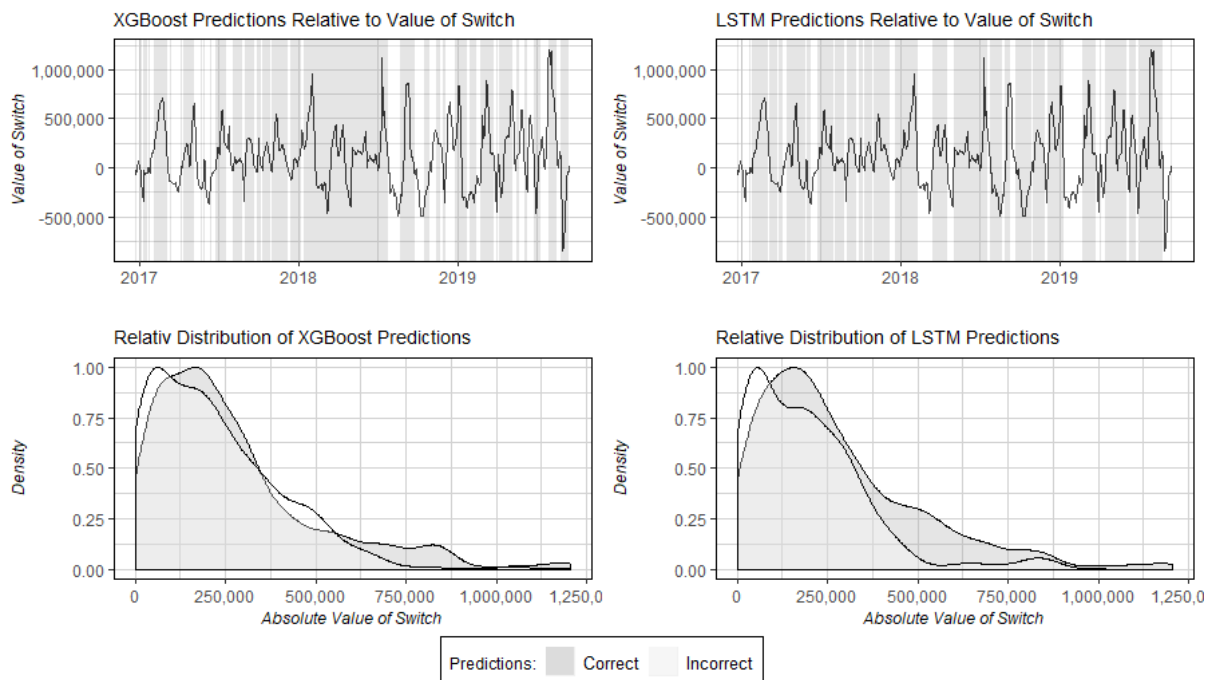
The final XGBoost hyperparameters imply complex relationships and patterns in the data. A high number of deep trees with a corresponding low *learning rate* is needed to pick up the information effectively. The validation also showed that a high amount of regularization was required to be able to reduce overfitting and generalize well onto new data. The use of *column subsampling* to decorrelate trees and reduce the power of a few



single explanatory variables was found the most effective. However, *gamma* and *minimum child weight* was also used to limit unnecessary tree depth, due to the relatively high maximum tree depth.

Both the LSTM and XGBoost models have a relatively high out-of-sample accuracy. However, the imbalanced classes make traditional accuracy somewhat misleading. The Kappa value accounts for the no-information rate and is used to control for imbalanced classes. The imbalance classes imply a small premium towards the Australian route, as this route is selected as the optimal route more often. While LSTM achieves the highest accuracy, the XGBoost model seems to outperform the LSTM model when considering the Kappa value. This can be explained by how XGBoost is more effective in correctly classifying the less common class, the Brazil trips (See confusion matrix A3).

To further investigate the predictions, we compare the predictions against the value of switching trips. The more the value of switch deviates from zero, the more costly it is to make an incorrect prediction. Therefore, we analyze the relationship between the correct and incorrect predictions, relative to the cost of making wrongful decisions.



**Figure 6.2:** Investigation of when the XGBoost and LSTM prediction are correct and incorrect compared to the value of switch at the time. The value of switch is given as the relative expected earnings, from the current time to the end of the planning horizon, if one goes to Australia instead of Brazil.

The time series plots in Figure 6.2 displays at what time periods the predictions are correct and incorrect along with the value of switch. Visually, both methods seem to classify the most extreme values correctly. However, the LSTM method appears more stable, while the XGBoost method's accuracy varies more with time. For example, the XGBoost predictions appear weak in mid-2017 and late 2018, but very accurate in early 2018.

