Forecasting of Danish stocks prices using artificial neural networks

Andreas Borup Jørgensen - 20164559
Mette Koch Møller - 20164146
Robert Høstrup - 20166322
10/4. semester - Speciale

AALBORG UNIVERSITET

4. Juni 2021

Forecasting of Danish stocks prices using artificial neural networks

A study comparing forecasts from artificial neural networks with forecasts from autoregressive moving average models.

Autors:
Andreas Borup Jørgensen,
Mette Koch Møller
Robert Høstrup

Supervisor:
Hamid Reza
raza@business.aau.dk

Thesis Period:
Spring semester 2021

Page numbers excluding/including appendix and bibliography:
77 / 94

Andreas Borup Jørgensen ajarge16@student.aau.dk	Date
Mette Koch Møller mmalle16@student.aau.dk	Date
Robert Høstrup rhastr16@student.aau.dk	Date

Aalborg University Business School
Aalborg University
Denmark

June 4, 2021

Abstract

Economists describe the stock market as an efficient market so that there is no way to forecast its future prices. Rapid development in computational power has led to machine learning techniques with stronger predictive power becoming more prominent.

The scope of this thesis is to examine artificial neural networks' ability to forecast the Danish stock market. A total of six Danish stocks are forecasted using daily one-step-ahead forecasting. The time period used for this purpose is the year between 2010 and 2019, where the first eight years is used for training and testing, while the last is used for forecasting.

Different architectures are tested, including FNN, RNN, LSTM, GRU, bi-LSTM, and bi-GRU. Each architecture is tested both with and without the inclusion of extra explanatory variables. In order to evaluate the models' forecasts, ARIMA and ARIMAX are used as baseline models.

The simpler FNN structure without the addition of explanatory variables proved, via out-of-sample forecasting, to be the more accurate of the artificial neural network models. The forecasts from the FNN models were, in four of the six stocks forecasted, statistically less accurate when compared to the forecasts from the ARIMA and ARIMAX models. It was found that ANNs need further development before their usage in forecasting stock market prices becomes feasible.

Contents

List of Figures **v**

List of Tables **vi**

1 Introduction **1**

1.0.1 Artificial Intelligence . . . 2

1.1 Problem statement . . . 3

1.2 Problem delimitation . . . 3

1.3 Course of action . . . 4

1.4 Contribution . . . 5

2 Literature review **7**

2.1 AutoRegressive models . . . 9

2.1.1 ARIMA models . . . 9

2.1.2 ARIMA with extra explanatory variables . . . 9

2.2 Machine Learning . . . 10

2.2.1 Supervised Machine Learning . . . 11

2.2.2 Artificial Neural Network . . . 12

2.3 Comparative studies . . . 14

3 Stock market - Theory **16**

3.1 The stock markets and the economy . . . 16

3.2 The efficient market hypothesis . . . 17

3.2.1 Random walks . . . 17

3.2.2 The efficient market hypothesis' critics . . . 18

4 ARIMA - Theory **19**

4.1 Stationarity . . . 19

4.2 AutoRegressive Integrated Moving Average . . . 19

4.3 AutoRegressive Integrated Moving Average with external regressors . . . 20

4.4 Akaike Information Criterion . . . 21

5 ANN - Theory **23**

5.1 Artificial Neural Networks . . . 23

5.1.1 Training an ANN . . . 25

5.2 Recurrent Neural Networks . . . 27

5.2.1 Long-short-term memory . . . 28

5.2.2 Gated Recurrent Unit . . . 29

5.2.3 Bidirectionality . . . 30

5.3 Hyper parameters . . . 31

6 Evaluation methods **32**
6.1 Root Mean Square Error 32
6.2 Mean Absolute Percentage Error 32
6.3 Diebold-Mariano test 33

7 Methodology **35**
7.1 Out-of-sample testing 35
7.2 Explanatory variable selection 35
7.3 Missing values 36
7.4 ARIMA(X) selection method 37
7.5 ANN selection methods 37

8 Data selection **39**
8.1 Time period 39
8.2 Dependent variables 40
8.3 Explanatory variables 41
8.4 Summary 43

9 Exploratory data analysis **44**
9.1 Summary 52

10 Selection of ARIMA(X) models **53**
10.1 Summary 58

11 Selection of ANN models **59**
11.1 Summary 64

12 Results **65**
12.1 Summary 69

13 Discussion **71**
13.1 Does the Danish stock market behave as a random walk? 71
13.2 Effect of adding explanatory variables 72
13.3 Comparison to existing literature 73
13.4 Possible improvements to methodology 73
13.5 Real life application of methods 74
13.6 Summary 75

14 Conclusion **76**

15 Appendix **83**
15.1 Appendix 1 - Granger Causality test 83
15.2 Appendix 2 - Dependent variables 84

15.3 Appendix 3 - Hyperparameter tuning 85
15.4 Appendix 4 - Structure for ANN testing models 86
15.4.1 Structure of ANN for Carlsberg 86
15.4.2 Structure of ANN for Genmab 87
15.4.3 Structure of ANN for Jyske Bank 88
15.4.4 Structure of ANN for Mærsk B 90
15.4.5 Structure of ANN for SimCorp 91
15.4.6 Structure of ANN for Vestas 92
15.5 Appendix 5 - Abbreviation used in the thesis 94

List of Figures

1	Simple FNN	23
2	Recurrent layer	27
3	LSTM gate	28
4	GRU gate	30
5	Illustration of time period	39
6	Carlsberg stock price	46
7	Genmab stock price	47
8	Jyske Bank stock price	48
9	Mærsk B stock price	49
10	SimCorp stock price	50
11	Vestas stock price	51
12	Final structure for Carlsberg	60
13	Final structure for Genmab	60
14	Final structure for Jyske Bank	61
15	Final structure for Mærsk B	61
16	Final structure for SimCorp	62
17	Final structure for Vestas	62
18	Forecasted stock prices for ARIMA(X) and ANN	69
19	Carlsberg without explanatory variables	86
20	Carlsberg with explanatory variables	86
21	Genmab without explanatory variables	87
22	Genmab with explanatory variables	88
23	Jyske Bank without explanatory variables	88
24	Jyske Bank with explanatory variables	89
25	Mærsk B without explanatory variables	90
26	Mærsk B with explanatory variables	90
27	SimCorp without explanatory variables	91
28	SimCorp with explanatory variables	92
29	Vestas without explanatory variables	92
30	Vestas with explanatory variables	93

List of Tables

1	Overview of literature reviewed	7
2	Augmented Dickey–Fuller p-value	44
3	Descriptive statistic for Carlsberg	46
4	Descriptive statistic for Genmab	47
5	Descriptive statistic for Jyske Bank	48
6	Descriptive statistic for Mærsk B	49
7	Descriptive statistic for SimCorp	50
8	Descriptive statistic for Vestas	51
9	AIC for ARIMA(X) models	56
10	RMSE and MAPE for ARIMA and ARIMAX	57
11	RMSE and MAPE for all ANN models	63
12	Results of forecasting	65

1 Introduction

The task of forecasting stock prices has long been of interest to both economists and investors. For investors, the interest stems from the potential competitive advantage one could achieve with accurate forecasts. Economists disagree on whether changes in the economy causes stock market changes, stock market changes causes economic changes or the two simply follow the same patterns independently of each other. What economists do agree on, however, is the strong correlation between the stock market and the economy as a whole. This was seen both in the vast decline in aggregate demand in 1974-1975, a few years after the stock market collapse in 1973-1974 (Bosworth et al., 1975) and in the rise in the unemployment rate in the wake of the financial crises in 2007-2008 (Hurd and Rohwedder, 2010). It is safe to say that anyone who seeks to understand the whole economy and how it works must consider the motions of the stock market. One way economists have historically described the stock market is through the efficient market hypothesis.

The stock market is often characterised as a highly efficient market, as described in the efficient market hypothesis. The hypothesis states that when new information arises, it is incorporated into the stock prices without delay. This means that all information that could be used to predict a stock price is already incorporated in the stock's current price. Therefore, it is assumed that there is no point in trying to forecast stock prices (Malkiel, 2003). The validity of the hypothesis has often been questioned, both by Malkiel himself, but also by others (W. Lo and MacKinlay, 1999), who observed short-run pricing irregularities. Grossman and Stiglitz (1980) also observed these irregularities and further pointed out that the existence of a professional market for stock trading necessitates informational imperfections. The question is whether these short-run effects can be classified and forecasted. In 2003 Malkiel pointed out that this would be possible one day:

"*Undoubtedly, with the passage of time and with the increasing sophistication of our databases and empirical techniques, we will document further apparent departures from* *efficiency and further patterns in the development of stock returns*" - Malkiel (2003)

Much has happened since 2003, and the available techniques to forecast the stock market has indeed expanded. One technique that has seen a vast transformation since then is artificial intelligence. This technique has gained traction in recent years and is on its way to change many industries (Agrawal et al., 2020). As such, it seems like an obvious candidate to predict the stock market.

1.0.1 Artificial Intelligence

Artificial Intelligence (AI[1]) is a wide field of study, and it can be quite hard to give a precise definition. John McCarthy, known as one of the leading figures of AI, described it as:
"*It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.*" - McCarthy (2007)
So, in essence, AI makes it possible to use computer technology to perform human tasks or even tasks beyond human abilities. One subset of AI that is often used to forecast is Machine Learning (ML). The basic principles of ML are to make a computer learn by itself without explicitly giving it instructions on what to learn. Some ML models are even programmed to mimic the human brain. Although ML sounds like a modern and new method, its principles have been known for many years. The idea of modelling the brain was already introduced in 1949 by Donald Hebb (Foote, 2019). ML had a slow start in its early years, partly caused by "AI winters", where a reduction of funding and interest led to a significant decrease in development. A big reason for these winters was the lack of computational power and data available at the time. However, this has changed in recent years (Schuchmann, 2019).

A rapid increase in computers, becoming stronger and cheaper, has led to the task of developing ML not only lying with trained computer engineers but instead with everyone interested. This means that in recent years, ML has become very popular due to it

[1] All abbreviations used in this thesis and their full terms can be found in *Appendix 5 - Abbreviation used in the thesis*

becoming feasible for not only professional developers but also everyday people (Brownlee, 2020a).

The continuous increase of data and computer power combined with the feasibility for the general public has made ML a much more powerful and accessible tool. The ML methodology covers a wide selection of different techniques. One technique that is often used to forecast stock prices is the *Artificial Neural Network* (ANN). ANNs are one of the ML techniques that tries to imitate the human brain. The advantages of ANNs over other ML models are that ANNs are often better at working with more complex and sequential data.

1.1 Problem statement

Malkiel's statement from 2003 about how future empirical techniques could be able to pattern the stock market combined with the huge development in both ML models and computer power begs the obvious question: Has ML models reached a point where they can feasibly predict future stock prices? The wide range of different ML techniques of cause makes this question far too comprehensive to answer within the scope of this thesis.

Instead, only the before mentioned branch of ML called *Artificial Neural Network* (ANN) will be examined. To generally assess the predictive power of ANNs across all the different stock markets seem unrealistic as well. For this reason, ANN is only examined using the Danish stock market.

Based on the above, this thesis will be produced using the following problem statement:

How well do artificial neural network models predict future prices of the Danish stock market?

1.2 Problem delimitation

The problem statement is already narrow and precise but still needs some delimitation. The ML method, ANN, covers a broad spectrum of model types and testing them all would be far too big a task to fit in any one thesis. The model types tested in this thesis are: A Feed-forward Neural Network (FNN), a Recurrent Neural Network (RNN), a Long-

Short-Term-Memory (LSTM), a bidirectional-LSTM (bi-LSTM), a Gated-Recurrent-Unit (GRU), and a bidirectional-GRU (bi-GRU).

Training ANN models is both a time consuming and computationally heavy task. This means that forecasting every Danish stock would either be too time-consuming or mean that the models would not be sufficiently tuned. The tuning method chosen in this thesis dictates that 1800 different models are trained per stock forecasted. In this thesis, a total of six stocks will be used to examine the problem statement.

1.3 Course of action

The answering of the problem statement is based on an empirical test. The ANN models investigated will be set to forecast the daily stock prices of six Danish stocks in the year 2019, using one-step-ahead forecasting. The daily stock prices from 2010-2018 will be used to train and evaluate the models before actually forecasting. To evaluate these forecasts, the AutoRegressive Integrated Moving Average (ARIMA) will be set to forecast the same stocks in the same period, using the same forecasting method. As such, the ARIMA will be used as a baseline model, and the results will be compared with those of the ANN models. It is common practice in empirical studies to use ARIMA models as a comparison since they are simple to implement and produce forecasts that are generally accurate (Yuan et al., 2016; Lütkepohl and Xu, 2010; Navares et al., 2018; Thomakosa and GuerardJr, 2004; Li and Li, 2017).

The data set is split up into two parts to imitate a real-life forecasting situation. The first part of the data (2010-2018) is used to train and test the ARIMA and ANN models, where only one model will be chosen for each type per stock. The last part of the data (2019) will be used to make the actual forecasts by the models chosen in the first period and evaluate the final results.

The part of the data where the models are trained and tested is further split up into two parts: Training and testing. The training period is first used to train[2] different ARIMA and ANN models. After the models are trained on the training period, they are used

[2]Building, fitting, regressing, and similar terms will all be referred to as "train" for convenience.

to forecast the test period. This method is used to check how well models trained on a specific period can make predictions on another. The results are further used to choose which ARIMA and which ANN model will be used to make the final forecasts.

The performance of ANN models is often improved when feeding the models more relevant information than just the time series of the dependent variable. For this reason, a set of explanatory variables will be included in the models of the thesis. Every ANN model type will be trained both with and without the inclusion of these explanatory variables. The ANN models use five time-steps to forecast the dependent variables. The ARIMA models will also be trained with and without these explanatory variables to ensure the comparison between ARIMA and ANN models is valid. The extra explanatory variables will be used with one lag in the AutoRegressive Integrated Moving Average with explanatory variables (ARIMAX) models. The explanatory variables include all the stocks from the C25 index, stock indices, government bonds, currency conversion rates, and gold and oil futures.

Only one ARIMA(X)[3] and one ANN model will be chosen to forecast each dependent variable. To better simulate a real-world application, the forecast period is only used with the final models, and no changes in the model structure will occur based on the results.

1.4 Contribution

The idea of using machine learning to predict stock prices is not entirely new. Therefore, this thesis does not aim to make groundbreaking discoveries but instead to add to a growing existing literature. This addition is based on the combination of both machine learning techniques and adding additional information to the models. The stock market this thesis will try to forecast is the Danish stock market. Although forecasting has been done before, machine learning-based forecasting of the Danish stock market has not been attempted before to the best of the writers' knowledge. Forecasting the Danish stock market might be a solid addition to the literature since Denmark is a small export economy and its stock market is well developed. A well-developed stock market should mean that the Danish stock market is most likely an efficient market rather than a young

[3]This notation is used when referring to both ARIMA and ARIMAX models

stock market that has not reached a point of efficiency.

2 Literature review

In this chapter, existing literature that contains either ARIMA models, ML models, or both are reviewed. Since the purpose of this thesis is to forecast stocks, only literature regarding stock forecasting is considered. The purpose of reviewing existing literature is to get an overview and understanding of similar research. This chapter is split into three sections. In section 2.1 only existing literature concerning ARIMA models will be presented. Section 2.2 presents literature that focuses on machine learning alone. Section 2.3 presents articles that compare ARIMA and ANN models.

Table 1 presents the most important information for the ARIMA, machine learning, and comparative literature. The table contains the name of the author(s), the publishing year, a brief summary of the study, and their conclusions.

Table 1: Overview of literature reviewed

ARIMA literature

Author	Year	Summary	Conclusion
Mondal et al.	2014	Examination of different fitting period lengths when building an ARIMA.	No signigicant difference between an ARIMA created on 23-, 18-, 12- or 6-months' ability to forecast.
Afeef et al.	2018	ARIMA(1,1,1) forecast of Pakistani oil company's stock price.	Strong ability to forecast in the short run.
Virtanen and Yli-Olli	1987	Examination of uni-, multivariate and combined models' abilities to forecast.	The Combined model proved to predict stronger forecasts with lower RSME.

(a) Overview of econometric litterature

Machine Learning literature

Author	Year	Summary	Conclusion
Dingli and Fournier	2017	Comparison of different SVM models for Tech and Finance industry.	Mixed result. Often Linear regression and SVM were the better model.
Shen et al.	2012	Comparison of different SML models and creation of a trading model.	SMV gave the better forecast and outperformed the benchmark trading model.

Abe and Nakayama	2018	Comparison of RF, SVR and, deep learning models.	A deep learning model with more layers were the better model.
Naik and Mohan	2019	Comparison of RF, SVR, ANN, and a deep learning.	The deep learning model outperformed the other models.
Beyaz et al.	2018	Comparison of SVR and ANN models.	SVR had the lowest RMSE.
Mehtab et al.	2020	Comparison of SVM, ANN, and LSTM models.	A LSTM model were the superior model.
Balaji et al.	2018	Comparison of CNN, LSTM, GRU, and ELM models.	GRU were better for shorter forecasts and ELM were better for longer forecasts.

(b) Overview of Machine Learning literature

Comparative studies

Author	**Year**	**Summary**	**Conclusion**
Du	2018	Comparison of ARIMA, ANN and Hybrid models.	ANNs outperform ARIMA and Hybrid models outperform both.
Siami-Namini et al.	2018	Comparison of LSTM and ARIMA in financial and general economic data.	LSTM achieves 84-87 per cent lower RMSE.
Isenahd and Olubusoye	2014	Comparison of ARIMA and ANN on Nigerian stock market.	The ANN outperform the ARIMA.

(c) Overview of comparative literature

As can be seen in table 1, the existing literature shows that both ARIMA and machine learning models are able to make somewhat accurate stock market predictions. Since the literature uses either different evaluation metrics or evaluation metrics that can not be compared on different stocks (e.g. RMSE), the different results across the literature can not be directly compared. The machine learning literature, in particular, shows that artificial neural networks are often better at forecasting stock prices than other supervised machine learning models. The comparative studies all come to the same conclusion, namely that artificial neural networks outperform ARIMA models. The following sections will explain these articles in depth.