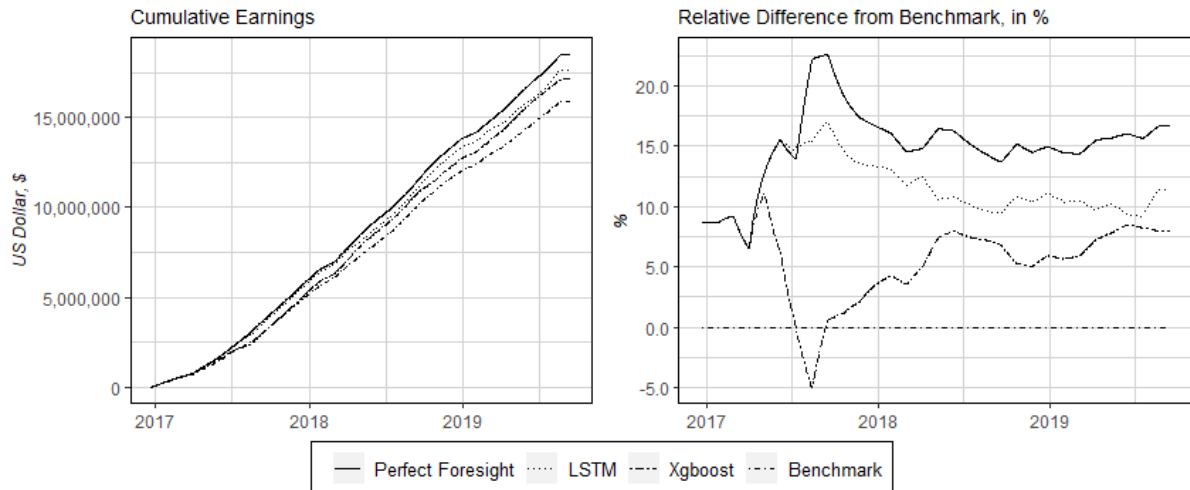
The density plot from Figure 6.2 visualizes the relative distribution of the absolute value of switch, categorized by whether the observations are predicted correctly or not. The distributions show that the models are in general better at correctly classifying optimal routes as the absolute switching value increases. This is a good sign, as we are able to avoid making wrongful decisions when the opportunity cost is (too) high. Most of the incorrect predictions are occurring with lower values of switch, which could reflect how the models are indifferent between the choices when the opportunity cost is minimal. The LSTM model appears better at correctly classifying observations with an absolute value of switch over 300,000. However, the XGBoost model performs better when the value of switch is higher than 750,000, yet the importance of this is questionable due to the small number of trips within this range. Therefore, it is interesting to look at the difference in the mean of the absolute switching values. The difference between the mean of correctly and incorrectly classified trips is tested statistically using a one-sided t-test.<sup>14</sup> The resulting low p-value for both methods indicate that we can reject the null hypothesis of equal mean, thus implying that the difference is statistically significant, verifying the visual analysis.

### 6.3 Expected Cumulative Earnings

To evaluate the models' effectiveness as a decision support system, we calculate the expected earnings achieved. The three-years of predicted decisions are used to simulate the expected earnings obtained by applying the machine learning models to select which route to take. Figure 6.3 depicts the cumulative earnings throughout the three-year period, compared to the benchmark and perfect foresight strategy

---

<sup>14</sup>Welch Two Sample t-test



**Figure 6.3:** Cumulative earnings from LSTM and XGBoost vessel routing, compared to coin and perfect foresight. Three year chartering period, December 2016 to December 2019.

From the cumulative wealth plot to the left in Figure 6.3, we can see that both the machine learning models outperform the benchmark strategy. As explained in Section 5.1.2, our benchmark strategy also functions as a proxy for the average earnings achieved by market participants. Therefore, by outperforming the benchmark, we demonstrate that machine learning methods can outperform the market average.

The plot on the right-hand side of Figure 6.3, portrays the relative earnings of the chartering strategies as a difference in percentage from the benchmark strategy. Again, we can see that both ML-methods outperform the benchmark strategy at the end of the time horizon. The perfect foresight method consistently has cumulative earnings that are around 17% higher than the benchmark strategy. The perfect foresight's relative outperformance of the benchmark strategy seems comparable to the findings by Prochazka et al. (2019), where they found perfect foresight to have 23% higher earnings in the Capesize market. A potential explanation for the difference in relative earnings could be that their testing period was 2009-16 and that they had different spatial dynamics due to different route selection.

Figure 6.3 reveals that the relative cumulative earnings at the end of the time period is about 11.4% higher with LSTM, and 7.9% higher with XGBoost, compared to the market average. Even though the earnings obtained at the end of the planning horizon are the most important, due to the nature of the optimization problem, the relative movements

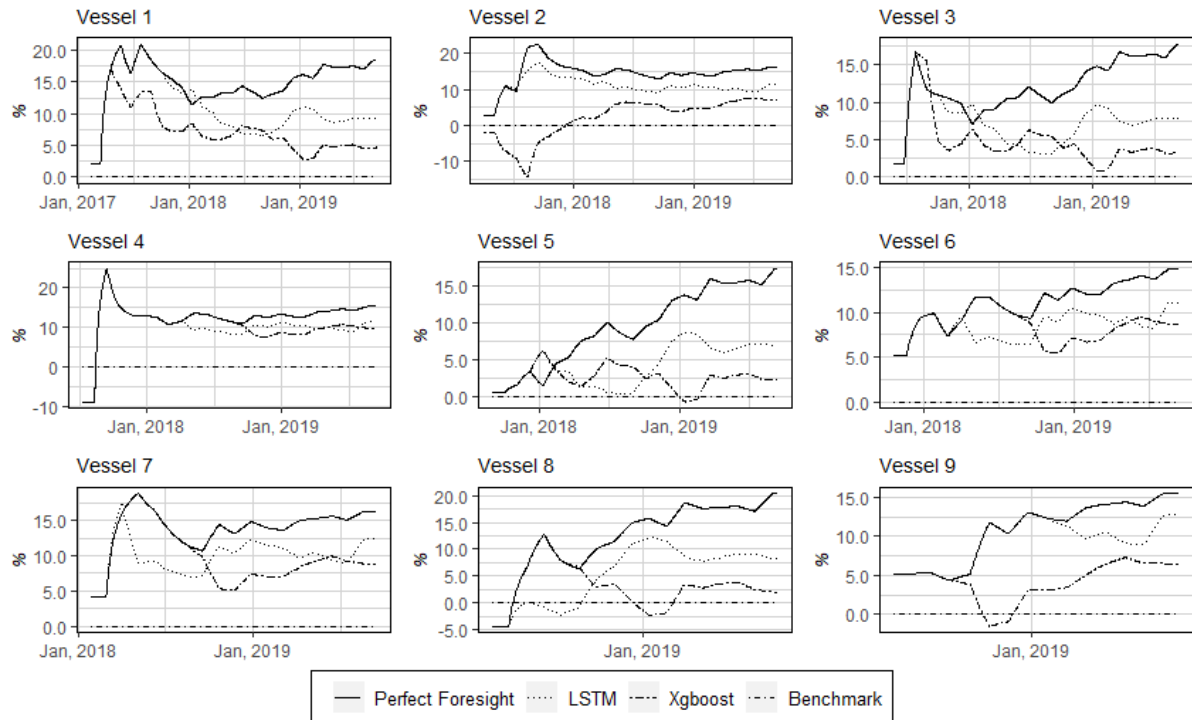
show some interesting trends. Firstly, the LSTM and perfect foresight earnings are the same until mid-2017 and seem to, in general, follow the same movements. However, the perfect foresight cumulative earnings are in a short period lower than the LSTM earnings, before drastically increasing. This movement demonstrates how the perfect foresight optimization can forgo short-term earnings to secure an improved vessel positioning to capitalize on increased freight rates. While the XGBoost earnings also start at the same level as the perfect foresight earnings, they quickly drop to below the market average. Thus, implying that the XGBoost model, unlike the LSTM, cannot pick up the information needed to make the optimal choices during this period. The XGBoost model has its strongest period from mid-2017 to 2018, which coincides with the period it had the best accuracy (see Figure 6.2). Overall, our results show that a decision-maker who applies our method and continuously updates the input data can outperform the assumed average earnings achieved in the market.

The results obtained by using our machine learning methods is somewhat comparable to the relative earnings achieved by Prochazka et al. (2019), using machine learning with limited foresight. They found that limited foresight gave between 10-25% gain relative to coin in the Capesize sector.<sup>15</sup> However, their problem specification and time period also made the relative outperformance with perfect foresight higher. A relatively higher outperformance by the perfect foresight method indicates there is more economic gain in the market. Therefore, the difference in perfect foresight gain can partly explain why the results are lower. Note that while the methods used are different, the comparison, nonetheless, gives an indication of the performance of the ML-models relative to limited foresight.

To evaluate the robustness of our results, we simulate the earnings of 9 additional vessels starting at different points in time. The resulting relative earnings to the perfect foresight method and benchmark method can be seen in Figure 6.4.

---

<sup>15</sup>See Prochazka et al. (2019), Figure 3 for the relative performance of limited foresight in the Capesize sector



**Figure 6.4:** Relative cumulative earnings compared to benchmark and perfect foresight, for a fleet of 9 additional ships, with different starting points in time.

In Figure 6.4, the earnings achieved by the individual vessel is depicted. The figure shows that the relative outperformance of the market is somewhat unstable, at least for the XGBoost method. The LSTM method seems to consistently result in about 10% higher earnings than the benchmark, while the XGBoost method achieves earnings from 2% to 10% higher. For the entire 10 vessel fleet, XGBoost in total performs 6.14% better than the market average, while the LSTM method outperforms the market with 10.95%. Therefore, the results show that both methods can outperform the market average, even when applied to a fleet. However, the perfect foresight method obtains 18.75% higher earnings than the market average. Thus, it is possible to achieve even higher earnings with the machine learning methods if one is able to increase the prediction accuracy.