2.1 AutoRegressive models

This section covers literature where ARIMA and ARIMAX models are used to forecast stock markets. Section 2.1.1 includes two studies that forecast two different stock markets using ARIMA models. Section 2.1.2 covers a study that compares the forecast strength between a normal ARIMA, a linear model with external explanatory variables, and an ARIMAX that is a combination of the two.

2.1.1 ARIMA models

ARIMA models are often used to forecast financial assets. Mondal et al. (2014) and Afeef et al. (2018) both studies different ARIMA models and their ability to forecast stock markets.

In their article *Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices* Mondal et al. (2014) examines the Indian stock market index NSE using an ARIMA(1,0,2). Mondal et al. uses the time period 2012-2014 for their study. They test four different training period lengths: 23-, 18-, 12-, and 6-months. No significant difference was found in the models' abilities to forecast stock prices across different sectors. The models in the study all had an accuracy[4] of at least 85 per cent (Mondal et al., 2014).

In Afeef et al.'s (2018) study *Pakistan Oil & Gas Development Company Limited*, they focus on one company. They create different ARIMA models to forecast the Pakistani oil company *Pakistan Oil & Gas Development Company Limited*. Based on a time period from 2004 to 2018, they find that an ARIMA(1,1,1) have the strongest fit. Afeef et al. conclude that the model's ability to forecast is strong in the short run (Afeef et al., 2018).

2.1.2 ARIMA with extra explanatory variables

Virtanen and Yli-Olli (1987) wrote an article named *Forecasting stock market prices in a thin security market*, where they examined different econometric ways to forecast the Finish stock market. The study examined the time frame 1975-1986, where the years 1975-1984 were used as data to build the models, while the years 1985-1986 were used to forecast. Virtanen and Yli-Olli trained three different types of models; a univariate

[4] Accuracy is calculated using MAE

ARIMA model, a multivariate time series model and a combination of the two. The univariate method led to the construction of two different models: A classic ARIMA$(0, 2, 1)$ and an ARIMA with a seasonal component ARIMA$(0, 2, 1)(0, 1, 1)^4$. Both models were significant, with p-values smaller than one per cent. The multivariate model was created with six explanatory variables: Lagged versions of the stock, anticipated future cash flow, return on Finish state bonds, money supply (Finish Mark), inflation, and the Swedish stock market (Stockholm Stock Exchange). The coefficients of all explanatory variables proved to be significant at a five per cent level. Two combined models were created: One where the standard ARIMA ($f(A)$) was combined with the multivariate model ($f(M)$) and one where the ARIMA with a seasonal component ($f(AS)$) were combined with the multivariate model ($f(M)$). The combined models[5] had the formulas:

$$y_t = \alpha + \beta_1 f(M) + \beta_2 f(A) + \epsilon \tag{1}$$

$$y_t = \alpha + \beta_3 f(M) + \beta_4 f(AS) + \epsilon \tag{2}$$

β_2 proved to be insignificant in the model with no seasonal component, as to why the model was not used for forecasting. Virtanen and Yli-Olli used many different methods to examine the strength of the models' forecasts. The different methods all pointed to the same conclusion. The combined model (RMSE = 0.069) outperformed both the univariate ARIMA (RMSE = 0.095) and the multivariate model (RMSE = 0.082) (Virtanen and Yli-Olli, 1987).

Summary

The ARIMA method proved to be able to forecast stock markets without considerable errors in the three studies examined. The last study by Virtanen and Yli-Olli showed that a combination (ARIMAX) of ARIMA and multivariate models produces the strongest forecasts.

2.2 Machine Learning

A variety of machine learning studies have tried different approaches and models in order to forecast the stock market. In this section, existing literature regarding machine learning

[5] These models are what this thesis refers to as ARIMAX models

and stock forecasting is presented. The simpler models are presented first, and more advanced models are introduced later on.

2.2.1 Supervised Machine Learning

Several different machine learning approaches have been used across studies when trying to forecast the stock market. Traditional Supervised Machine Learning (SML) models are often used, both Dingli and Fournier (2017) and Shen et al. (2012) use SML models in their studies.

Dingli and Fournier (2017) wrote the article *Financial Time Series Forecasting – A Machine Learning Approach* where they created several SML models to predict the stock price for the tech and finance industry. They created three different models: A classification that predicts whether the price goes up or down, a regression that predicts the change in the price, and a regression that predicts the actual price. Each of the models is estimated daily, weekly, monthly, quarterly, and yearly. They trained their model with explanatory variables[6] which includes 200 different technical indicators divided into three categories: Currency exchange rates, world indices, and commodity prices. All the explanatory variables are not included in each model; only the explanatory variables with statistical importance were selected. The explanatory variables were selected based on an ANOVA f-test. Therefore each model is created with a different number of explanatory variables (Dingli and Fournier, 2017).
Dingli and Fournier test the forecasting ability of different models, e.g. logistic regression, Support Vector Model (SVM), Decision Trees (DT), and a Random Forest (RF). The classification models are evaluated using accuracy, and the regression models are evaluated using Root Mean Square Error (RMSE). Different models performed best, depending on the time period and stock. However, linear regression and SVM were often the better models. For the regressions models, they found that only including a small number of explanatory variables were better when using daily data compared to the other frequencies. The best model for the finance industry only had one explanatory variable and three for the tech industry. On the other hand, weekly data had 38 and 69 explanatory variables.

[6] In the machine learning terminology, explanatory variables are described as features.

Shen et al. (2012) also tested several SML models in their article *Stock Market Forecasting Using Machine Learning Algorithms.* They created classification and regression models as well and found that SVM was a better model at forecasting the NASDAQ, DJIA, and S&P 500. When predicting whether the stock price rose or fell, the models had an accuracy of 74.4, 77.6, and 76 per cent, respectively. Like Dingli and Fournier (2017) they created models with explanatory variables. Which explanatory variables to include were based on cross-correlation. Different models were tested, and they found that only including four explanatory variables lead to a higher accuracy than models that included more.

Additionally, they found that SVM is sensitive to the training set's size, where a longer training period leads to higher accuracy and lower RMSE. The article differs by creating three different trading models: One based on the results from the SVM model and two benchmark models. In most cases, the trading model containing the SVM results outperformed the benchmark models (Shen et al., 2012).

2.2.2 Artificial Neural Network

In the article *Deep Learning for Forecasting Stock Returns in the Cross-Section*, the writers Abe and Nakayama (2018) create several machine learning models in order to forecast the one-month-ahead stock price for the MSCI Japan index. The explanatory variables are 25 different factors, e.g. book-to-market ratio, market beta and volatility (Abe and Nakayama, 2018).

Abe and Nakayama create different models consisting of an RF, an SVR[7], and 16 different deep learning models. The different deep learning models are differentiated by their number of layers, units and percentage of dropout. The models are evaluated by their correlation coefficient, directional accuracy, and mean square error (MSE). Generally, the deep learning models had better results, where a greater amount of layers led to a better model (Abe and Nakayama, 2018).

Naik and Mohan (2019) found the same results in their article *Stock Price Movements Classification Using Machine and Deep Learning Techniques-The Case Study of Indian*

[7]Same principle as an SVM model but for a regression problem.

Stock Market. Comparing an RF, SVM, Artificial Neural Network (ANN), and a deep learning model, they found that deep learning outperformed the remaining models for a classification (Naik and Mohan, 2019).

Beyaz et al. (2018) also compared an ANN and an SVR model in their article *Comparing Technical and Fundamental indicators in stock price forecasting*. They tried to combine fundamental and technical analysis in a machine learning model. Opposite Naik and Mohan (2019), Beyaz et al. found that the SVM had a lower RSME compared to the ANN model. Beyaz et al. Also found that this was the case regardless of the combination of explanatory variables (Beyaz et al., 2018).

Mehtab et al. (2020) create eight different SML and ANN models in their article *Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models*. They differ by creating four different Long-Short-Term-Memory (LSTM) models. They found that the LSTM models were superior to the SML and standard ANN models (Mehtab et al., 2020).

In their article *Applicability of Deep Learning Models for Stock Price Forecasting An Empirical Study on BANKEX Data*, Balaji et al. (2018) focus on deep learning models. They create fourteen different models based on the techniques: Convolutional Neural Networks (CNN), Gated Recurrent Unit (GRU), LSTM, and Extreme Learning Machines (ELM). The models differ by the number of layers and how many steps ahead the models forecast. They tested the models on different stock prices from the bank sector. All of the deep learning models had good results. The GRU model tends to have better results for a shorter forecast, while the ELM model tends to be the better model for longer forecasts (Balaji et al., 2018).

Summary

Using SML models, adding only the most significant explanatory variables often led to better results. However, compared to SML models, the ANN models often turned out as the better model. When comparing a standard ANN to the more complicated Recurrent Neural Network (RNN) models, the RNN architectures, such as LSTM and GRU, outperformed the simpler models.

2.3 Comparative studies

In the past two sections, literature which either uses ARIMA *or* machine learning models have been reviewed. In the following section, the literature that compares ARIMA and ANN models performance when forecasting stock market prices is reviewed.

In their article, *Application and analysis of forecasting stock price index based on combination of ARIMA model and BP neural network*, Du (2018) uses forecasting of the Shanghai Securities composition stock index to compare ARIMA, ANNs, and a hybrid of the two. The data consist of daily data points from January to December 2017. The models created were an ARIMA(2,1,1), a feed-forward neural network, and a hybrid ARIMA-Neural network model that uses the residual values from the ARIMA model as an explanatory variable for the neural network model.
The testing data shows that the ARIMA was the worst model (RMSE = 89.21), the neural network was the second-best (RMSE = 65.73), and the hybrid was the best model by far (RMSE = 18.02) (Du, 2018).

Siami-Namini et al. (2018) compares ARIMA and LSTM models' ability to forecast both financial and general economic data in their article *Forecasting economic and financial time series: ARIMA vs. LSTM*. The data used consist of six stock prices collected monthly from 1985 to 2018 for the financial forecasts. A range of different consumption metrics, M1 money supply, and trade data was used for the general economic forecasts. The models compared are an ARIMA model and an LSTM model with four units in its LSTM layer.
The results presented by Siami-Namini et al. show that the LSTM model on average has an 87 per cent lower RMSE in financial forecasts and 84 per cent lower RMSE in general economic forecasts. They also show that the amount of iterations in the LSTM-model does not affect the final forecasting accuracy (Siami-Namini et al., 2018).

In their article *Forecasting nigerian stock market return using arima and artificial neural network models*, Isenahd and Olubusoye (2014) similarly conclude that ANNs[8] (RMSE = 0.0150 - 0.1055) outperform ARIMA models (RMSE = 5) when used for forecasting. The

[8]These are referred to as TECH in the paper

neural networks used were feed-forward neural network models, and the ARIMA was a non-seasonal (3,0,1) (Isenahd and Olubusoye, 2014).

Summary

The comparative literature all come to similar conclusions; ANN models outperform ARIMA models in all articles. This out-performance is also not negligible, with the ANNs achieving a significantly lower RMSE in all the literature reviewed.

The existing literature is reviewed, and the theory about the stock market, ARIMA(X), and ANN are presented in the following chapters.

3 Stock market - Theory

Before forecasting the stock market, it is important to establish why the stock market is interesting to economists. It is also prudent to take a more thorough look at the efficient market hypothesis to understand its central concepts.

3.1 The stock markets and the economy

The stock market has an undeniable link to the economy as a whole. It serves as both a way for private consumers to save money that yields higher expected returns and as a way to supplement or entirely replace income. How private consumers savings and income develop is highly dependent on how the stock market develops. If the stock market experiences a sharp fall, it might lower the general consumption for the consumers. On the other hand, a sharp rise in stock prices might boost general consumption since consumers would need to save less of their income to achieve the same amount of total savings. For businesses, the stock market can act as a low-cost way to raise the capital needed for expansion, and the amount of capital that a business can raise depends on the current stock market. Many modern pension funds also use the stock market to both save money and expand their current capital. Like private consumers, the pension funds success is highly reliant on the stock market's performance to ensure their customers' savings. A sharp fall in the stock market might destroy many peoples retirement savings and thereby lessen their future consumption as retirees. Lastly, the stock market has often acted as a way of measuring the general public confidence in the economy since private consumers with negative confidence in the economy would be unlikely to gamble their money in the stock market.

One thing is for economists to know the links between the stock market and the economy, but another is for economists to know how the stock market itself develops. One of the most popular theories economists use to describe the stock market is the efficient market hypothesis.

3.2 The efficient market hypothesis

The ideas behind the efficient market hypothesis have been around for a long time, but the first to truly codify its concepts was Fama (1970). He compiled the theory and litterateur to show that the stock market was, in fact, a highly efficient market that followed a random walk. The efficient market hypothesis gained even more traction when Malkiel wrote his book *a random walk down wall street* in 1973.
The efficient market hypothesis states that information is absorbed in the market incredibly quickly, to a point where any available information concerning a specific stock is incorporated instantaneously into the stocks price (Malkiel, 2016). Another way of viewing this would be to say that changes in stock price from one day to the next is entirely random (Enders, 2015) since new information occur randomly. This being the case, the hypothesis also suggests that there is no available information left to make predictions on the future stock prices (Malkiel, 2016). Though bubbles like the dot-com bubble in the late 1990s and the housing bubble in 2007-2008 seem to contradict the theory that the stock market is efficient, they are in effect examples of when the market is, in fact, efficient. The lead up to the bubbles themselves may be considered irrational, but the fact that the bubbles burst is a sign that the market has the power to self-correct, thereby being efficient (Malkiel, 2016). As the title of Malkiel's book suggests, the efficient market hypothesis simply states that the stock market behaves like a random walk.

3.2.1 Random walks

A random walk is a mathematical way of describing random stochastic trends. In the case of stock price forecasting, it would be assumed that the current price (y_t) is equal to the former price (y_{t-1}) plus a white-noise component (ϵ_t). In a time series, this would mean that changes to the current price are dependent only on the white-noise component, i.e. a random change (Enders, 2015). A random walk can be written as:

$$y_t = y_{t-1} + \epsilon_t \tag{3}$$

As can be seen in this equation, the price will remain constant unless a random change occurs. The equation above is what mathematicians and statisticians refer to as a statistical random walk. The stock market, however, has shown sign of not being a true statistical random walk which has led to some criticism of the theory (Malkiel, 2016).

3.2.2 The efficient market hypothesis' critics

Although the efficient market hypothesis has many proponents, it also has its fair share of critics. Some critics point to the crisis and bubbles the stock market has historically encountered as proof that the market is inefficient (Malkiel, 2003). Others have shown that short-term momentum exists in the market, so that positive news about a stock often leads to a "bandwagon" effect drawing investors to the stock (Malkiel, 2003). This means that rises in a stock price often leads to higher rises, and falls leads to sharper falls. The critics also suggest that the stock market is not perfect at reacting to news and sometimes under reacts. This would mean that a stock is undervalued compared to the true value of the stock (Malkiel, 2003). Another criticism is that the mere fact that professional markets for stock trading exist necessitates the existence of imperfect prices. Grossman and Stiglitz (1980) observes short-run pricing irregularities that would be consistent with an imbalance in information between professional and amateur traders. This would help to explain why a professional market for stock trading exists.

4 ARIMA - Theory

Since ARIMA and ARIMAX will be used as baseline models, the theory concerning these models will be explained in this chapter. In section 4.1 theory concerning stationarity will be explained. ARIMA theory will be explained in section 4.2, while the ARIMA with extra explanatory variables will be explained in section 4.3. Section 4.4 will explain the AIC selection criterion that is used to choose the best ARIMA(X) model.

4.1 Stationarity

In order to be able to forecast a time series using ARIMA models, the series must be stable and predictable. This means that the series needs to be stationary. Several conditions need to be met in order to deem a time series stationary. These conditions state that a time series' mean, unconditional variance, and autocovariance must be independent of time (Enders, 2015).

Time series that stems from raw data are often non-stationary. Raw data tends to have a trend or is seasonally correlated, making it hard to model and forecast. Non-stationary time series, however, can become stationary by differentiating the data (Enders, 2015).

4.2 AutoRegressive Integrated Moving Average

A model often used to forecast a stationary time series is the AutoRegressive Moving Average (ARMA) model. The ARMA model consists of two components; an AR- and an MA-component (Enders, 2015).

The AR component is a linear model that makes the forecast based on lagged versions of the dependent variable. The model is expressed by:

$$AR(p) \quad y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} + \varepsilon_t \tag{4}$$

Where p indicates the number of lags, y_t is the dependent variable, α_0 is a constant, α_i are parameters, and ε_t is a white noise component. (Enders, 2015)

The second part of an ARMA model is the MA component. The MA component describes a model where forecasts are made based on earlier periods' error terms. The model is expressed by:

$$MA(q) \quad x_t = \sum_{i=0}^{q} \beta_i \varepsilon_{t-i} \tag{5}$$

Where q denotes the number of lags, ε_t is a white noise component, and β_i are parameters. (Enders, 2015)

As explained, the ARMA model combines the AR and MA components. The ARMA model is therefore expressed by:

$$ARMA(p,q) \quad y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{i=0}^{q} \beta_i \varepsilon_{t-i}. \tag{6}$$

Such, the ARMA structure is a model that uses earlier values of a variable and model errors to forecast its future values.

As explained in section 4.1, stationarity in the data set is a requirement for forecasting using ARMA models. If the data used for the ARMA model is deemed non-stationary and therefore differentiated, the ARMA is denoted as an AutoRegressive Integrated Moving Average (ARIMA) model. In an ARIMA *(p,d,q)* p and q still denotes the number of AR and MA components, while d denotes how many times the data set is differentiated to become stationary (Enders, 2015).

4.3 AutoRegressive Integrated Moving Average with external regressors

A way to incorporate extra explanatory variables is the AutoRegressive Integrated Moving Average with external regressors (ARIMAX) model. The ARIMAX combines the ARIMA model structure with a standard linear regression model. A standard linear regression model is a strong tool to describe correlations between variables. Linear regression models are often better suited for cross-sectional data, since time series data often, if not always, are autocorrelated ($corr(y_t, y_{t-1}) \neq 0$). Linear time series regressions will result in a biased model because an autocorrelated dependent variable means that the error terms will be autocorrelated as well (Date, 2020).