To further quantify the performance, we can look at the average daily earnings in USD over the period. The difference in percent will, as explained in Section 5.3, be the same as the relative difference in cumulative earnings. The relative outperformance in USD per day will depend on the freight rates in the period. Therefore, the outperformance in USD per day will be different if one was to implement the approach in a time period with different freight rates. However, in our case, each ship using the perfect foresight method,

had average earnings of about 2,664 USD more than the market average per day. The LSTM method and XGBoost approach, on the other hand, had average daily earnings of about 1,667 USD and 1,010 USD more than the market average. Therefore, the average participant in the market could, according to our results, increase their yearly earning with about 600,000 USD per vessel in their fleet by using the LSTM approach.

Overall, of the two machine learning methods, the LSTM method seems more stable and achieves the highest cumulative earnings. Looking back at Figure 6.2, the more stable prediction accuracy over time can explain the less volatile performance by the LSTM method. The LSTM method's consistently higher earnings relative to XGBoost can be explained by the density plot in Figure 6.2. The density plot revealed that the LSTM method is better at correctly classifying trips with a high absolute value of switch.

## 6.4 Limitations and Further Research

The limitations of the approach taken in this paper mainly stem from the assumptions made in the optimization method and the benchmark used. A limitation in the optimization is that we assume the ship must always be sailing and is always able to get full cargo; this could be considered unlikely. Cargo is not necessarily available and could lead operators having to reposition the vessel at an expense. Furthermore, the constant availability of contracts through space and time could be considered unlikely, imposing some waiting time for vessels between contracts.

We see two main possibilities for further work done on this topic. Firstly, the optimization problem can be expanded to include more routes, or altered to incorporate different factors relevant to the chartering strategy. For example, one could include decision variables for speed, waiting days, refueling, and ship service.

Secondly, when it comes to predictive modeling, an interesting idea for further research is to test whether one could achieve better results by retraining the models more frequently. Even though this would be computationally demanding, it is reasonable to assume that in a real-life business application, one would want all information possible used in training. Thus, enabling the best decision making possible. Therefore, one could, for example, perform the optimization and retrain the predictive models daily. The increased training and decreased out-of-sample period should, in theory, lead to improved results. However,

the magnitude of the improvement is difficult to say anything conclusive about. Further work could also investigate the addition of more explanatory variables because, even though we have included the most promising variables in literature, there might be other variables that can increase the predictive power. For example, using the continuously growing amount of AIS data to find the port and route-specific variables could potentially improve the predictive power.

Another alternative approach would also be to forecast freight rates and use the forecasted rates as input in a predictive model trained following the limited foresight method applied by Prochazka et al. (2019). This method would however quickly become computationally demanding as it would require a high number of forecasts.<sup>16</sup>

---

<sup>16</sup>The forecasts would be used as input to predict the optimal decision. Therefore, one would need  $n \cdot t$  number of forecasts, where  $n$  is the number of days with foresight and  $t$  is the number of days in the testing period.

---

## 7 Conclusion

In this thesis, we have evaluated to what degree machine learning is applicable as a decision support system for optimal spot chartering for Capesize vessels. The approach presented aims to predict optimal future routes based on historical data. Accordingly, we establish an estimation of possible economic gains by learning from previous market trends and their respective optimal chartering strategy. The results obtained give an indication of whether machine learning techniques are suitable to support future chartering decisions.

We formulated a time-series classification problem using Long-Short-Term-Memory and Extreme Gradient Boosting to predict optimal future trip choices. The methodology was adopted to a two-trip scenario, predicting the optimal chartering strategy for the China-Brazil and China-Australia round trips. By classifying future trips daily, we were able to predict around 71% of the optimal trips correctly for a three-year chartering period. Controlling for imbalanced classed, a kappa of 0.29 and 0.30, was achieved by LSTM and XGBoost, respectively.

Furthermore, the predictions were evaluated using the value of geographical switching to analyze the implications of correct and incorrect predictions. The value of switch analysis suggests a statistically significant difference between the mean of the absolute value of switch for the correct and incorrect predictions. This indicates that the models perform better when there is a higher value to be captured by geographical switching. Consequently, implying that the model is, to some degree, able to identify and analyze geographical mispricing. This finding is important in itself and provides useful insights to both model evaluation and opportunities for future research.

Overall, our results show that by predicting optimal trip choices, one can achieve about 11% higher cumulative earnings relative to average earnings of market participants in the years 2017-2020. To further evaluate the robustness of these results, we calculate the cumulative earnings from ten different vessels, starting at different points in time. The stable 10-11% in relative cumulative earnings further strengthen the conclusion from the one vessel analysis, reducing the likelihood of randomness in the results. Subsequently, the results suggest that there is value to be captured by using historical data to make chartering decisions, which could further strengthen future outlooks of decision-makers in



the market.

The nature of the classification method enables our approach to be applicable for scenarios with multiple different round trip choices and can be expanded to include one-way trips by introducing ship positioning.

## References

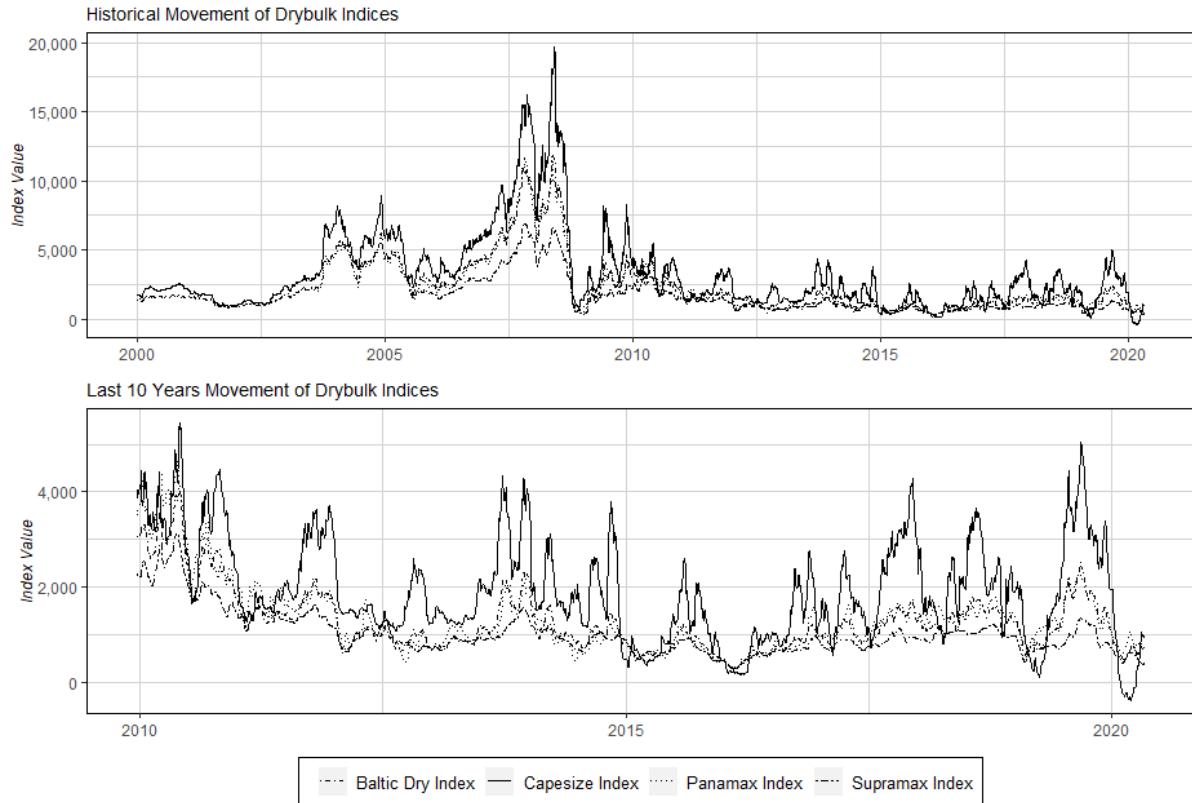
- Adland, R., Benth, F. E., and Koekebakker, S. (2018). Multivariate modeling and analysis of regional ocean freight rates. *Transportation Research Part E: Logistics and Transportation Review*, 113:194–221.
- Adland, R., Bjercknes, F., and Herje, C. (2017). Spatial efficiency in the bulk freight market. *Maritime Policy & Management*, 44(4):413–425.
- Adland, R. and Strandenes, S. (2006). Market efficiency in the bulk freight market revisited. *Maritime Policy & Management*, 33(2):107–117.
- Batchelor, R., Alizadeh, A., and Visvikis, I. (2007). Forecasting spot and forward prices in the international freight market. *International Journal of Forecasting*, 23(1):101–114.
- Benth, F. E. and Koekebakker, S. (2016). Stochastic modeling of supramax spot and forward freight rates. *Maritime Economics & Logistics*, 18(4):391–413.
- Berg-Andreassen, J. (1997). Efficiency and interconnectivity in international shipping markets. *International Journal of Transport Economics/Rivista internazionale di economia dei trasporti*, pages 241–257.
- Brownlee, J. (2018). *Better Deep Learning: Traion Faster, Reduce Overfitting, and Make Better Predictions*. Machine Learning Mastery.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Chollet, F. et al. (2015). Keras. <https://keras.io>.
- Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.
- Dunne, R. A. and Campbell, N. A. (1997). On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. In *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne*, volume 181, page 185. Citeseer.
- Engle, R. F. and Granger, C. W. (1987). Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society*, pages 251–276.
- Fan, S., Ji, T., Gordon, W., and Rickard, B. (2013). Forecasting baltic dirty tanker index by applying wavelet neural networks.
- Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.
- Gal, Y. and Ghahramani, Z. (2015). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *arXiv preprint arXiv:1506.02142*.