The autocorrelation problem can be accounted for by combining the linear regression structure with an ARIMA model and thus creating an ARIMAX(p,d,q). The ARIMAX is expressed by:

$$y_t = \phi X_{t-1} + \eta_t \tag{7}$$

Where ϕ is a vector of coefficients, X_{t-1} is a vector containing the explanatory variables at time $t-1$, and η_t is the ARIMA part:

$$\eta_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{i=0}^{q} \beta_i \varepsilon_{t-i}. \tag{8}$$

Such, the ARIMAX structure is a model that uses earlier values of a dependent variable, earlier model errors and lagged versions of a set of explanatory variables to forecast a dependent variable's future values.

4.4 Akaike Information Criterion

Selecting the order of p and q when modelling an ARIMA(X)(p,d,q) often end up in a trade-off between the models simplicity and how well it fits the data. Too high orders of p and q are often associated with overfitted models, where the model is very good at explaining the training data but bad at forecasting data that lies outside of the training data. Too low orders of p and q increase the sum of squared residuals, which is associated with a weak model.

The Akaike Information Criterion (AIC) is a measurement that accounts for both the risk of over-and underfitting. The measurement is constructed so that lower AIC values are associated with a better model. The formula for AIC is showed in equation 9 (Enders, 2015).

$$AIC = T \ln\left(\text{sum of squared residuals}\right) + 2n \tag{9}$$

Where T is the number of usable observations, and n is the number of estimated parameters ($p + q+$ a possible constant term).

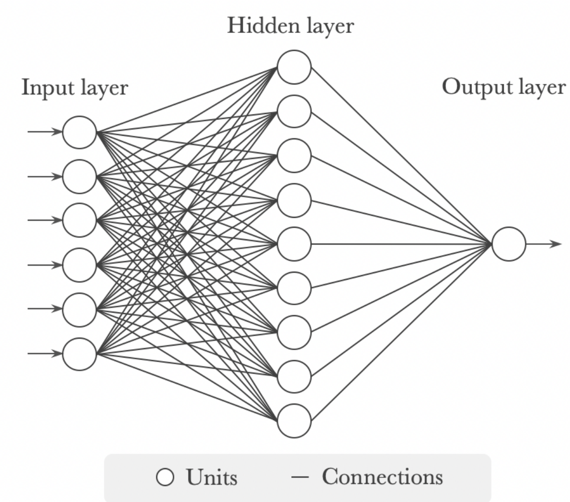
5 ANN - Theory

A variety of artificial neural networks will be used in this thesis. For this reason, the theory needed to understand and create both feed-forward and recurrent neural networks is explained in this chapter. First simple feed-forward neural networks are explained in section 5.1. Recurrent neural networks are explained in section 5.2, and finally, the process of tuning hyperparameters are explained in section 5.3.

5.1 Artificial Neural Networks

An Artificial Neural Network (ANN) is a machine learning method that is able to deal with more complex systems than traditional supervised machine learning methods. An ANN works by creating units and layers that do computations based on input and relay those computations further down the network. The ANN is created this way to imitate the neurons and links in the human brain(Sanderson, 2017a). There are many types of ANNs, but the simplest and most easily used to explain the general features of ANNs is a simple Feed-forward Neural Network (FNN). A simple FNN consists of one input, one hidden, and one output layer. It is important to note that there can be multiple hidden layers. A simple FNN is illustrated below in figure 1:

Figure 1: Simple FNN



Reference: Own creation based on (Sanderson, 2017a)

The first part of an FNN is the *input layer*. This layer consists of one unit for each explanatory variable in the data. Each unit receives the data of one explanatory variable and assigns it a *weight* to determine how important this explanatory variable is to the model. Aside from the weights, each unit also contains a bias and an activation function (Sanderson, 2017a). This is presented in equation 10:

$$\sigma(w_1 a_1 + b) \tag{10}$$

Where σ is the activation function, w is the weight, a is the data, and b is the bias.

The *bias* regulates whether the unit sends data through the network. The bias, input, and weights in each unit are different, but they share the same activation function. The units' *activation function* is used to normalise the output and determine whether the units' output is sent further down the network. Since this thesis uses a regression model, the Rectified Linear Unit (ReLU) function is used. This activation function changes all negative values to zero and does nothing to positive values. This means data is only sent through the network if the unit's output is above 0. The advantage of using ReLU is that it is an efficient activation function that also allows for back-propagation (Sanderson, 2017a). Back-propagation is explained in detail in section 5.1.1. Each unit's result in the input layer are then sent through the network to the next layer, which is the hidden layer (Sanderson, 2017a).

The *hidden layer* differs from the input layer in how the number of units is decided. Instead of being decided based on the number of explanatory variables in the data, the number of units is decided solely at the researcher's discretion. The chosen amount of units is often a trade-off between predictive power and overfitting the model. The units in the hidden layer differ from the units in the input layer in the data they receive. Instead of receiving raw data, they receive the results from all the units in the previous layer. Since they receive more data, they also have to assign more weights (Sanderson, 2017a). This is presented in equation 11:

$$\sigma(w_1 a_1 + w_2 a_2 + w_3 a_3 + ... + w_i a_i + b) \tag{11}$$

The notations are the same as in equation 10

The weights and bias are different in each unit, but the units in the hidden layer not only share an activation function but also shares the same data. The results from the units in the hidden layer are then sent to the output layer (Sanderson, 2017a).

The number of units in the *output layer* depends on the problem investigated. There would be one unit for a regression, and for a classification, there would be units equal to the number of classes predicted. The units in the output layer work the same way as the units in the hidden layer. Since the output layer is the last layer in an FNN, its results are not sent to another layer but are instead the entire model's result (Sanderson, 2017a). The way an ANN, in general, achieves accurate forecasts is by training and changing its weights and biases so that the output of the last layer is closer to the true values.

5.1.1 Training an ANN

Initially, the weights and biases are randomly selected in the model. However, they can be changed to improve the model's performance. The process of changing the weights and biases is often referred to as training the model (Sanderson, 2017c).

When training an ANN, it is important to have a *loss function* to evaluate the model's performance. The loss function describes how far off the correct answer the model's predictions are. In this thesis, the loss function is the RMSE measurement described in section 6.1. The goal of training an ANN is to find a local minimum of the loss function. Reaching this local minimum is commonly done by using *back-propagation* (Sanderson, 2017b). To understand back-propagation, it is helpful to start by considering the output layer, which consists of only one unit. When running a neural network, each observation is sent through all the layers, and the final prediction is the output of the last unit, as shown in equation 11. When an ANN makes a prediction, it is possible to calculate the loss of that prediction and then calculate how the model should be changed to reach the local minimum (Sanderson, 2017c). A model's output can be changed by changing the variables shown in equation 11, namely w, a and b. Changing the units' weight and bias is relatively simple since their initial value is randomly chosen, but changing the unit's input is more complicated. For the first unit in a model, the input is simply the raw data, but for all the following units, the input is the output of all units in the previous layer.

To change the input of the last unit, the output of the second to the last layer needs to be changed. This change is done by back-propagation, which is a method of changing the weights, bias and input in a unit with respect to how it will change the following units and layers (Sanderson, 2017c).

In theory, the most accurate way to reach a local minimum of the loss function is to calculate the loss individually for all observations. This method is often considered too time-consuming and demands too much computational power (Sanderson, 2017c). To solve these problems, an *optimiser* is often used to speed up the process of back-propagation. One solution is to gather the data in batches and send them through the network together, thereby calculating their combined loss. This method is not as accurate but is significantly faster and reaches the same local minimum in the end (Sanderson, 2017c). To further increase the efficiency of the training process, momentum can be added to the model. In this thesis, the Adaptive Momentum Estimation (Adam) method is used to optimise the training. This method has been chosen since it offers the best balance of computational ease and accuracy. One crucial part of any optimiser is the learning rate. The *learning rate* in an ANN is a way to determine how fast a model adapts to new information. A high learning rate would result in a model that often changes its weights based on new information, whereas a low learning rate would result in a model that would be less prone to change its weights (Brownlee, 2020b).

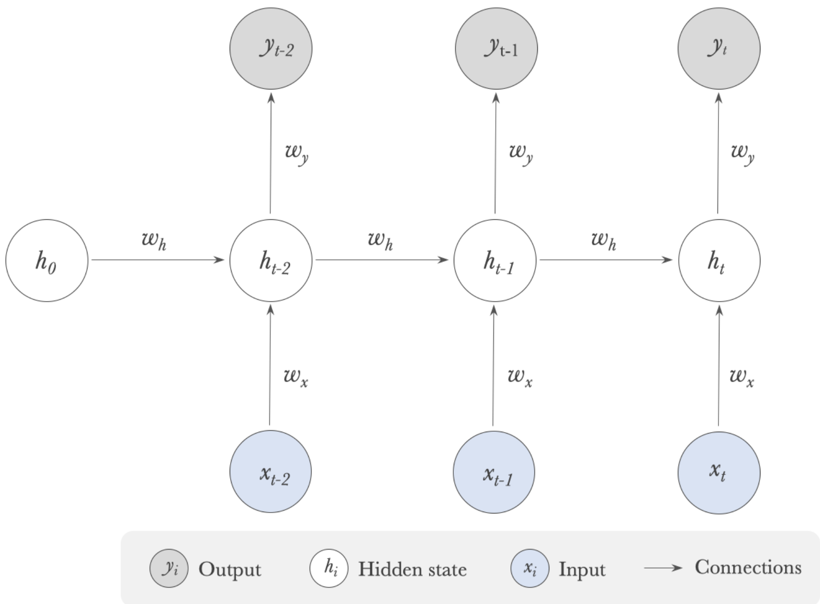
If an ANN is prone to overfit during training, it can be beneficial to add a *dropout layer* to the model. Adding dropout is a way of randomly 'killing' units in the previous layer. In effect, this means that some of the calculations and connections established in the previous layer are not carried forward to the next. Randomly 'killing' units might seem counterintuitive at first, but the process introduces randomness, which is often beneficial to combat overfitting. Thus the model often achieves better results even though it is effectively working with less information (Brownlee, 2019).

One drawback of FNNs is that they cannot use sequential data. This can be fixed by employing an RNN.

5.2 Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a type of ANN that deals with sequential data, like time series. The major difference between an RNN and an FNN is that the former incorporates memory about earlier instances in the data. This is done by adding a hidden state to the units. The *hidden state* transfers what has been learned earlier in the sequence to the later parts (Pi, 2018b). The way an RNN is used is to place RNN layers in the first part of FNNs. This means that an RNN does not necessarily consist of only recurrent layers but rather that the first layer or layers have a recurrent structure. It is also possible to have feed-forward layers after a recurrent layer to further strengthen the model's predictive ability. An example of a recurrent layer can be seen in figure 2.

Figure 2: Recurrent layer



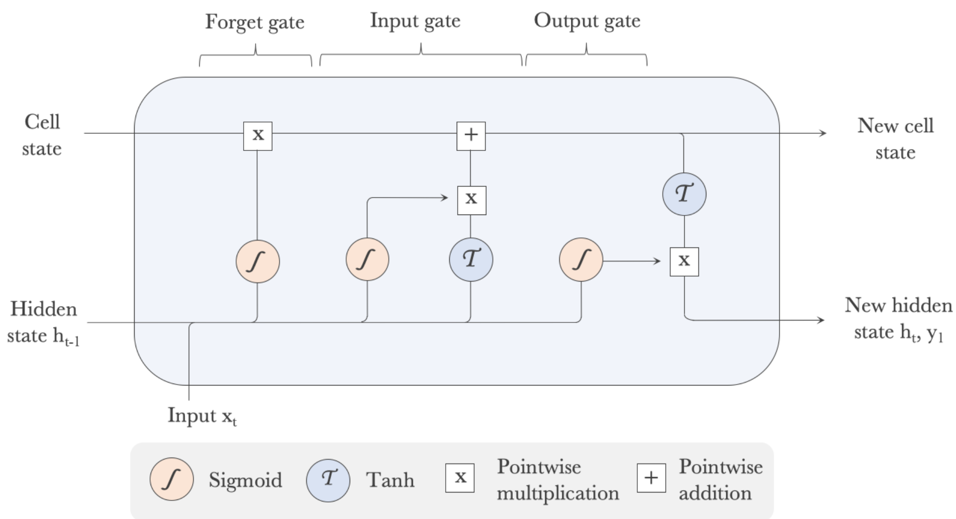
Reference: Own creation based on (Venkatachalam, 2019)

As illustrated in figure 2, the units assign weights to not only the output of the previous unit but also the output of the earlier step in the sequence. The weight w_h represents the memory transfer and allows the RNN to retain a memory of the sequence. One drawback of this simple way of creating RNNs is that it suffers from short-term memory. This short-term memory occurs because of the *vanishing gradient problem*. The vanishing gradient problem occurs because the earliest layers of an ANN often experiences the least amount of change during back-propagation. Thus, the last layers are often weighted higher when the ANN makes its final prediction, and for RNNs, only the most recent data hold any significant weight in the prediction (Pi, 2018b). There are ways to alleviate this by using more advanced architectures such as LSTM and GRU.

5.2.1 Long-short-term memory

A way to alleviate the vanishing gradient problem is to use a type of RNN called the Long-Short-Term Memory (LSTM) architecture. The LSTM architecture utilises a gate structure that makes it possible to control what is remembered and forgotten by the network (Pi, 2018a). An example of this gate structure is illustrated below in figure 3.

Figure 3: LSTM gate



Reference: Own creation based on (Pi, 2018a)

In the LSTM architecture, *the cell state* is used to transport information through the sequence. It acts as the LSTM architecture's memory.

The first part of the LSTM is the *forget-gate*. This gate receives the cell and hidden state from the previous part of the sequence. It also receives input from the current part of the sequence. The input and hidden state are passed through a sigmoid function that creates values between 0 and 1. Values closer to 0 should be forgotten, and values closer to 1 should be remembered. After being passed through the sigmoid function, it is multiplied by the cell state. Thereby the model chooses how much of the earlier cell state should be kept.

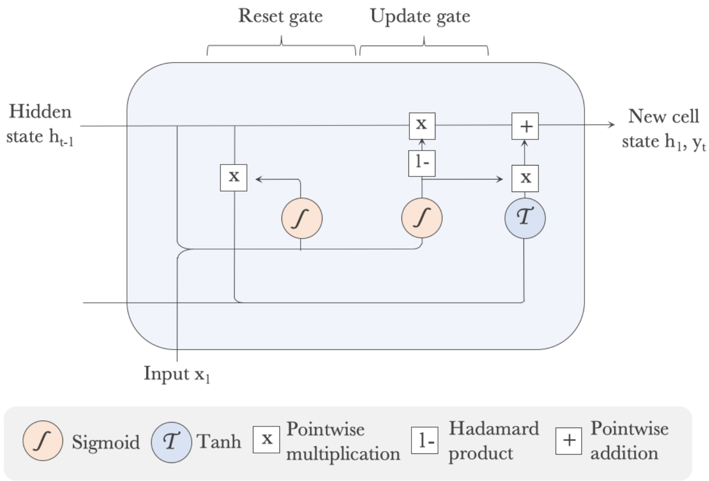
The next part of the LSTM is the *input gate*. In this gate, the hidden state and input are passed through both a sigmoid and tanh function. The sigmoid function again controls what is kept and what is forgotten. The tanh function ensures that the input values do not become unmanageable by limiting them to be between -1 and 1. The sigmoid and tanh function results are then multiplied to determine what the model should keep from the current input and then added to the cell state.

The last part of the LSTM is the *output gate*. In this gate, the hidden state and input are first passed through a sigmoid function. Then the cell state is sent through the tanh function. The transformed input and hidden state are multiplied by the cell state to determine what should be kept for the output. The result is a new hidden state that is passed on to the next part of the sequence. The cell state is similarly passed on to the next part of the sequence. The new hidden state also acts as the output for the current part of the sequence.

5.2.2 Gated Recurrent Unit

Another way of dealing with the vanishing gradient problem is to use a Gated Recurrent Unit (GRU) architecture. A GRU is similar to an LSTM, but it foregoes the cell state and instead only uses the hidden state to transfer information. As seen in figure 4 a GRU only has two gates: a reset gate and an update gate.

Figure 4: GRU gate



Reference: Own creation based on (Pi, 2018a)

The GRU works by first sending both the new input and a hidden state through a sigmoid function in the *reset gate*. The result of this is then multiplied with the hidden state to decide what information from the past should be used. Next is the *update gate*; here, the input and hidden state is passed through another sigmoid function and added to both the past hidden state and to the input. The update gate acts as a way of deciding what information from the current input should be used to update the current and new hidden states. Lastly, the input is passed through a tanh function and updated with the information from the update gate. This new input/hidden state is then multiplied with the current hidden state to achieve the output of the gate. Like in an LSTM, this output acts as both the output of the gate and as the hidden state for the next gate in the sequence. Another trait shared between the GRU, and LSTM architectures are their ability to be bidirectional.

5.2.3 Bidirectionality

Bidirectionality means that the sequence is passed through the recurrent layer twice, once from the first part of the sequence to the last and once from the last part of the sequence to the first. The output of each of these is then calculated together to reach the final output of a gate.

The advantage of bidirectionality is that it sometimes picks up on connections that the

RNN might otherwise overlook early in the sequence.

5.3 Hyper parameters

A crucial part of improving the performance of any machine learning model is to tune its hyperparameters. It is important to note that the hyperparameters are different from the trainable parameters like biases and weights. The hyperparameters are defined when the model is constructed and does not change during training. In the case of an ANN, these hyperparameters include the number of units, the number of layers, activation functions and more. The tuning process entails changing these parameters, either manually or automatically, to achieve a better model. For the FNNs and RNNs used in this thesis, the hyperparameters that are changeable is the number of units, standard feed-forward layers, dropout rate, activation function and learning rate. However, for the LSTM and GRU layers and the bidirectional versions of these, the activation functions are limited to the sigmoid and tanh functions described in sections 5.2.1 and 5.2.2. These limitations are imposed in order to utilise GPU acceleration[9] which speeds up the training and tuning process that would otherwise take days to complete.

In order to evaluate the ANN and ARIMA(X) models' performance, different evaluation methods will be used. These are explained in the next chapter.

[9]This is done through NVIDIA's CUDA framework and moves the tuning and training from the CPU to the GPU, which is often more powerful.