- Gal, Y. and Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.
- Gençay, R., Dacorogna, M., Muller, U. A., Pictet, O., and Olsen, R. (2001). *An introduction to high-frequency finance*. Elsevier.
- Glen, D. and Rogers, P. (1997). Does weight matter? a statistical analysis of the ssy capesize index. *Maritime Policy and Management*, 24(4):351–364.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.
- Hemmati, A., Hvattum, L. M., Fagerholt, K., and Norstad, I. (2014). Benchmark suite for industrial and tramp ship routing and scheduling problems. *INFOR: Information Systems and Operational Research*, 52(1):28–38.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- 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.
- Jozefowicz, R., Zaremba, W., and Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In *International conference on machine learning*, pages 2342–2350.
- Kanamoto, K., WADA, Y., and SHIBASAKI, R. (2019). Predicting a dry bulk freight index by deep learning with global vessel movement data. In *Transdisciplinary Engineering for Complex Socio-technical Systems: Proceedings of the 26th ISTE International Conference on Transdisciplinary Engineering, July 30–August 1, 2019*, volume 10, page 105. IOS Press.
- Kavussanos, M. G., Visvikis, I. D., et al. (2010). The hedging performance of the capesize forward freight market. *Eds.) Cullinane, K., The International Handbook of Maritime Economics and Business, Edward Elgar Publishing*.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Koekebakker, S., Adland, R., and Sødal, S. (2006). Are spot freight rates stationary? *Journal of Transport Economics and Policy (JTPEP)*, 40(3):449–472.
- Laulajainen, R. (2007). Dry bulk shipping market inefficiency, the wide perspective. *Journal of Transport Geography*, 15(3):217–224.
- Lyridis, D., Zacharioudakis, P., Mitrou, P., and Mylonas, A. (2004). Forecasting tanker market using artificial neural networks. *Maritime Economics & Logistics*, 6(2):93–108.
- Nielsen, D. (2016). Tree boosting with xgboost-why does xgboost win " every " machine learning competition? Master's thesis, NTNU.
- Prochazka, V., Adland, R., and Wallace, S. W. (2019). The value of foresight in the drybulk freight market. *Transportation Research Part A: Policy and Practice*, 129:232–245.

- Research, C. (2015). Sources methods for the shipping intelligence weekly. [http://www.clarksons.net/archive/research/archive/SNM/SIW\\_SNM.pdf](http://www.clarksons.net/archive/research/archive/SNM/SIW_SNM.pdf).
- Semeniuta, S., Severyn, A., and Barth, E. (2016). Recurrent dropout without memory loss. *arXiv preprint arXiv:1603.05118*.
- Stopford, M. (2009). *Maritime economics*. Routledge.
- Sutherland, A. et al. (2013). *The Front Office Manual: The Definitive Guide to Trading, Structuring and Sales*. Springer.
- Tsioumas, V. and Papadimitriou, S. (2015). Excess returns in the spot market for bulk carriers. *Maritime Economics & Logistics*, 17(4):399–415.
- Tsioumas, V. and Papadimitriou, S. (2018). The dynamic relationship between freight markets and commodity prices revealed. *Maritime Economics & Logistics*, 20(2):267–279.
- Tvedt, J. (2003). Shipping market models and the specification of freight rate processes. *Maritime Economics & Logistics*, 5(4):327–346.
- Xiangfu, T. (2018). Imo 2020 fuel regulation impact on freight cost: A stance from capesize market. <https://www.linkedin.com/pulse/imo-2020-fuel-regulation-impact-freight-cost-stance-from-xiangfu-tan/>.

# Appendix

## A1 Index Volatility



**Figure A1.1:** Historical values of the Baltic Exchange Capesize drybulk indices. Data obtained from the Clarkson Shipping Intelligence Network.

## A2 Tripdetails

**Table A2.1:** Abstract of *Bulkcarrier Voyage Details (2009 Onwards)*, obtained from Clarkson Research

No.	Voyage		Cargo Size Tonnes	Voyage Dist. Miles		Voyage Time - Days					Bunker Port
	Load	Discharge		Ladden	Ballast	Sea Time	Sea Margin	Port Time	Total Voyage Time	Oper. Speed Knots (L/B)	
B146	Tubarao	Qingdao	176,000	11,086	4,974	54.4	2.7	9.5	69.0	12.0/13.0	Singapore*
B148	Dampier	Qingdao	174,000	3,582	3,500	23.7	1.2	7.5	35.1	12.0/13.0	Qingdao

\* Vessel will take bunker twice, one on the laden leg, the other on the ballast leg (Xiangfu, 2018)