6 Evaluation methods

Different evaluation methods will be used to evaluate and compare ARIMA(X) and ANN models' ability to forecast financial assets. RMSE and MAPE will be used to both choose the best models to forecast and to evaluate these forecasts. The mathematics and intuition of these is explained in sections 6.1 and 6.2. The Diebold-Mariano test will be used to measure whether there is a difference accuracy of the forecasts produced by the ARIMA(X) and the ANN models. The method behind the Diebold-Mariano test is explained in section 6.3.

6.1 Root Mean Square Error

Root Mean Square Error (RMSE) is used to test the models' abilities to forecast. RMSE is also used as the loss function when training the ANN models.

RMSE describes an average deviation between the foretasted and the actual values, with a heavier weight on large deviations and a lighter weight on smaller deviations. The formula for calculating RMSE is shown in equation 12.

$$RMSE = \sqrt{\sum_{t=1}^{n} \frac{(y_t - \hat{y_t})^2}{n}} \tag{12}$$

Where y_t denotes the actual value at time t, $\hat{y_t}$ denotes the forecasted value at time t, and n denotes the number of forecasts.
The deviation is squared in the formula, which secures that RMSE always is presented in absolute values. This also has the before mentioned effect that large deviations get a heavier weight. The RMSE statistic is constructed so that it cannot be used to evaluate forecasts across different dependent variables. This is the case since the statistic's size is correlated with the size of the dependent variable.

6.2 Mean Absolute Percentage Error

Another way of calculating an average deviation between forecasted and actual values is the Mean Absolute Percentage Error (MAPE). It differs by calculating the difference in percentage instead of in level and that it weighs deviations equally regardless of size. The formula for calculating MAPE is shown in equation 13.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}|\frac{y_t - \hat{y}_t}{y_t}| \tag{13}$$

The notations are the same as in equation 12.

The advantage of the MAPE statistic is that it can be used to evaluate forecasts across different dependent variables. The dependent variables in this thesis are different Danish stocks, and the MAPE statistic makes it possible to evaluate the forecasts across these different stocks.

6.3 Diebold-Mariano test

The Diebold-Mariano (DM) test is used to investigate whether the difference in accuracy across models is significant. The DM test tests the difference by an ordinary student's t-test. Since the models in this thesis always use one step ahead forecasting, the DM test statistic can be calculated as:

$$DM = \frac{\bar{d}}{\sqrt{\frac{\gamma_0 + 2\gamma_1 + ... + 2\gamma_q}{(H-1)}}} \tag{14}$$

Where H represents the number of forecasts used to calculate $\bar{d}$, γ_i is the i'th autocovariance of the d_t sequence, $\bar{d}$ represents the mean loss difference between model 1 and model 2:

$$\bar{d} = \frac{1}{H}\sum_{i=1}^{H}[g(e_{1i}) - g(e_{2i})] \tag{15}$$

Where $g(e_{1i})$ and $g(e_{2i})$ is the errors at time i, in model 1 and 2 respectively.

As shown in equation 14, the DM statistic is calculated by the mean of the loss differences between model 1 and 2, and the standard deviation of that mean $\left(\sqrt{\frac{\gamma_0 + 2\gamma_1 + ... + 2\gamma_q}{(H-1)}}\right)$. It is clear from the formula that if one or more γ_i are negative, the standard deviation of $\bar{d}$ could become negative. In order to counter such circumstances, Newey and West standard deviation is calculated and used.

The last step is to compare the DM statistic with critical values obtained from a student t-test with $1 - H$ degrees of freedom. For this thesis, the null hypothesis is that model 1 does not have more accurate forecasts than model 2 (Enders, 2015).

With the theory in place, the next chapter will explain how it will be used and implemented throughout this thesis.

7 Methodology

In this chapter, the overall methodology for this thesis will be presented. To answer the problem statement *How well do artificial neural network models predict future prices of the Danish stock market?*, twelve different ANN's will be trained for each of the six Danish stocks investigated. These twelve models include six models that only use earlier prices to forecast and six models that use a set of explanatory variables. The selection of these explanatory variables is explained in section 7.2. Out of these twelve, only one model will be used to make the actual forecasts. The selection process for choosing this model is covered in section 7.5. To analyse the forecasts of the ANN models, one ARIMA(X) model for each stock is trained to forecast the same stocks in the same period. The process of finding the best ARIMA(X) model is explained in section 7.4. The selection of both the best ANN and ARIMA(X) follow an out-of-sample test strategy that is explained in section 7.1. Because this thesis uses data from different stock exchanges, several missing values occur in the data. The process of dealing with the missing values is explained in section 7.3. The empirical analyses are performed using the coding language R in the program RStudio[10].

7.1 Out-of-sample testing

The selection of both the best ARIMA(X) and ANN model follow an out-of-sample testing strategy. In out-of-sample testing, the data is split up into two parts: Training and testing. The models are first trained using the training part of the data, and then to generalise the model, it is tested using new data from the testing part of the data. The chosen time period and the division into training and testing periods are explained later in section 8.1.

7.2 Explanatory variable selection

Not all of the explanatory variables later described in section 8.3 are used when forecasting the dependent variables. The explanatory variables are selected by using a Granger

[10] The codes and overall data manipulation can be found at https://github.com/andreasbj77/Master-thesis

Causality test for each dependent variable. The Granger Causality test examines whether one time series can be used to predict another time series. A lower p-value is associated with a more significant explanatory power (Enders, 2015). For this thesis, the Granger Causality test examines five lags and uses a critical value equivalent to a p-value of one per cent. The one per cent critical p-value was decided after first attempting to use a five per cent critical value. This resulted in poorly performing models in the out-of-sample test, so it was decided to lower the critical p-value to one per cent. If an explanatory variable meets the one per cent criteria, the variable will be included in the model.

7.3 Missing values

Missing values (NA) occur in the data used in this thesis. The primary reason for the occurrence of NAs is that the markets across the world have different trading days due to, e.g. holidays. Neither the ARIMA(X) nor the ANN approaches allow NAs. NAs do not occur in the dependent variables since they are all traded on the Copenhagen stock exchange. This means that NAs are only a problem in the explanatory variables.

For this thesis, the NAs is replaced using *interpolation*. Interpolation is a statistical approach to find an unknown data point between two existing points. The method relies on the assumption that observations do not differ significantly from each other. Since this thesis uses day-to-day stock prices, the daily change in the stock price is somewhat limited. Since stock market prices often do not develop linearly, the method *spline* is chosen. Spline imputes the missing data using a cubic polynomial between the existing points. This means that the missing data can be imputed while still keeping the non-linear form characterising the stock market.

Generally, imputing NAs is not a perfectly precise method. Imputing NAs is often more precise when they occur individually rather than when they occur in long sequences. Therefore the NAs distribution in the data was investigated before imputing. All NAs in the data occur individually or in sequences of two, which means that the errors created when imputing are kept to a minimum.

7.4 ARIMA(X) selection method

Two different types of ARIMA models will be trained for each dependent variable, a standard ARIMA and an ARIMAX model. The selection of which models will be used for forecasting is divided into two parts. In the first part, the R function Auto.Arima() is used to choose the optimal model structures based on AIC. The function is restricted to only look at stationary models. Eight model structures will be chosen for each dependent variable based on AIC: Four ARIMA models and four ARIMAX models.
In the second part, the eight model structures will be tested using the testing period. The eight different ARIMA(X) models are used to forecast the testing period using a one-step-ahead forecasting method. The models are then evaluated using RMSE and MAPE, where the one with the lowest score is chosen as the model to produce the actual forecasts. This selection process means that only one model is chosen to forecast each dependent variable.

7.5 ANN selection methods

Twelve different ANN model types will be trained for each of the six dependent variables. These twelve are made from six different architectures, which are: An FNN, a simple RNN, an LSTM, a GRU and a bidirectional version of both the LSTM and the GRU models. Each of these six architectures is trained both with and without extra explanatory variables, meaning that twelve model types will be trained in total for each of the six dependent variables. The creation and selection of ANN models are made using the training and testing data described in section 7.1.
First, the models are created by tuning the hyperparameters of each individual model. This tuning is done automatically through the KerastuneR package in R. This method randomly selects values from a pre-defined search space[11] and trains models, using the training period, for a different amount of epochs. This process is repeated for all model types within each dependent variable and leads to 150 individual models that are then compared, and the best model is then selected as the best of its type. With twelve model types per dependent variable and six dependent variables, this selection process results

[11] These can be found in Appendix 3 - Hyperparameter tuning

in $10,800$[12] models in total across all model types and dependent variables.
After creating the best model for each of the twelve model types, the models are compared using the testing period. The models forecast the testing period using a one-step-ahead method, and their results are compared using the RMSE and MAPE statistic. Based on these results, the best model for each dependent variable is selected.

With the theory and methodology established, the last thing needed before starting to answer the problem statement is to select which data will be used in this thesis. This is done in the following chapter.

[12] 150 models $\cdot$ 12 model types (6 with explanatory variables and 6 without) $\cdot$ 6 dependent variables $= 10,800$ models in total

8 Data selection

This chapter describes how, when, and what data is selected[13]. The six dependent variables are chosen from the C25 stock index. How and what stocks are chosen are explained in section 8.2. The time period and data frequency chosen in this thesis are daily stock prices from the start of 2010 to the end of 2019. The time period is further divided into training, testing, and forecasting periods. The reasoning behind choosing this time period and frequency are presented in section 8.1. The explanatory variables tested by a Granger causality test in this thesis include all the stocks from the C25 index, stock indices, government bonds, currency conversion rates, and gold and oil futures. The reasons for including these are explained in section 8.3.

8.1 Time period

The time period for this thesis ranges from the 1st of January 2010 to the 30th of December 2019. The time period is divided into three sub-periods: a training period, a test period and a forecast period. Figure 5 illustrates the division of the different time periods.

Figure 5: Illustration of time period

Reference: Own creation

The training period ranges from the 1st of January 2010 until the 29th of December 2017, represented by the black line. The blue line represents the test period on figure 5 and range from the 1st of January 2018 until the 28th of December 2018. Lastly, the forecast period ranges from the 2nd of January until the 30th of December 2019 and is represented by the red line.

[13]The Data used in the thesis can be found at https://github.com/andreasbj77/Master-thesis/tree/main/Data

This specific time period is chosen to avoid obvious structural breaks from economic crises. By choosing a period from 2010 to 2019, the structural breaks from the financial crisis in 2008 and the corona crisis in 2020 are avoided.

For this thesis, daily stock prices are chosen. The choice of daily prices means that the models have enough data to train properly, without adding bias from the imputation of too many missing values. Hourly frequency is available for most of the data and would give the models significantly more data for training, which could benefit the ANN models. However, this thesis contains data from different stock exchanges worldwide, which have different time zones and therefore different opening hours. This disparity in opening hours would result in a huge amount of missing data points which could bias the models when imputed. Weekly frequency would mean much less data, and in the period chosen, this would not be enough data for proper training, testing and forecasting periods. As such, a daily frequency is used in this thesis.

8.2 Dependent variables

The dependent variables selected for this thesis is stocks selected from the Danish C25-index in 2018, which contains the 25 most traded Danish stocks. 2018 is chosen so that the selection is not influenced by the forecasting period (2019). Only stocks that have been publicly traded from 2010 to 2019 is considered. The stocks in the index are divided into sectors as defined by the Danish newspaper *Børsen*. One stock from within each sector is chosen based on the highest stock price on the last day of the testing period. One stock from each sector is chosen in order to cover as many different stock types as possible.

The stocks chosen as the dependent variables for this thesis are: *Carlsberg*, *Genmab*, *Jyske Bank*, *A.P. Møller - Mærsk B*, *SimCorp*, and *Vestas Wind Systems*.

All the stock prices are gathered directly from Yahoo finance. The closing price for each day will be used as both the dependent forecasting target and explanatory lagged values. The stock price of all stocks in the C25-index at the starting date of the forecasting period can be found in *Appendix 2 - Dependent variables*

8.3 Explanatory variables

In addition to past values of the dependent variables, this thesis will also include several other explanatory variables. These explanatory variables have all been selected to study whether additional information improves the models' performance when forecasting. As with the dependent variable, the closing price of the extra variables will be used as the price when relevant.

As described in section 7.2, all explanatory variables are not used to forecast each dependent variable. The explanatory variables are instead selected based on a Granger causality test for each of the dependent variables. The p-values from the Granger causality tests can be found in *Appendix 1 - Granger Causality test*. The values written in bold represent which explanatory variables are included in the models for each dependent variable. What variables are tested, and the reasoning for doing so is described below.

Stocks in the C25 index

All available stocks within the Danish C25-index are used as explanatory variables.[14] The stocks in the C25-index should provide a good indication of the stock market's general performance, and they should also be an indicator for the performance within each sector. The correlation between the stocks and the dependent variables might be positive if the stocks are of companies that the dependent variables rely on in their supply chain or if they simply show a general rise in stock price within the sector. They might also be negatively correlated if the stocks are of a company that is in direct competition with the dependent variables.
All stock data is collected from Yahoo finance.

C20- and SP500-index

The stock indices C20[15] and SP500 are used as extra explanatory variables. These are included to provide information on the general performance of the Danish and the American stock markets. The SP500 index is often used as a proxy for the American stock market, which accounts for 56 per cent of the total global stock market (Statista, 2021), as to why it is a good indicator of the general tendencies of global stock markets. The

[14]Due to data limitations, not all stocks in the index are included
[15]Due to data limitations, the C20-index will be used instead of the C25-index

C20 index should be a good indicator of the general performance of the Danish stock market. It is expected that both stock markets are positively correlated with the dependent variables since a rise in the general market oftentimes means a rise in the individual stock prices.
The SP500 is from Yahoo finance, and the C20 is collected from investing.com.

2- and 10-year government bonds

The interest rate on 2- and 10-year Danish government bonds are included to provide information on the expectations of the agents acting in the market. The rationale behind including the government bonds is that they should act as both an alternative good to the stock market and a proxy of the market expectations. The interest rate of 2-year government bonds acts as a proxy for the short-term expectations, and the 10-year government bonds act as a proxy for the long-term expectations. In both cases, a rise in the interest rate is expected to be an indicator of worsening expectations of the potential return on stocks. This is due to government bonds being an alternative good to investments in stocks, so a fall in demand for stocks would lead to a rise in demand for government bonds. For those reasons, it is expected that the interest rate of both 2- and 10-year government bonds will be negatively correlated with the dependent variables.
Interest rates on 2- and 10-year bonds are collected from investing.com.

Currency conversion rates

Since the Danish economy is predominantly export-focused, the currency conversion rates of the most significant Danish export receivers is included as an explanatory variable. These are the USA, Sweden, Norway, The UK, and China (Udenrigsministeriet, 2018). Countries in the euro-zone are not included since Denmark has a fixed exchange rate policy with the euro. The currency exchange rate is stated in the value of one foreign currency in Danish kroner. These exchange rates act as a gauge for the relative price level between Denmark and its most important trading partners. A rise in the value of a foreign currency compared to the Danish krone would mean that it would be cheaper to buy Danish goods, all other things being equal. Thus the expectation is that the correlation between foreign exchange rates and the dependent variables are positive. There might also be a negative correlation between certain exchange rates and certain stocks if

a Danish company depends on imported raw resources for their production.
All currency exchange rates are collected from The National Bank of Denmark's StatBank.

Gold and oil futures

Gold and oil futures are included to gain information on alternative goods. Gold is especially known for being a 'safe haven' for investments during economic downturns. The expectations for gold futures are that their price should rise when the stock market shows signs of instability. Therefore gold futures are expected to have a negative correlation with the dependent variables during economic downturns or even a positive correlation if the stock market is overheated.
Oil can be an alternative investment and a key part of many companies production or supply chain. When the expected oil prices rise, it is expected that more people switch from investing in stocks to investing in oil futures, given their greater potential return. A rise in oil prices might also mean a rise in production cost for companies relying on oil. This means that a rise in oil prices is expected to be negatively correlated with stock prices.
The gold futures are collected from investing.com, and the oil futures are from Yahoo finance.

8.4 Summary

The dependent variables are all from the C25 index and are chosen based on sectors and the stock's price. The dependent variables chosen to forecast include: Carlsberg, Genmab, Jyske Bank, A.P. Møller - Mærsk B, SimCorp, and Vestas Wind Systems. This thesis seeks to forecast the dependent variables daily stock price in 2019. The models used for forecasting are trained using daily stock prices from 2010 to the last day of 2017, while the models' forecasts are tested and validated using data from 2018. The extra explanatory variables considered is based on a theoretical correlation with the dependent variables. The considered explanatory variables include: The stocks from the C25 index, stock indices, government bonds, currency conversion rates, as well as gold and oil futures. In the following chapter, the chosen data will be examined and analysed.

9 Exploratory data analysis

Before creating the models, each dependent variable is investigated and analysed. The purpose of the analysis is to characterise and compare the dependent variables to get a general sense of the data. As explained, the selection method for the ANN and ARIMA(X) models follow an out-of-sample testing strategy. This means that the models chosen to forecast the forecasting period are the models that made the most precise forecasts of the testing period. Therefore, one could assume that models for variables where the testing and forecasting period are similar would produce more accurate forecasts compared to variables where the periods are dissimilar. One thing that characterises the market as a whole is that the stock prices tend to rise over the training and forecasting periods. The prices in the testing period, on the other hand, tend to resemble a random walk. Thus, better models at forecasting the testing period might not be the better models for forecasting the forecasting period.

First, each dependent variable's augmented Dickey-Fuller p-value is investigated to see if the models are stationary. Then each dependent variable is analysed individually. The testing, training and forecasting period, as well as the entire period as a whole, are analysed. A table of descriptive statistics and a plot of the stock price are presented for each individual dependent variable. The training period in the figures are shown in black, the test period is printed in blue, and the forecasting period is shown in red.

Stationarity

As described in section 4.1, the ARIMA approach states that the data should be stationary when forecasting. This is tested by the Augmented Dickey-Fuller (ADF) test, which tests if the data has a unit root. The ADF statistics for the training periods of the dependent variables are printed in table 2.