## A2.1 Historical Tripcharter Rates

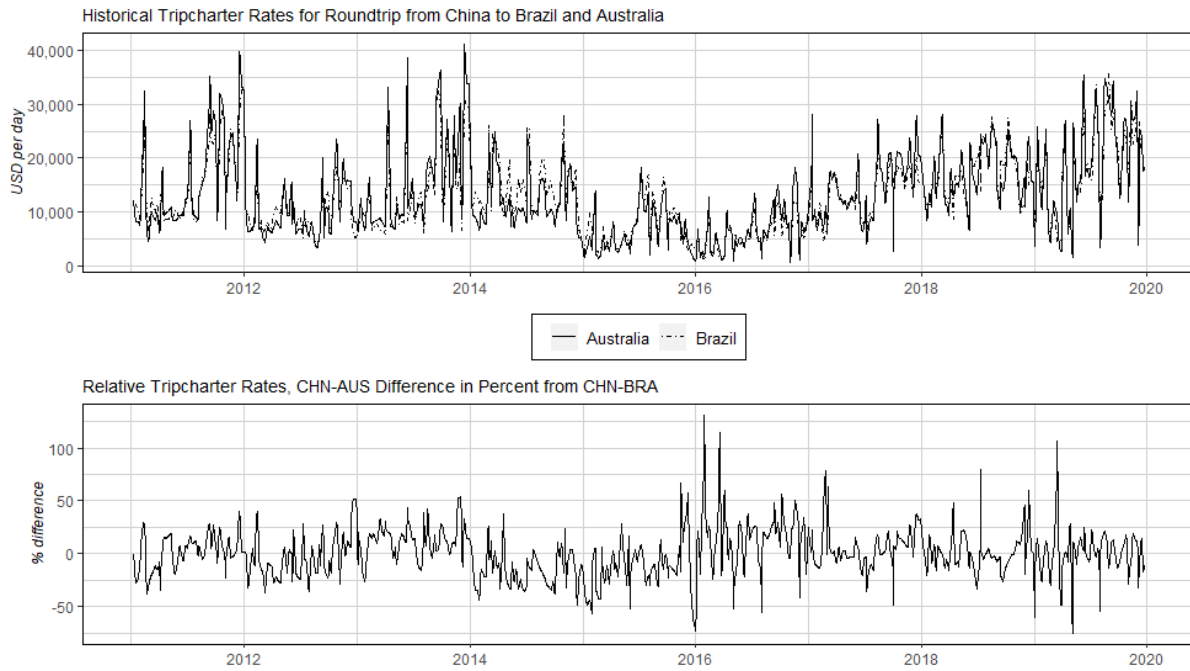


Figure A2.1: Historical Tripcharter Rates.

## A3 Confusion Matrix

Table A3.1: Confusion Matrix with XGBoost predictions.

		Reference	
		0	1
Prediction	0	195	156
	1	162	483

Table A3.2: Confusion Matrix with LSTM predictions.

		Reference	
		0	1
Prediction	0	120	47
	1	237	592

## A4 T-Test

**Table A4.1:** Value of switch sorted by predictions.

Model	Predictions	Number of observations	Mean Value of Switch	Mean Absolute Value of Switch
XGBoost	Correct	678	150125.59	268920.17
XGBoost	Incorrect	318	23680.80	215583.43
LSTM	Correct	712	207296.07	277143.19
LSTM	Incorrect	284	-134785.77	188582.58

**Table A4.2:** Output from one-sided t-test with unequal variance.†

Model	Difference in absolute mean	P-value	T-value	Degrees of Freedom	Lowerbound*	Upperbound*
XGBoost	53336.74	2.449e-05	4.08	794.97	31823.95	Inf
LSTM	88560.61	6.162e-12	6.89	711.57	67389.83	Inf

† Tested for if the absolute mean of incorrectly predicted decisions is higher than correctly predicted decision  
 \*The lower and upperbound are the values from a 95% confidence intervall

## A5 Individual Vessel Results

**Table A5.1:** Overview of cumulative earnings achieved by each vessel in the fleet.

Vessel	Vessel		Cumulative Earnings				Relative Cumulative Earning (%)		
	Start Date	End Date	Perfect Foresight	Benchmark	LSTM	XGBoost	Perfect Foresight	LSTM	XGBoost
Initial Vessel	2016-12-23	2019-09-14	18518344	15870224	17677614	17133639	16.69	11.39	7.96
Vessel 1	2017-02-11	2019-09-14	15080157	12728519	13906517	13324048	18.48	9.25	4.68
Vessel 2	2017-04-02	2019-09-14	17842962	15354550	17126442	16441794	16.21	11.54	7.08
Vessel 3	2017-05-22	2019-09-14	13712199	11645012	12552851	12026954	17.75	7.80	3.28
Vessel 4	2017-07-11	2019-09-14	15891488	13774141	15373787	15137206	15.37	11.61	9.90
Vessel 5	2017-08-30	2019-09-14	12574994	10715828	11453566	10953465	17.35	6.88	2.22
Vessel 6	2017-10-19	2019-09-14	14197684	12363860	13737844	13432181	14.83	11.11	8.64
Vessel 7	2018-01-27	2019-09-14	12088512	10391873	11674708	11310675	16.33	12.34	8.84
Vessel 8	2018-03-18	2019-09-14	9330850	7751991	8393352	7908559	20.37	8.27	2.02
Vessel 9	2018-05-07	2019-09-14	10175952	8812703	9935876	9381107	15.47	12.74	6.45