Table 2: Augmented Dickey–Fuller p-value

	Carlsberg	**Genmab**	**Jyske Bank**	**Mærsk B**	**SimCorp**	**Vestas**
P-value	0.30	0.59	0.48	0.53	0.32	0.44
P-value (diff. data)	>0.01	>0.01	>0.01	>0.01	>0.01	>0.01

Own creation based on calculations in R. Values are rounded for convenience

The test results show that all the six dependent variables are non-stationary, with p-values between 30-60 per cent. Table 2 also shows the results from ADF-tests after the variables are differentiated. The results show that all the dependent variables are stationary after differentiating the data series, all with p-values under one per cent. Since the dependent variables are non-stationary without differentiating but become stationary when differentiated, one would assume that an ARIMA(p,1,q) is the optimal ARIMA structure.

Carlsberg

Carlsberg has the lowest average daily change, both when looking at training, testing, forecasting, and the entire period as a whole. This does not speak to whether the Carlsberg stock price changes over the whole time period but speaks more to the size of the day to day changes. The test period has the lowest standard deviation and the lowest average percentage change, which also becomes apparent when looking at figure 6. The price in the testing period seems to develop somewhat randomly without any drift or trend, while the development of the forecasting prices shows a clear upward trend.

Table 3: Descriptive statistic for Carlsberg

	Mean	**Standard Deviation**	**Average daily change (%)**
Training	555.17	83.68	1.09
Test	739.40	29.06	0.71
Forecast	895.82	88.65	0.86
Entire period	607.22	136.51	1.03

Own creation based on calculations in R. Values are rounded for convenience

The apparent difference in price development could mean that the better econometric model at forecasting the testing period might be a random walk without drift, while the better model at predicting the forecasting period is a model with drift. The same problem could arise for the chosen ANN model since the model is chosen based on having the most precise testing period forecasts.

Figure 6: Carlsberg stock price

Own creation in R

Genmab

Genmab has the second-highest daily percentage changes. The overall high percentage changes primarily come from the training and testing period, whereas the volatility is much lower in the forecasting period as seen in table 4.

Table 4: Descriptive statistic for Genmab

	Mean	**Standard Deviation**	**Average daily change (%)**
Training	448.65	478.54	1.87
Test	1082.86	114.22	1.76
Forecast	1251.29	170.88	1.27
Entire period	591.21	520.78	1.80

Own creation based on calculations in R. Values are rounded for convenience

The models are chosen based on the test period, so they might have some of the same problems as the Carlsberg models when making their forecasts. The price development in the test period seen in figure 7 looks somewhat like a random walk, with a slight downward trend, while the stock price in the forecast period has a very clear upward trend. The development in the training period and the forecasting period look similar, but this might play a minor role since the final model selection is solely based on out-of-sample forecast errors.

Figure 7: Genmab stock price

Own creation in R

Jyske Bank

Unlike Carlsberg and Genmab, Jyske Banks full time period does not seem to have a clear trend. The stock price starts at around 200 in 2010 and ends around 240 in 2019 as seen in figure 8. There is some change in the price when looking at the individual periods. The training period starts with a small rise in the first year's stock price, followed by a small decline the following year. The rest of the training period is characterised by an overall rise in the stock price, with some declining sub-periods.

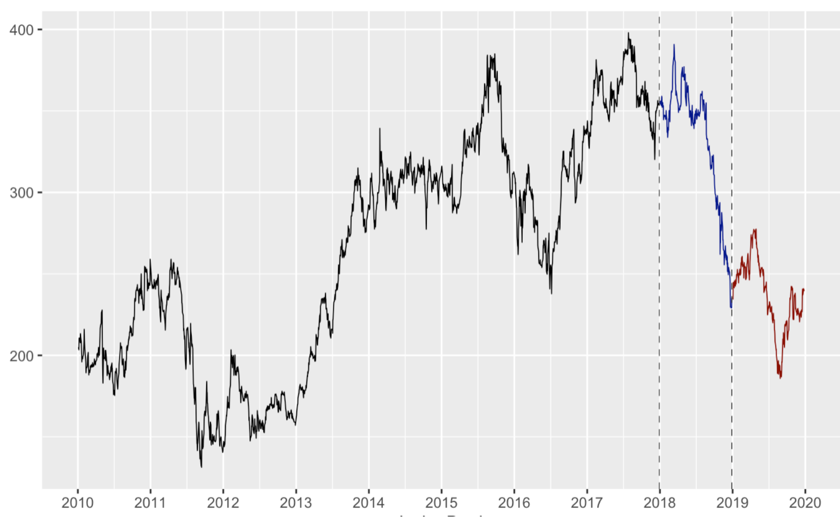
Table 5: Descriptive statistic for Jyske Bank

	Mean	**Standard Deviation**	**Average daily change (%)**
Training	263.34	70.12	1.23
Test	330.86	37.98	1.00
Forecast	235.79	21.89	1.17
Entire period	607.22	136.51	1.03

Own creation based on calculations in R. Values are rounded for convenience

What also separates the development of Jyske Bank's stock price from Carlsberg and Genmab is that the testing and forecasting periods show signs of the same trend. Both periods are characterised by a fall in the stock price, though with a larger fall in the testing period. Because the periods are a lot alike, one could expect that the selected ARIMA(X) and ANN models would produce more precise forecasts than if only one of the periods had a downward trend.

Figure 8: Jyske Bank stock price



Own creation in R

Mærsk B

The Mærsk B stock price development as a whole is reminiscent of Jyske Bank. As seen in figure 9 Mærsk B has a versatile stock price, and if it were not for the last months of 2019, it would end at the same price level as it started at. There is no clear trend in the stock price development. The stock price rises a lot between 2012-2015, like in the case of Jyske Bank, but falls to its earlier level in the next couple of years.

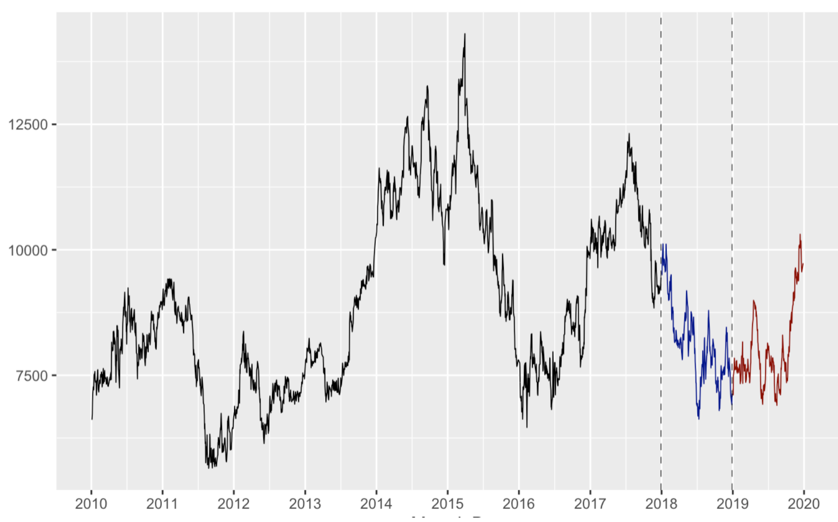
Table 6: Descriptive statistic for Mærsk B

	Mean	**Standard Deviation**	**Average daily change (%)**
Training	8930.32	1800.47	1.38
Test	8127.31	779.17	1.73
Forecast	8017.89	775.13	1.48
Entire period	8760.10	1683.78	1.42

Own creation based on calculations in R. Values are rounded for convenience

What separates Mærsk B and Jyske Bank is that, while Jyske Bank had similar test and training periods, the price development for Mærsk B is different in the two periods. A down-turned trend characterises the test period. However, the prices in the forecast period resemble a random walk the first eight months, after which the prices see a considerable spike the next four months with a rise of over 33 per cent. This difference could mean that the models chosen will produce weak forecasts with relatively high MAPEs.

Figure 9: Mærsk B stock price



Own creation in R

SimCorp

The SimCorp stock price seen in figure 10 develops much like Carlsberg and Genmab, with a clear upward trend throughout the entire period.

Table 7: Descriptive statistic for SimCorp

	Mean	**Standard Deviation**	**Average daily change (%)**
Training	205.77	111.78	1.23
Test	472.77	71.68	1.42
Forecast	615.29	62.39	1.28
Entire period	272.88	172.52	1.26

Own creation based on calculations in R. Values are rounded for convenience

Both the testing and forecasting periods experience significant rises in the stock prince at the beginning of the periods, followed by a downwards trend. The most significant difference between the two periods is that the stock price in the forecasting period takes a big swing upward at the end of the period.

Figure 10: SimCorp stock price

Own creation in R

Vestas

Vestas has the highest average percentage change in both the training period and the entire period as a whole, as seen in table 8. The test and forecast periods' average percentage change, however, are somewhat lower. This can also be seen in figure 11. The Vestas' stock price starts the training period with a downward trend for the first 2-3 years, followed by an upward trend until three months before the period ends, where the stock price drastically falls.

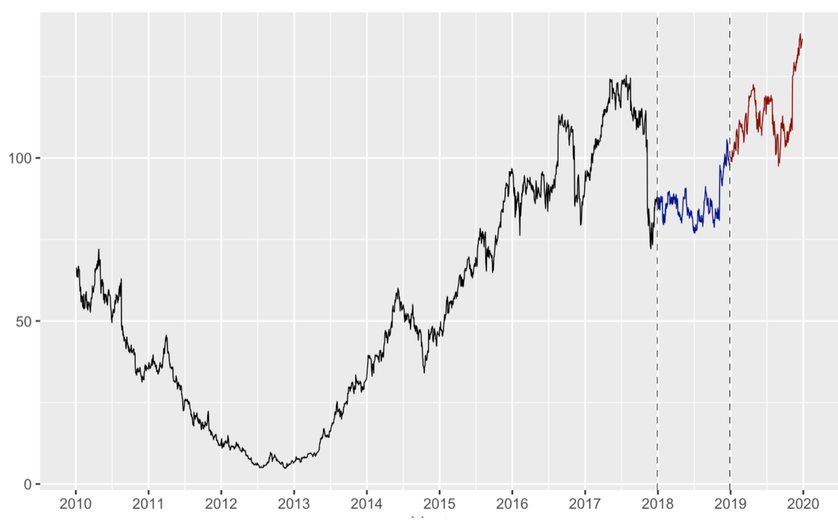
Table 8: Descriptive statistic for Vestas

	Mean	**Standard Deviation**	**Average daily change (%)**
Training	52.67	34.75	2.15
Test	85.98	5.83	1.45
Forecast	113.11	9.08	1.30
Entire period	61.97	36.95	1.99

Own creation based on calculations in R. Values are rounded for convenience

The price developments in the test and forecasting period follow the same patterns, where the first nine months of the periods resemble a random walk, followed by a significant rise in the prices. The irregularities in how the stock price moves might be something that the ANN model can pattern and pick up on, while the linearity of the ARIMA(X) models could make it harder for them to predict.

Figure 11: Vestas stock price



Own creation in R

9.1 Summary

The dependent variables were investigated by an ADF test in order to check if the data is stationary. None of the dependent variables are stationary, but after being differentiated once, they all become stationary.
Both Carlsberg and Genmab have vastly different price developments in their testing and forecasting periods. This could lead to the model chosen in the testing period not being the optimal model for forecasting the forecast period. SimCorp shows the same signs, but the periods are more similar. Vestas' testing period resemble a random walk, with an upward spike in the last months. The same patterns apply to the forecasting period. This is something the ANN models might be able to pick up on. Jyske Bank and Mærsk B do not seem to follow any clear trend. Though Jyske Bank seems to have similar training and forecasting periods, which could make one expect more precise forecasts.

Each stock price has been investigated and analysed. The best ARIMA(X) and ANN models are selected and evaluated in the following chapters.

10 Selection of ARIMA(X) models

The best ARIMA(X) models for each dependent variable will be selected in this chapter. The detailed description of the method used to choose the suitable ARIMA(X) models for forecasting is described in section 7.4. In essence, the method is split up into two parts; first, eight models are chosen, four ARIMA and four ARIMAX, based on the models' AIC. The eight models that were chosen and their respective AICs are presented in table 9. These eight models are used to forecast the testing period. Then the best model out of those eight is chosen based on which model has the lowest RMSE and MAPE. The evaluation statistics are shown in table 10, where the model with the lowest evaluation statistics for each dependent variable is written in bold. The selection process for each dependent variable will be explained one by one in this chapter.

Carlsberg

The explanatory variables chosen for the ARIMAX models for Carlsberg are the lag of the individual stock prices of Danske Bank and DSV, the Oil price and the American SP500 stock index. The reasoning behind choosing these is described in section 7.2.

When looking at the models of Carlsberg, the stock price is differentiated in the ARIMA structures since they all follow a (p,1,q) structure. In contrast, all the ARIMAX models follow a (p,0,q) structure where the stock price is not differentiated. The Carlsberg stock price itself is, as shown in chapter 9, non-stationary, as to why one would expect models that follow a (p,0,q) structure would be non-stationary as well. The code used to train the model and calculate AICs is restricted to only look for stationary models. To be certain that the models are stationary, the unit roots has been examined. The roots are inside the unit circle, which deems the models stationary. There can be many reasons why the ARIMAX models are stationary without differentiating the non-stationary dependent variable. Carlsberg's stock price data could suffer from structural breaks, making it non-stationary, which the explanatory variables equalise. This thesis will not investigate this matter further since it lies beyond the scope of the problem statement.

When comparing the AICs, one would expect that the ARIMAX(0,0,28) is the better model for forecasting Carlsberg's stock price. The AICs of the eight models are very similar, as to why the out-of-sample testing might end up with different results.

The results from the out-of-sample testing show that the ARIMA models, which has a MAPE at around 0.70 per cent, outperforms the ARIMAX models that all have a MAPE at about 0.78 per cent. The ARIMA(0,1,0) outperforms the other ARIMA structures with 0.002 percentage points, which really is not a lot, and there is probably no significant difference between the models. However, the method of this thesis does demand that only one model is used to make the actual forecasts. Since the ARIMA(0,1,0) technically outperforms the others, it is chosen to forecast the stock price of Carlsberg.

Genmab

The explanatory variables chosen for the ARIMAX models for Genmab are the lag of the stock price of Lundbeck, the exchange rates between Danish kroner and US dollars, and the American SP500 stock index.

The ARIMAX models outperform the ARIMA models based on AIC. The results of the out-of-sample testing point to the same conclusion. The forecasts from the ARIMAX models' MAPE are approximately 0.18 percentage points lower than those from the ARIMA forecasts. The error statistics across the ARIMAX models show ambiguous results. The RMSE statistic points to the conclusion that the ARIMAX(0,1,1) produces the most precise forecasts, while the MAPE statistic points to the ARIMAX(1,1,0). The results differ because RMSE puts a heavier weight on large errors, while MAPE is calculated using a simple average. The difference between the models is so small that one could argue that there is no real difference in the predictive strength of the models. The method for this thesis dictates that only one ARIMA(X) model is used to forecast each dependent variable, and since the out-of-sample testing technique points to an ambiguous answer, the model with the lowest AIC is chosen. ARIMAX(0,1,1) has the lowest AIC of the two and is chosen as the final model to forecast the stock price of Genmab.

Jyske Bank

The explanatory variable chosen for the ARIMAX models for Jyske Bank is the lag of the American SP500 stock index.

Looking at AIC, one would assume that the ARIMAX models slightly outperform the

ARIMA models. The out-of-sample testing points to the same conclusion. The evaluation statistics for the ARIMAX models are very similar; they all have a MAPE of 1.004 after rounding. However, the ARIMAX(1,1,0) has the lowest evaluation statistics, which is why this structure is chosen to make the actual forecasts of Jyske Bank's stock price.

Mærsk B

The explanatory variables chosen for the ARIMAX models for Mærsk B are the lag of the individual stock prices of FLSmidth and NovoNordisk, the Oil price and the American SP500 stock index.

Looking at AIC, one would assume that the ARIMAX models would outperform the ARIMA models. Despite this, the out-of-sample testing results show that the ARIMA models outperform the ARIMAX models with a MAPE around 0.01 percentage points lower. This means that the ARIMA(0,1,0) is determined to forecast Mærsk B's stock price.

SimCorp

The explanatory variables chosen for the ARIMAX models for SimCorp are the lag of the individual stock prices of DSV, Genmab, SydBank, WilliamDemant and the American SP500 stock index.

The ARIMAX models have a lower AIC estimate than the ARIMA models and are expected to perform better in out-of-sample testing. As is expected, the ARIMAX models have both lower RMSEs and MAPEs. Like in the case of Genmab, RMSE and MAPE show different results as to why AIC will be the final selection criteria. The models in question are ARIMAX(1,1,0) and ARIMAX(0,1,1), where the latter has a lower AIC, which means that the ARIMAX(0,1,1) is chosen to forecast the stock price of SimCorp.

Vestas

The explanatory variables chosen for the ARIMAX models for Vestas are the lag of the individual stock prices of Genmab, NovoNordisk, the exchange rates between Danish kroner and US dollars, the exchange rates between the Danish and Norwegian kroner, the Danish C20 stock index and the American SP500 stock index.

Even though the Vestas data series is non-stationary, the ARIMAX models follow a (p,0,q) structure, like in the case of Carlsberg. The models unit roots have been investigated and show that the models are stationary.

Looking at AIC, one would suspect that the ARIMA models' forecasts outperform the ARIMAX models' forecasts. The out-of-sample tests show similar results, where the ARIMA models' forecasts MAPEs is around 0.2 percentage points lower than the ARIMAX models' forecasts. The RMSE and MAPE statistics show that the ARIMA(3,1,0) + drift is the better forecasting model, and it is chosen as the final model.