**Table A5.2:** Overview of average daily earnings in USD achieved by each vessel in the fleet.

Vessel	Vessel		Average Daily Earnings (USD)				Difference from benchmark (USD)		
	Start Date	End Date	Perfect Foresight	Benchmark	XGBoost	LSTM	LSTM	XGBoost	Perfect Foresight
Initial Vessel	2016-12-23	2019-09-14	18601	15942	17211	17757	1815	1269	2659
Vessel 1	2017-02-11	2019-09-14	15950	13464	14093	14709	1245	630	2486
Vessel 2	2017-04-02	2019-09-14	19931	17153	18366	19131	1978	1213	2778
Vessel 3	2017-05-22	2019-09-14	16223	13779	14231	14853	1073	452	2444
Vessel 4	2017-07-11	2019-09-14	19973	17313	19025	19322	2009	1712	2659
Vessel 5	2017-08-30	2019-09-14	16876	14384	14702	15373	989	319	2492
Vessel 6	2017-10-19	2019-09-14	20429	17793	19329	19769	1976	1536	2636
Vessel 7	2018-01-27	2019-09-14	20304	17457	18999	19610	2153	1542	2848
Vessel 8	2018-03-18	2019-09-14	17110	14220	14505	15393	1174	286	2891
Vessel 9	2018-05-07	2019-09-14	20556	17806	18954	20072	2266	1148	2750

## A6 Variable Importance

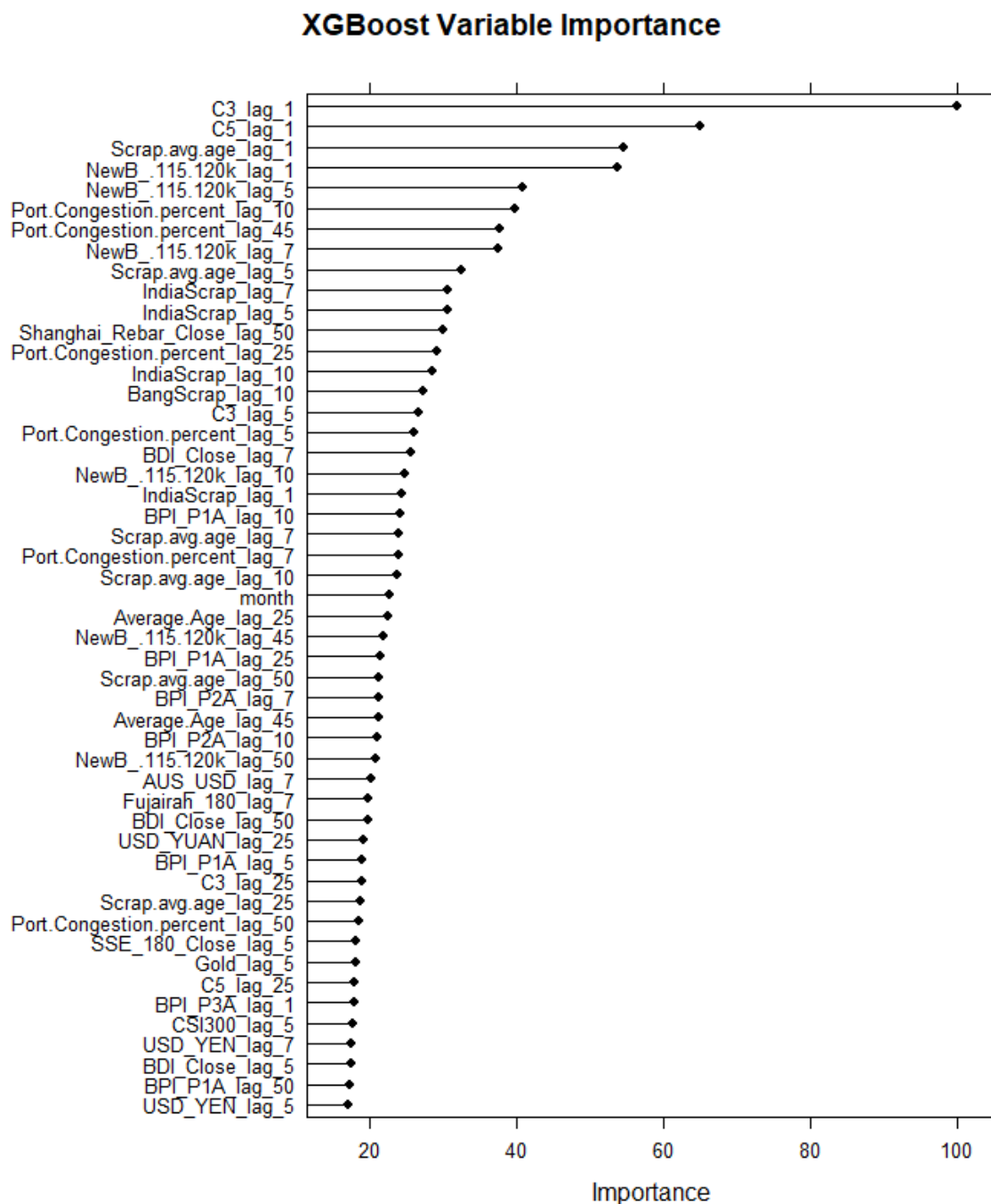


Figure A6.1: XGBoost Variabel Importance top 50 variables.