Table 9: AIC for ARIMA(X) models

ARIMA	**Carlsberg**	**Genmab**	**Jyske Bank**	**Mærsk B**	**SimCorp**	**Vestas**
(0,1,0)	14031.57	16226.5	11511.93	26095.06	11358.79	-
(0,1,0) + d	14032.63	16226.27	11513.34	26096.92	11358.90	-
(1,1,0) + d	14033.85	16229.24	11515.34	26099.81	11361.86	-
(0,1,1) + d	14032.87	16228.28	11514.39	26098.9	11360.88	-
(2,1,1)	-	-	-	-	-	7231.059
(3,1,0)	-	-	-	-	-	7229.316
(3,1,0) + d	-	-	-	-	-	7231.192
(3,1,1)	-	-	-	-	-	7231.282

(a) ARIMA

ARIMA	**Carlsberg**	**Genmab**	**Jyske Bank**	**Mærsk B**	**SimCorp**	**Vestas**
(0,0,26)	14261.72	-	-	-	-	-
(0,0,27)	14249.59	-	-	-	-	-
(0,0,28)	14231.01	-	-	-	-	-
(0,0,29)	14231.36	-	-	-	-	-
(0,1,0)	-	16218.96	11461.32	-	11324.56	-
(1,1,0)	-	16221.61	11461.81	26038.83	11326.19	-
(0,1,1)	-	16220.68	11460.88	26038.28	11325.11	-
(2,1,2)	-	16220.79	-	-	-	-
(1,1,1)	-	-	11463.82	26040.64	11324.51	-
(0,1,2)	-	-	-	26039.87	-	-
(0,0,33)	-	-	-	-	-	7707.791
(0,0,34)	-	-	-	-	-	7696.4
(0,0,35)	-	-	-	-	-	7685.676
(0,0,36)	-	-	-	-	-	7669.293

(b) ARIMAX

Own creation based on calculations in R. Values are rounded for convenience

Table 10: RMSE and MAPE for ARIMA and ARIMAX

	Carlsberg		**Genmab**		**Jyske Bank**		**Mærsk B**		**SimCorp**		**Vestas**	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
(0,1,0)	**7.018**	**0.704**	29.301	1.768	4.717	1.009	**175.724**	**1.729**	10.100	1.418	-	-
(0,1,0) + d	7.025	0.706	29.303	1.770	4.726	1.012	175.806	1.731	10.096	1.417	-	-
(1,1,0) + d	7.032	0.706	29.303	1.770	4.739	1.013	175.886	1.732	10.095	1.418	-	-
(0,1,1) + d	7.033	0.706	29.303	1.770	4.739	1.013	175.886	1.7318	10.095	1.417	-	-
(2,1,1)	-	-	-	-	-	-	- -	-	-	1.625	1.451	
(3,1,0)	-	-	-	-	-	-	-	-	-	-	1.623	1.446
(3,1,0) + d	-	-	-	-	-	-	-	-	-	-	**1.622**	**1.446**
(3,1,1)	-	-	-	-	-	-	-	-	-	-	1.622	1.447

(a) RMSE and MAPE for ARIMA

	Carlsberg		**Genmab**		**Jyske Bank**		**Mærsk B**		**SimCorp**		**Vestas**	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
(0,0,26)	7.495	0.779	-	-	-	-	-	-	-	-	-	-
(0,0,27)	7.426	0.772	-	-	-	-	-	-	-	-	-	-
(0,0,28)	7.467	0.776	-	-	-	-	-	-	-	-	-	-
(0,0,29)	7.465	0.772	-	-	-	-	-	-	-	-	-	-
(0,1,0)	-	-	29.080	1.740	4.581	1.004	-	-	10.004	1.402	-	-
(1,1,0)	-	-	29.080	**1.737**	**4.558**	**1.004**	177.617	1.745	**10.008**	1.397	-	-
(0,1,1)	-	-	**29.079**	1.738	4.558	1.004	177.456	1.744	10.009	**1.397**	-	-
(2,1,2)	-	-	29.306	1.755	-	-	-	-	-	-	-	-
(1,1,1)	-	-	-	-	4.558	1.004	177.640	1.745	10.018	1.400	-	-
(0,1,2)	-	-	-	-	-	-	177.640	1.740	-	-	-	-
(0,0,33)	-	-	-	-	-	-	-	-	-	-	1.828	1.681
(0,0,34)	-	-	-	-	-	-	-	-	-	-	1.811	1.658
(0,0,35)	-	-	-	-	-	-	-	-	-	-	1.835	1.681
(0,0,36)	-	-	-	-	-	-	-	-	-	-	1.835	1.678

(b) RMSE and MAPE for ARIMAX

Own creation based on calculations in R. Values are rounded for convenience

10.1 Summary

In total, three ARIMA and three ARIMAX models were chosen to forecast the six dependent variables. Two of the ARIMA models chosen resembles a random walk with their (0,1,0) structure; these models were chosen for the Carlsberg and the Mærsk B stock. The last ARIMA is chosen to forecast the Vestas stock and has a (3,1,0)+drift structure. In the selection process of both Genmab and SimCorp, the RMSE and MAPE pointed to different conclusions. This meant that the models with the lowest AIC were chosen, which were ARIMAX(0,1,1) for both the Genmab and SimCorp stock. The four ARIMAX models for the Jyske Bank stock had very similar testing forecast error statistics with a MAPE of 1.004 after rounding. The ARIMAX(1,1,0) ultimately had the lowest MAPE as to why it was chosen to forecast the Jyske Bank stock.

11 Selection of ANN models

The best ANN model for forecasting the dependent variables is selected in this chapter. The process by which ANN models are selected is described in section 7.5. The models are first trained using the training data, and their results are then validated using the testing data. The training part of the data is also where the tuning of ANN models takes place. During training, 150 models of each type are created and trained, and the best model of each type is then carried forward to the testing part. Here the results of the models are tested and validated, and the model with the lowest RMSE and MAPE is selected for forecasting a given dependent variable. For Carlsberg, for example, there would be 150 of each of the model types, i.e. FNN, RNN, LSTM, GRU, bidirectional LSTM and bidirectional GRU. All these models are then trained, and the best of each type is tested and compared to the other types to choose the final model. This is repeated for all dependent variable, both with and without extra explanatory variables, which leads to a total of $10,800$ trained models, of which six is selected.

The explanatory variables used to forecast the dependent variables are the same as in the ARIMAX forecast. Five timesteps are used in all recurrent models, which would be equivalent to including five lags of all explanatory variables in the ARIMAX models. The RMSE and MAPE for all dependent variables after out-of-sample testing are presented in table 11.

The specific model for each dependent variable is explained below. The structure of the models that were not selected for each dependent variable can be found in *Appendix 4 - Structure for ANN testing models*

Carlsberg:

The best ANN architecture for forecasting Carlsberg without explanatory variables is an FNN with an RMSE of 6.98, while the best architecture to predict the stock with explanatory variables is a GRU with an RMSE of 11.44. The results produced by models without extra explanatory variables are better across the board; no model architecture achieves better results when extra explanatory variables are included. Carlsberg is notable for having the lowest MAPE for all model architectures out of all the dependent variables.

The FNN without extra explanatory variables is chosen to predict the forecasting period since it is the best model for forecasting Carlsberg by far, with a MAPE of 0.7 per cent. The final model structure can be seen in figure 12.

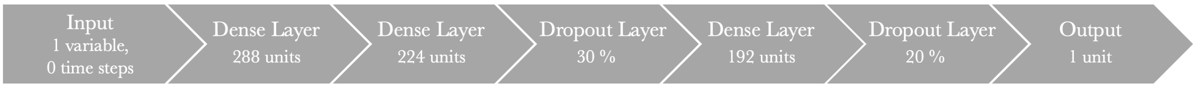


Figure 12: Final structure for Carlsberg

The model has three standard dense layers[16], two dropout layers, and a learning rate of 0.01. Since the model is an FNN without extra explanatory variables, the input only consists of one variable and no timesteps.

Genmab

The FNN is the best architecture for forecasting the testing period with an RMSE of 29.57 without extra explanatory variables and 31.38 with extra explanatory variables. The inclusion of extra explanatory variables once again worsens the models' performances across all model architectures. Additionally, the FNN without explanatory variables achieves a MAPE of 1.79 per cent and a MAPE of 1.90 per cent if explanatory variables are included. For these reasons, the FNN architecture without explanatory variables is chosen to forecast Genmab's stock price during the forecasting period. The final model structure can be seen in figure 13.

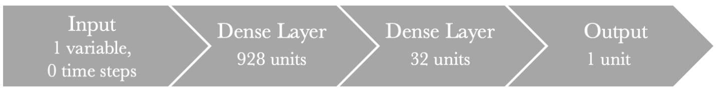


Figure 13: Final structure for Genmab

The model consists of two standard dense layers, no dropout, and a learning rate of 0.001. As with Carlsberg, the model's input only consists of one variable and no timesteps. Although the model structure consists of fewer layers than Carlsberg, it is still a very complex model due to the high amount of units (928) in the first dense layer.

Jyske Bank

The testing data shows that the best architecture to forecast Jyske Bank is the FNN without additional explanatory variables, with an RMSE of 4.83. If additional explanatory

[16] Dense layers are what feed-forward layers are referred to in the Keras framework

variables are included, the GRU becomes the best architecture, with an RMSE of 6.30. Although the FNN without additional explanatory variables is once again the best model, the evaluation statistics show that the inclusion of explanatory variables can benefit some model architectures. The LSTM, GRU, and bi-directional GRU all improve their RMSE and MAPE when additional explanatory variables are included. The average RMSE for Jyske bank with and without explanatory variables is also much closer than Carlsberg and Genmab. Despite this, the FNN without explanatory variables is still the best option and is chosen as the final model. The final model structure can be seen in figure 14.

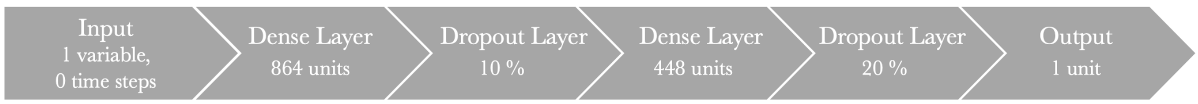


Figure 14: Final structure for Jyske Bank

The model consists of two standard dense layers, two dropout layers, and a learning rate of 0.01. The model's input only consists of one variable and no timesteps.

Mærsk B

The best model architecture for Mærsk B is the FNN without extra explanatory variables with an RMSE of 177.25 and a MAPE of 1.75. The LSTM architecture improves slightly when adding extra explanatory variables, but on average, the models without explanatory variables perform better. The FNN without extra explanatory variables is chosen as the final model, and its structure can be seen in figure 15.

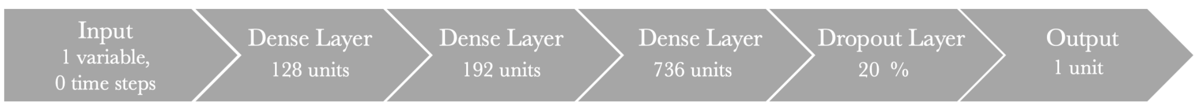


Figure 15: Final structure for Mærsk B

The model consists of three dense layers, one dropout layer, and a learning rate of 0.001. The model's input only consists of one variable and no timesteps.

Simcorp

Forecasting Simcorp during the testing period shows that the FNN architecture is once again the best. It is notable, though, that only the FNN with extra explanatory variables is able to achieve results comparable to the models without extra explanatory variables.

All other model architectures with extra explanatory variables achieve much higher RMSEs and MAPEs than their counterparts without extra explanatory variables. Even the worst model architecture without extra explanatory variables is much better than the best model with extra explanatory variables, except for the FNN. The FNN without extra explanatory variables achieves an RMSE of 10.13 and a MAPE of 1.43 and is chosen as the final model. Its structure can be seen in figure 16.

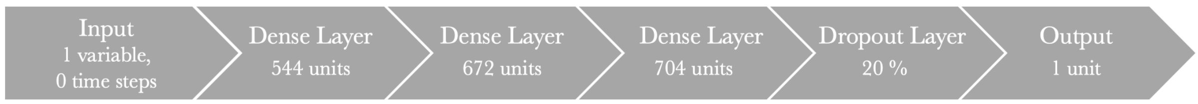


Figure 16: Final structure for SimCorp

The model consists of three dense layers, one dropout layer and a learning rate of 0.0001. The model's input only consists of one variable and no timesteps. Although the amount of layers is the same as with Mærsk B, the amount of units is far higher for SimCorp, and the learning rate is much lower. This suggests that the two might share the same general structure but that Simcorp's network is much more complex and learns more slowly over time.

Vestas

As with the other dependent variables, the FNN without extra explanatory variables is the best architecture for forecasting Vestas. It achieves an RMSE of 1.62 and a MAPE of 1.44. Unlike the results for SimCorp, the results of model architectures with and without extra explanatory variables are quite close. Unlike Jyske Bank, though, none of the model architectures are improved when adding explanatory variables. Once again, the FNN architecture without extra explanatory variables is chosen as the final model, and its structure can be seen in figure 17.

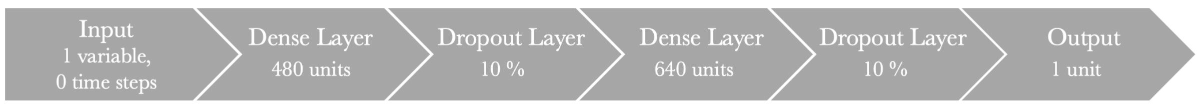


Figure 17: Final structure for Vestas

The model consists of two standard dense layers, two dropout layers, and a learning rate of 0.001. The model's input only consists of one variable and no timesteps.

Table 11: RMSE and MAPE for all ANN models

	Carlsberg		**Genmab**		**Jyske Bank**		**Mærsk B**		**Vestas**		**SimCorp**	
	Excl	Incl.	Excl	Incl.	Excl	Incl.	Excl	Incl.	Excl	Incl.	Excl	Incl.
FNN	**6.98**	11.66	**29.57**	**31.38**	**4.83**	6.79	**177.25**	**249.48**	**1.62**	**1.69**	**10.13**	**10.95**
RNN	10.06	13.10	41.82	45.23	6.35	6.38	261.32	283.58	2.38	2.67	14.83	48.06
LSTM	11.67	12.71	46.00	47.06	6.78	6.65	284.51	280.91	2.55	2.57	17.12	45.06
bi-LSTM	11.44	13.24	43.53	50.14	7.11	7.58	310.75	314.86	2.87	2.94	18.49	29.53
GRU	10.44	**11.44**	40.92	42.09	6.50	**6.30**	263.16	281.06	2.40	2.62	15.72	28.42
bi-GRU	12.18	12.39	40.92	47.93	7.53	7.26	301.58	303.78	2.75	3.07	18.65	28.55
Mean	**10.46**	12.42	**41.71**	43.97	**6.52**	6.83	**266.43**	285.61	**2.43**	2.59	**15.82**	31.76

(a) RMSE for training period

		FNN	**RNN**	**LSTM**	**bi-LSTM**	**GRU**	**bi-GRU**
Carlsberg	Excl.	**0.7**	**1.03**	**1.20**	**1.23**	**1.07**	**1.27**
	Incl.	1.25	1.41	1.41	1.44	1.25	1.33
Genmab	Excl.	1.79	2.54	2.20	2.43	2.02	2.34
	Incl.	1.90	2.82	3.00	3.21	2.68	3.06
Jyske Bank	Excl.	1.05	1.49	1.61	1.68	1.54	1.75
	Incl.	1.61	1.56	1.56	1.78	1.49	1.70
Mærsk B	Excl.	1.75	2.47	2.76	3.09	2.54	3.00
	Incl.	2.60	2.76	2.73	3.15	2.71	3.03
Vestas	Excl.	1.44	2.00	2.20	2.43	2.02	2.34
	Incl.	1.47	2.21	2.22	2.47	2.17	2.49
SimCorp	Excl.	1.43	2.15	2.53	2.72	2.31	2.71
	Incl.	1.63	8.36	8.56	5.48	4.53	4.83
Mean	Excl.	**1.36**	1.95	2.21	2.31	1.99	2.36
	Incl.	1.74	3.18	3.24	2.92	2.47	2.74

(b) MAPE for training period

Own creation based on calculations in R. Values are rounded for convenience

11.1 Summary

The best performing models for all variables were FNNs. Although, some other model types were close to performing as well as the FNNs, such as the GRU for Jyske Bank and Carlsberg. The inclusion of variables improved only a few models. The final model selected for all dependent variables was an FNN without extra explanatory variables.

The best ARIMA(X) and ANN models have been found for each dependent variable. The final models are used to predict the forecasting period, and the results are presented and analysed in the following chapter.

12 Results

This chapter presents and evaluates the results of the forecasts made by the best ARIMA(X) and the best ANN models selected in the previous chapters. As explained in chapter 7, one ARIMA(X) and one ANN are used to forecast each dependent variable. The specific models used to forecast are explained in chapter 10 and 11. Each model is used to predict the forecasting period using one-step-ahead forecasting. The resulting evaluation statistics and the Diebold-Mariano (DM) p-values of the forecasts can be seen in table 12. The average daily percentage changes for the forecasting period, presented in chapter 9, are included in the table. These are included to compare the models to the actual volatility of the stock prices since the average daily percentage changes are directly comparable to the MAPE evaluation statistic. Plots of the predicted and the actual prices can be seen in figure 18. Since this thesis uses one-step-ahead forecasting, the errors become very small relative to the actual prices. This means that the differences in ARIMA(X) and ANN forecasts can be very hard to spot in the plots. Therefore, the results and discussion will be based on the evaluation statistics rather than the graphical representation.

Table 12: Results of forecasting

(a) Evaluation statistics

	RMSE			**MAPE**			**Average daily change (%)**
	ARIMA	ANN	Difference	ARIMA	ANN	Difference	
Carlsberg	**11.65**	11.75	0.91 %	**0.85**	0.86	0.01	0.86
Genmab	**20.70****	21.12	1.99 %	**1.26****	1.27	0.01	1.27
Jyske Bank	**3.81****	4.03	5.55 %	**1.21****	1.35	0.14	1.17
Mærsk B	160.75	**160.57**	-0.12 %	1.48	**1.47**	-0.004	1.48
SimCorp	**10.69****	11.16	4.25 %	**1.27****	1.33	0.06	1.28
Vestas	**2.05**	2.06	0.39 %	**1.30**	1.31	0.01	1.30
Average			2.16 %			0.04	

*ARIMAX models

Own creation based on calculations in R. Values are rounded for convenience

(b) Diebold-Mariano test

	Carlsberg	**Genmab**	**Jyske Bank**	**Mærsk B**	**SimCorp**	**Vestas**
P-value	0.01	0.02	0.03	0.45	0.003	0.28

Own creation based on calculations in R. Values are rounded for convenience

The error statistics in table 12 show that, aside from Mærsk B, the ARIMA(X) models outperform the ANN models when forecasting all the dependent variables. Because the statistics are based on one-step-ahead daily forecasting, the differences in MAPE and RMSE generally become smaller than if it was based on lower frequency data, e.g. monthly, with longer forecasting horizons. These small average differences in RMSE and MAPE might not seem like a lot, but considering that the difference is every day for a year, these small differences could start to add up. The results will be presented individually for each of the dependent variables.

Carlsberg

Although the ANN for Carlsberg is outperformed by the ARIMA(X), the ANN does get a MAPE equal to the daily percentage change. This is despite the difference in trend between the forecasting and testing period. The ANN model has a 0.91 per cent higher RMSE and a 0.01 percentage points higher MAPE. The DM test was set to test whether the ARIMA(X) forecasts were statistically more accurate than the ANN forecasts. The null hypothesis states that the accuracy of the ARIMA(X) forecasts is not higher than the forecasts of the ANN. The test has a p-value of 0.01, which means that the null hypothesis is rejected. Thus it is concluded that the ARIMA(X) forecasts are more accurate than the ANN forecasts, at a significant level.

Genmab

The ANN is outperformed by the ARIMA(X) when forecasting the Genmab stock. The ANN forecasts do get a MAPE equal to the average daily percentage change, which is a sign that the forecasts are somewhat precise. The fact that the testing period was markedly different from the training and forecasting period does not seem to have had a noticeable effect. The difference in RMSE between the ANN and the ARIMA(X) is 1.99 per cent, and the MAPE difference is, like in the case of Carlsberg, 0.01 percentage points. A DM test is performed on the ANN and ARIMA(X) forecast errors. The test has a p-value of 0.02, which means that the null hypothesis is rejected. Thus, the ARIMA(X) forecasts are more accurate than the ANN forecasts, at a significant level.

Jyske Bank

The biggest difference in model performance between ANN and ARIMA(X) is when forecasting Jyske Bank's stock price. The ARIMA(X) outperforms the ANN model with 5.55 per cent looking at RMSE and 0.14 percentage points looking at MAPE. Both the

ANN and the ARIMA(X) model struggle to forecast Jyske Bank's stock price. This can be seen in the difference between their MAPEs and the average daily percentage change. The ANN had a 0.18 percentage points higher MAPE, and the ARIMA(X) had a 0.04 percentage points higher MAPE. The reason the models struggle when forecasting Jyske Bank is most likely because the forecasting period trends downwards, while its training period trends upwards. The DM test shows that the ARIMA(X) forecasts are more accurate than those of the ANN at a significant level, with a p-value of 0.03.

Mærsk B

The only dependent variable where the ANN forecasts outperform those of the ARIMA(X) is Mærsk B. It outperforms with a very slim margin though, and out of the six dependent variables, this is where the difference is the smallest. The ANN outperforms with a 0.12 per cent lower RMSE and a 0.004 percentage points lower MAPE. In addition to being the only stock where the ANN outperforms, Mærsk B is also the only dependent variable where the ANN has a MAPE that is lower than the average daily percentage change, with a difference of 0.01 percentage points. The fact that the ANN outperformed the ARIMA(X) could be explained by Mærsk B having no clear trend and different training and testing periods. However, the outperformance is not significant. The DM test has a p-value of 0.45 which means that the ANN forecasts are not more accurate than those of the ARIMA(X) at a significant level.

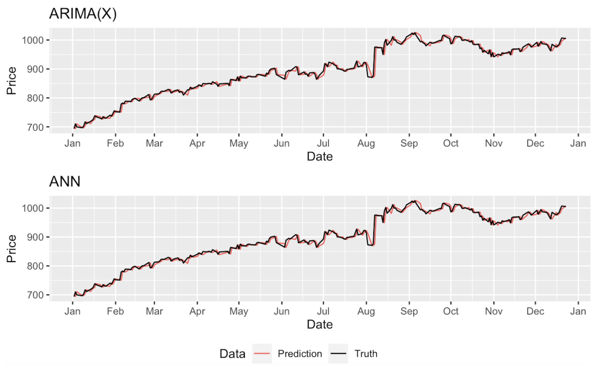
SimCorp

The ANN model for SimCorp struggled to forecast the testing period, so it comes as no surprise that it also struggles to predict the forecasting period. The ANN model is outperformed by 4.25 per cent looking at RMSE and 0.06 percentage points looking at MAPE, compared to the ARIMA(X) model. The model also fails to reach the average daily percentage change, with a MAPE that is 0.05 percentage points higher. The DM test shows that the ARIMA(X) forecasts are more accurate than those of the ANN at a significant level, with a p-value of 0.003.

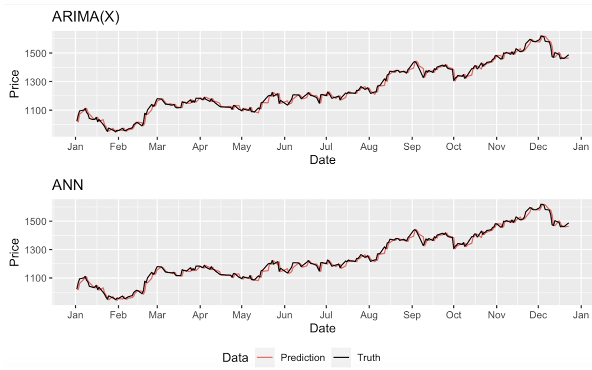
Vestas

Despite the irregularities in Vestas' stock price, the ANN was outperformed by the ARIMA(X) model. The ARIMA(X) forecasts had a 0.39 per cent lower RMSE and a 0.01 percentage points lower MAPE. The ANN ended up with a MAPE of 1.31, while the average daily percentage change was at 1.30. Surprisingly, the DM test shows that the ARIMA(X) forecasts are not more accurate than those of the ANN at a significant level, with a p-value of 0.28. This might seem peculiar since the models for Vestas had a bigger difference in RMSE and MAPE compared to the models for Carlsberg and Genmab. The answer to this lies in how the DM statistic is calculated. This is the case since the DM statistic is used to calculate the p-values. Recall from equation 14 that the DM statistic is calculated by the mean of the loss differences between the models divided by the standard deviation of that mean. In the case of Carlsberg, Genmab, and Vestas, the mean of the loss differences are approximately the same, but the standard deviation of the mean from Vestas is much higher than those from Carlsberg and Genmab. This leads to a lower DM statistic, which in turn leads to a higher p-value.

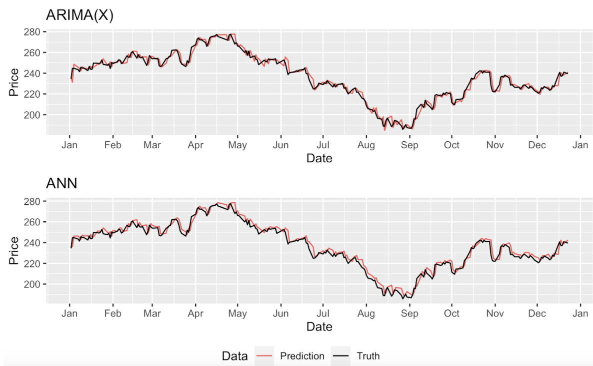
Figure 18: Forecasted stock prices for ARIMA(X) and ANN



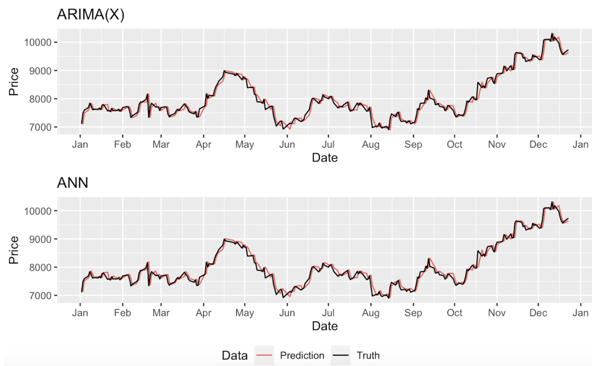
(a) Carlsberg



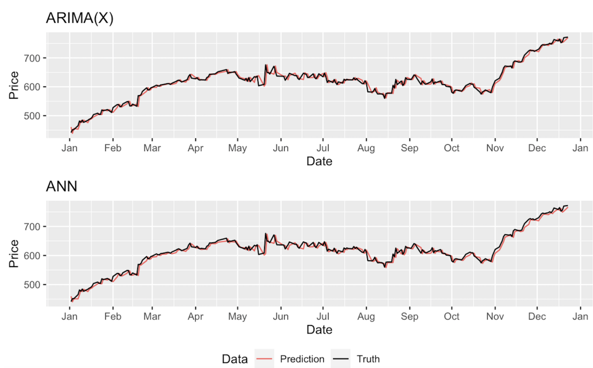
(b) Genmab



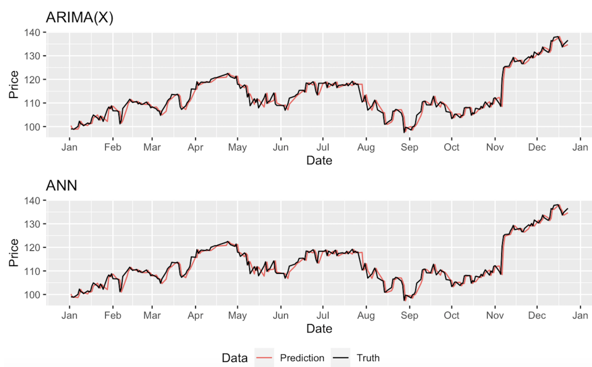
(c) Jyske Bank



(d) Mærsk B



(e) SimCorp



(f) Vestas

12.1 Summary

The results from the forecasts show that, aside from Mærsk B, the ANN models were outperformed by the ARIMA(X) models. In the case of Mærsk B, the ANN model outperformed the ARIMA(X) model, but only by a slim margin. The DM tests show that the differences in accuracy were significant for Carlsberg, Genmab, Jyske Bank, and SimCorp. The MAPEs from the ANN models were either lower or the same as the average daily percentage change when forecasting Carlsberg, Genmab, and Mærsk B. It was found that the ANN models struggled to forecast both Jyske Bank and SimCorp.

The final results for each dependent variable have been presented and analysed. In the following chapter, these results and their implications are discussed.

13 Discussion

The results presented in chapter 12 clearly show that ANNs are still not at a point where they can predict the Danish stock market as accurately as the ARIMA(X) model. For the ANN models to actually be useful, they would need to outperform the ARIMA(X) and get a lower MAPE than the average daily percentage change. This general underperformance of ANN models might lead one to conclude that nothing of interest has been gained from predicting the Danish stock market with ANN models in this thesis, but that is not the case. This chapter will explore and discuss some of the implications of the results of this thesis.

13.1 Does the Danish stock market behave as a random walk?

The fact that the ANN models are unable to outperform both the average daily percentage change and the ARIMA(X) models could be explained by the Danish stock market, in fact behaving as a random walk. This is further supported by the structure of all the best ANN models. In chapter 5 it was mentioned that a simple RNN should be superior at predicting time series data compared to an FNN. This is the case since RNNs, contrary to FNNs, examines the data as a sequence. Furthermore, the theory states that more complicated architectures of RNN such as LSTM and GRU, which uses a gated structure, should outperform the simple RNN architecture. Despite this, the out-of-sample testing showed that FNNs are the better predictor for all dependent variables, even though it does not factor in that the data is in a sequence. This suggests that a simpler model architecture is preferable. The reason might be that FNNs are better at imitating a random walk than RNNs, LSTMs, GRUs, bi-LSTMs, and bi-GRUs. It could be the case that RNNs, LSTMs, GRUs, bi-LSTMs, and bi-GRUs simply generate too much noise when trying to model connections that do not exist.

However, the ARIMA(X) selection suggest something different since the best models for forecasting the dependent variables was not exclusively random walks. This could signify that the Danish stock market has irregular price changes that can be predicted. The reason might also be the selection method, where the ARIMA(X) models selected to forecast only outperforms a random walk in the testing period by a slim margin, as explained in chapter 10. The largest departure from a random walk in the ARIMA models is Vestas with a (3,1,0) structure, but the difference between this structure and a random walk in the out-of-sample test is marginal. The same goes for when ARIMAX models are selected, where the ARIMAX models only outperform the random walk by a slim margin. Thus it can be assumed that the departures from the random walk structure for the ARIMA(X) models are more likely to be caused by the selection method used in this thesis, rather than a sign that the Danish stock market does not behave like a random walk. If the Danish stock market does, in fact, behave like a random walk, adding more information such as; the stock price of other companies, indices, interest rates, currency conversion rates, or prices of commodities, to the forecasting models should either worsen or not affect the forecasts of the models. This effect can be seen in the ANN models, where adding extra explanatory variables worsens the forecasts of almost all the models. The implications of this will be discussed in the next section.

13.2 Effect of adding explanatory variables

One big difference between the ARIMA(X) and ANN models is that some of the ARIMA(X) models benefit from the inclusion of extra explanatory variables. This is interesting for two reasons. Firstly, that the efficient market hypothesis suggests that extra explanatory variables are not useful when forecasting stock markets. Secondly, because the extra explanatory variables actually worsen the ANN models' performance. The fact that some ARIMA(X) models benefit from including extra explanatory variables could be explained by the same reasons that the ARIMA(X) for some dependent variables diverge from a random walk. The results of the out-of-sample testing are very close, and any ARIMAX models chosen over an ARIMA model might simply be due to the selection method used. The reasons why the ANN models performance worsen when the extra explanatory variables are included are more unclear. Theoretically, the ANN models should be able to adapt to unneeded information and learn to ignore it during training, but this does not seem to happen. The models' inability to properly adapt might be due to the extra explanatory variables simply adding too much noise to the models for them to filter out properly. Another reason could be that the chosen time period might not provide enough data for the model to learn to ignore the extra explanatory variables. A final reason could be the frequency of data used in this thesis. Dingli and Fournier (2017) suggest that the amount of extra explanatory variables that benefit a machine learning model is dependent on the frequency of the data, with higher frequencies like daily data needing fewer extra explanatory variables. Although their study focuses on traditional supervised machine learning, their findings could also be applied to ANN models. Further comparisons to existing literature are made in the next section.

13.3 Comparison to existing literature

The results presented in this thesis diverges heavily from the litterateur presented in section 2.3. All other comparative studies reviewed showed that an ANN is either always better or at least better in some cases compared to an ARIMA. The fact that the ARIMA models often outperform the ANN models in this thesis shows that the ANN methodology, when applied to the Danish stock market, might not be as powerful as other literature suggests. There could be many reasons why this thesis comes to a different conclusion, but without a clear look at the coding and complete methodology of these papers, it is hard to state clear reasons. Though, some hypothesis can be generated based on the data used by the authors. Du (2018) use data from much larger economies than Denmark. Siami-Namini et al. (2018) forecast weekly and monthly data instead of the daily data used in this thesis and Isenahd and Olubusoye (2014) forecasts a much younger and maybe more volatile market than the Danish stock market. As such, the differences in results might be explained by the difference in the data used by existing literature and this thesis.

13.4 Possible improvements to methodology

These differing results lead to a discussion of the methodology used in this thesis. As mentioned earlier, the selection method for ARIMA(X) models presents some problems. These are mainly that the ARIMA(X) models are so similar that it becomes almost impossible to select the definitively best model. Since this thesis' primary focus is not on the performance of ARIMA(X) models, though, this is more of an inconvenience than a problem. A general problem when training ANN models is that the models can be very hard to reproduce precisely. The number of changeable parameters also present a challenge to defining whether the best possible model has been achieved. With the amount of models trained and tested though, it is likely that the best possible model within the given set of parameters has been achieved. As for the number of changeable parameters, this was solved somewhat by using automatic tuning, which vastly improves the ability to test many different combinations of parameters. Of course, some limitations exist, both in the form of time constraints and computational power, limiting the number of possible parameter combinations that can be tested. Unfortunately, these limitations are hard to solve and present more of a trade-off between complexity and feasibility than any true solutions.

The selection of extra explanatory variables could also lead to problems. These problems could stem from both the extra explanatory variables included and those that were not. The extra explanatory variables that were included were selected based on the Granger causality test. Although this measurement is solid, the given p-value for selecting variables might be too high. This would mean that variables that have a too low predictive power were included in the models. The extra explanatory variables that were not included are a much more complex problem to solve. There will almost always be extra explanatory variables that could have been included but were not, so gathering variables that might be of interest is sometimes more of an art than a science. As with the limitations in ANN models, this problem also stems mainly from the trade-off between complexity and feasibility.

13.5 Real life application of methods

A discussion of the real-life application of the methods used in this thesis is also interesting since the work needed to implement the two methods are different. The results of this thesis show that ANNs cannot outperform ARIMA(X) models, but even if they had been able to, the real-life applicability would still be questionable. The time and resources needed to implement the methodology used for ANNs in this thesis are immense compared to the time and resources used to implement the ARIMA(X) models. The coding used for the ARIMA(X) models in this thesis took between 10 and 15 minutes to run, whereas the coding used for the ANN models took up to 2 days to run, even with GPU acceleration which speeds up the training and tuning process. This means that ANNs would have to not only be comparable to ARIMA(X) models but outperform them with a significant margin for them to be an effective tool for forecasting the Danish stock market. This could change in the future, however. As mentioned in the introduction, the development of both computational power and machine learning in general is happening at a rapid pace. As such, it is not unlikely that ANNs will be able to be both more efficient and better at forecasting the Danish stock market than ARIMA(X) models in the future.

13.6 Summary

Although the results produced in this thesis shows that ANN models cannot predict the Danish stock market better than an ARIMA(X) model, this chapter has pointed to some other interesting conclusions. The resulting structures of the ANN models suggest that the Danish stock market behaves like a random walk. This is also supported by the fact that adding extra explanatory variables has no or a negative impact on the models. Compared to the existing literature, these results are very different, with the ANNs often beating ARIMA models in the literature. A possible reason for this might be the different data used in this thesis. Although the selection method used in this thesis presents some difficulties, there is no true solutions, only trade-offs. Lastly, the real-world application of the methods used in this thesis was discussed. It is clear that the time and computational power needed to train the required amount of ANN models presents a challenge to its implementation. This means that even if the results of the ANN models were better, they would still be impractical to use.

14 Conclusion

The work in this thesis has been done in order to answer the problem statement:

How well do artificial neural network models predict future prices of the Danish stock market?

The problem statement was answered by forecasting a selection of Danish stocks with ANN models and comparing them to ARIMA(X) models used as baselines. The year 2019 was used as the forecasting period, where the daily stock prices of *Carlsberg*, *Genmab*, *Jyske Bank*, *A.P. Møller - Mærsk B*, *SimCorp*, and *Vestas Wind Systems* were forecasted using one-step-ahead forecasting. The years from the start of 2010 to the end of 2018 was used to train and select both the best ANN and ARIMA(X) models using an out-of-sample testing strategy.

The best ANN model type for all the dependent variables ended up being a feed-forward neural network, compared with a simple RNN, LSTM, GRU, bi-LSTM, and bi-GRU. None of the models used extra explanatory variables since they generally worsened model performance. Although the ANN models were of the same type, their structures differed across the dependent variables. It was evident in the selection of ARIMA(X) models that there were no big differences between the models tested. The models chosen to forecast ended up being: Carlsberg - ARIMA(0,1,0), Genmab - ARIMAX(0,1,1), Jyske Bank - ARIMAX(1,1,0), Mærsk B - ARIMA(0,1,0), SimCorp - ARIMAX(0,1,1) and Vestas - ARIMA(3,1,0) with drift. This meant that three ARIMA models and three ARIMAX were used to forecast.

The results of the forecasting period showed that the ANN models perform worse or similar compared to the ARIMA(X) models. The ANN and ARIMA(X) error statistics were close in some cases. After further inspection with a Diebold-Mariano test, it was determined that no model had a better accuracy than the other when forecasting Mærsk B and Vestas. However, the Diebold-Mariano test showed that the ARIMA(X) models were statistically more accurate when forecasting the remaining four stocks.

The discussion revolved around why this thesis got these results. One of the possible reasons pointed to were that the Danish stock market, in fact, behaves like a random

walk. This notion is further supported by the fact that the best ANN models proved to be feed-forward neural networks, especially since the addition of extra explanatory variables worsened most of the ANN models.
The results were also compared to the existing literature. Contrary to the results in this thesis, the existing literature showed that the ANN models heavily outperforms the ARIMA models. There can be many reasons as to why the results in this thesis differ, such as difference in data length and frequency, and the type of country investigated.

It was concluded that ANNs still need further development before using them to forecast stock market prices becomes feasible. Especially considering the time and computational power needed to train, tune, and test ANNs, compared to ARIMA(X) models.

References

Abe, M. and Nakayama, H. (2018). Deep learning for forecasting stock returns in the cross-section. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer.

Afeef, M., Ihsan, A., and Zada, H. (2018). Forecasting stock prices through univariate arima modeling. *NUML International Journal of Business & Management*.

Agrawal, A., Gans, J., and Goldfarb, A. (2020). How to with machine learning. *Harvard Business review*.

Balaji, A. J., Ram, D. H., and Nair, B. B. (2018). Applicability of deep learning models for stock price forecasting an empirical study on bankex data. *Procedia computer science*.

Beyaz, E., Tekiner, F., Zeng, X.-j., and Keane, J. (2018). Comparing technical and fundamental indicators in stock price forecasting. In *2018 IEEE 20th International Conference on High Performance Computing and Communications*.

Bosworth, B., Hymans, S., and Modigliani, F. (1975). The stock market and the economy. *Brookings Papers on Economic Activity*.

Brownlee, J. (2019). A gentle introduction to dropout for regularizing deep neural networks. https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/. Accessed 24. May 2021.

Brownlee, J. (2020a). Machine learning is popular right now. https://machinelearningmastery.com/machine-learning-is-popular/. Accessed 1. June 2021.

Brownlee, J. (2020b). Understand the impact of learning rate on neural network performance. https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/. Accessed 24. May 2021.

Date, S. (2020). Regression with arima errors. https://towardsdatascience.com/regression-with-arima-errors-3fc06f383d73. Accessed 24. May 2021.

Dingli, A. and Fournier, K. S. (2017). Financial time series forecasting-a machine learning approach. *Machine Learning and Applications: An International Journal*.

Du, Y. (2018). Application and analysis of forecasting stock price index based on combination of arima model and bp neural network. In *2018 Chinese Control And Decision Conference (CCDC)*. IEEE.

Enders, W. (2015). *Applied econometric time series*. Wiley, 4th edition.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*.

Foote, K. D. (2019). A brief history of machine learning. https://www.dataversity.net/a-brief-history-of-machine-learning/. Accessed 1. June 2021.

Grossman, S. J. and Stiglitz, J. E. (1980). On the impossibility of informationally efficient markets. *The American economic review*.

Hurd, M. D. and Rohwedder, S. (2010). Effect of the financial crisis and great recession on american household. *NBER Working Paper Series*.

Isenahd, G. M. and Olubusoye, O. E. (2014). Forecasting nigerian stock market returns using arima and artificial neural network models. *CBN Journal of Applied Statistics*.

KerastuneR (2021). Hyperparameters. https://keras-team.github.io/keras-tuner/documentation/hyperparameters/. Accessed 1. June 2021.

Li, S. and Li, R. (2017). Comparison of forecasting energy consumption in shandong, china using the arima model, gm model, and arima-gm model. *Sustainability*.

Lütkepohl, H. and Xu, F. (2010). The role of the log transformation in forecasting economic variables. *Empirical Economics*.

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *Journal of economic perspectives*.

Malkiel, B. G. (2016). *A random walk down wall street: The time-tested strategy for successful investing*. W. W. Norton & Company, 11th edition.

McCarthy, J. (2007). What is artificial intelligence? http://jmc.stanford.edu/articles/whatisai/whatisai.pdf. Accessed 29. May 2021.

Mehtab, S., Sen, J., and Dutta, A. (2020). Stock price prediction using machine learning and lstm-based deep learning models. *the Second Symposium on Machine Learning and Metaheuristic Algorithms and Applications (SOMMA'20)*.

Mondal, P., Shit, L., and Goswami, S. (2014). Study of effectiveness of time series modeling (arima) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*.

Naik, N. and Mohan, B. R. (2019). Stock price movements classification using machine and deep learning techniques-the case study of indian stock market. In *International Conference on Engineering Applications of Neural Networks*. Springer.

Navares, R., Díaz, J., Linares, C., and Aznarte, J. L. (2018). Comparing arima and computational intelligence methods to forecast daily hospital admissions due to circulatory and respiratory causes in madrid. *Stochastic Environmental Research and Risk Assessment*.

Pi, M. (2018a). Illustrated guide to lstm's and gru's: A step by step explanation. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21. Accessed 16. March 2021.

Pi, M. (2018b). Illustrated guide to recurrent neural networks. https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9. Accessed 16. March 2021.

Sanderson, G. (2017a). But what is a neural network? | deep learning, chapter 1. https://youtu.be/aircAruvnKk. Accessed 24. May 2021.

Sanderson, G. (2017b). Gradient descent, how neural networks learn | deep learning, chapter 2. https://youtu.be/IHZwWFHWa-w. Accessed 24. May 2021.

Sanderson, G. (2017c). What is backpropagation really doing? | deep learning, chapter 3. https://youtu.be/Ilg3gGewQ5U. Accessed 24. May 2021.

Schuchmann, S. (2019). History of the first ai winter. https://towardsdatascience.com/history-of-the-first-ai-winter-6f8c2186f80b. Accessed 1. June 2021.

Shen, S., Jiang, H., and Zhang, T. (2012). Stock market forecasting using machine learning algorithms. *Department of Electrical Engineering, Stanford University*.

Siami-Namini, S., Tavakoli, N., and Namin, A. S. (2018). A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*.

Statista (2021). Distribution of countries with largest stock markets worldwide as of january 2021, by share of total world equity market value. https://www.statista.com/statistics/710680/global-stock-markets-by-country/. Accessed 1. June 2021.

Thomakosa, D. D. and GuerardJr, J. B. (2004). Naıve, arima, nonparametric, transfer function and var models: A comparison of forecasting performance. *International Journal of Forecasting*.

Udenrigsministeriet (2018). Modtagerlande for dansk eksport. https://virksomhedsguiden.dk/erhvervsfremme/content/ydelser/modtagerlande-for-dansk-eksport/97d21f1c-149e-440b-a809-0dc02d94ff46/. Accessed 7. May 2021.

Venkatachalam, M. (2019). Recurrent neural networks. https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce. Accessed 16. March 2021.

Virtanen, I. and Yli-Olli, P. (1987). Forecasting stock market prices in a thin security market. *Omega*.

W. Lo, A. and MacKinlay, A. C. (1999). *A non-random walk down wall street*. Princeton University Press, 5th edition.

Yuan, C., Liu, S., and Fang, Z. (2016). Comparison of china's primary energy consumption forecasting by using arima (the autoregressive integrated moving average) model and gm(1,1) model. *Energy*.

15 Appendix

15.1 Appendix 1 - Granger Causality test

	Carlsberg	**Genmab**	**Jyske Bank**	**Mærsk B**	**SimCorp**	**Vestas**
Ambu	0.120	0.875	0.380	0.180	0.037	0.574
Carlsberg	-	0.945	0.497	0.720	0.135	0.479
Coloplast	0.405	0.731	´0.045	0.711	0.065	0.029
Danske Bank	**0.003**	0.455	0.196	0.255	0.040	0.415
DSV	**0.005**	0.494	0.079	0.433	**0.000**	0.072
FLSmidth	0.529	0.071	0.282	0.022	0.594	0.176
Genmab	0.030	-	0.580	0.108	**0.006**	**0.006**
GN	0.089	0.962	0.871	0.431	0.142	0.029
Jyske Bank	0.034	0.619	-	0.091	0.380	0.431
Lundbeck	0.014	**0.003**	0.037	0.707	0.454	0.026
Mærsk A	0.455	0.463	0.525	0.502	0.959	0.052
Mærsk B	0.326	0.590	0.446	-	0.978	0.092
Novo Nordisk	0.759	0.125	0.080	**0.005**	0.198	**0.001**
Novozymes	0.160	0.237	0.238	0.831	0.262	0.282
Royal Unibrew	0.120	0.307	0.560	0.895	0.097	0.020
SimCorp	0.046	0.054	0.059	0.772	-	0.070
SydBank	0.080	0.049	0.483	0.106	**0.009**	0.324
Tryg	0.054	0.357	0.010	0.102	0.057	0.063
Vestas	0.326	0.749	0.576	0.153	0.030	-
William Demant	0.047	0.063	0.173	0.361	**0.000**	0.364
C20index	0.123	0.260	0.058	0.089	0.015	**0.002**
SP500	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.003**
Two Year Bond	0.052	0.158	0.045	0.360	0.421	0.084
Ten Year Bond	0.153	0.303	0.021	0.576	0.030	0.203
CNY	0.510	0.110	0.398	0.248	0.077	0.036
GBP	0.905	0.221	0.474	0.380	0.046	0.050
NOK	0.066	0.125	0.967	0.510	0.047	**0.002**
USD	0.133	**0.004**	0.730	0.072	0.317	**0.002**
SEK	0.455	0.608	0.593	0.349	0.895	0.445
Gold Futures	0.312	0.160	0.526	0.389	0.257	0.631
Oil Price	**0.000**	0.069	0.073	**0.002**	0.291	0.514

15.2 Appendix 2 - Dependent variables

	Consumption	**Energy**	**Finance**	**Health Care**	**Industry**	**Technology**
Carlsberg	**699.00**	-	-	-	-	-
Royal Unibrew	455.80	-	-	-	-	-
Vestas	-	**100.36**	-	-	-	-
Jyske Bank	-	-	**234.50**	-	-	-
Tryg	-	-	164.80	-	-	-
SydBank	-	-	157.50	-	-	-
Danske Bank	-	-	128.70	-	-	-
Genmab	-	-	-	**1024.50**	-	-
Coloplast	-	-	-	611.20	-	-
Novo Nordisk	-	-	-	304.30	-	-
Novozymes	-	-	-	288.10	-	-
Lundbeck	-	-	-	283.10	-	-
GN	-	-	-	255.60	-	-
William Demant	-	-	-	187.40	-	-
Ambu	-	-	-	162.60	-	-
Mærsk B	-	-	-	-	**7230.67**	-
Mærsk A	-	-	-	-	6776.72	-
DSV	-	-	-	-	441.50	-
FLSmidth	-	-	-	-	298.00	-
SimCorp	-	-	-	-	-	**460.40**

15.3 Appendix 3 - Hyperparameter tuning

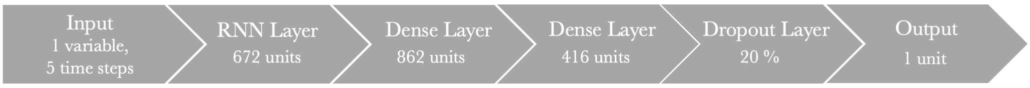
The Keras tuner package provides a variety of hyperparameter selection options. The ones used in this thesis are *range* and *choice*. The *range* is used by setting a minimum and maximum value for a given parameter and defining the step by which it changes. The *choice* is used to chose between predefined values (KerastuneR, 2021).

The number of units in the layers is decided with a range of 32 and 1024 with a step of 32.
The dropout layers are set to decide between a dropout rate of either 0, 0.1, 0.2 or 0.3.
Additional feed forwards layers can be added beside the first hidden layer in a range of 1-5 layers.
Deciding the learning is a choice between 0.01, 0.001 and 0.0001.

15.4 Appendix 4 - Structure for ANN testing models

15.4.1 Structure of ANN for Carlsberg

Figure 19: Carlsberg without explanatory variables

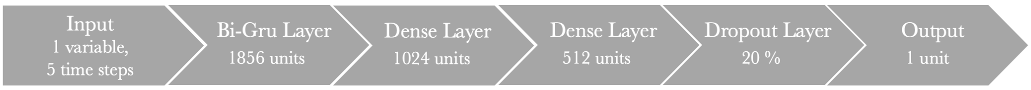


(a) RNN (0.001 LR)

(b) LSTM(0.01 LR)

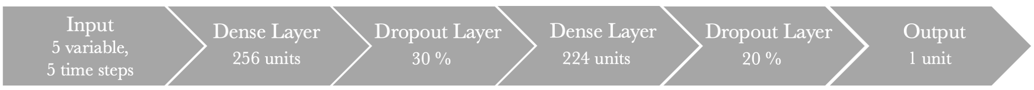
(c) Bi-LSTM(0.01 LR)

(d) GRU(0.01 LR)

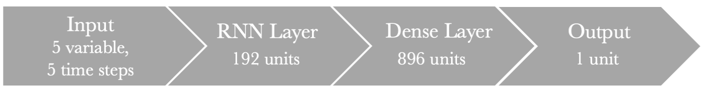


(e) Bi-GRU(0.001 LR)

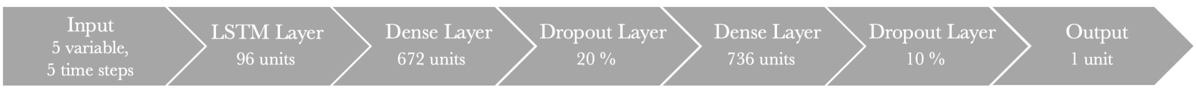
Figure 20: Carlsberg with explanatory variables



(a) FNN (0.01 LR)

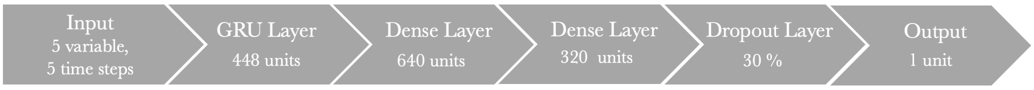


(b) RNN (0.001 LR)



(c) LSTM (0.001 LR)

(d) Bi-LSTM (0.001 LR)

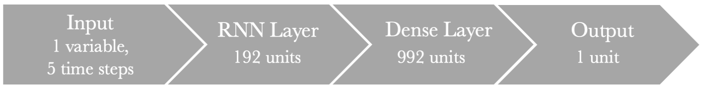


(e) GRU (0.001 LR)

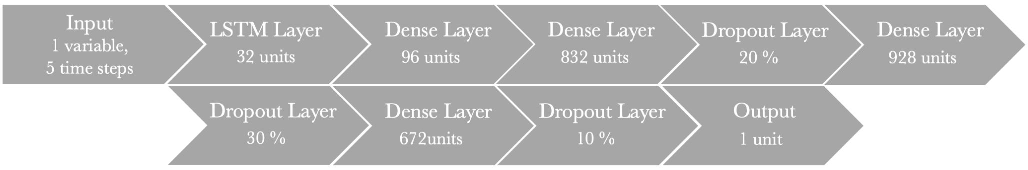
(f) Bi-GRU (0.001 LR)

15.4.2 Structure of ANN for Genmab

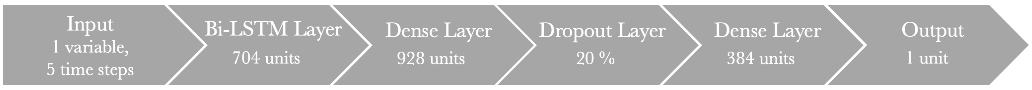
Figure 21: Genmab without explanatory variables



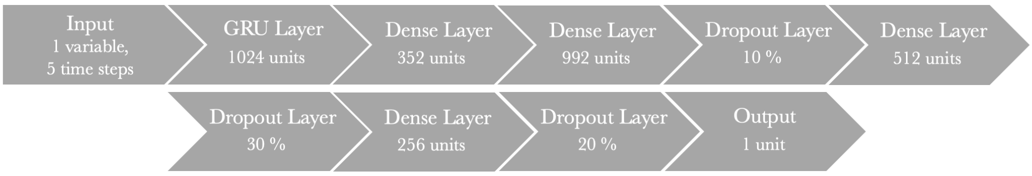
(a) RNN (0.0001 LR)



(b) LSTM (0.0001 LR)



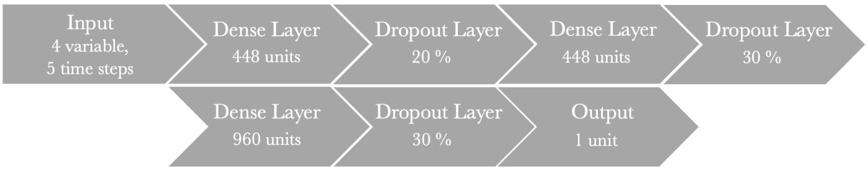
(c) Bi-LSTM (0.01 LR)



(d) GRU (0.001 LR)

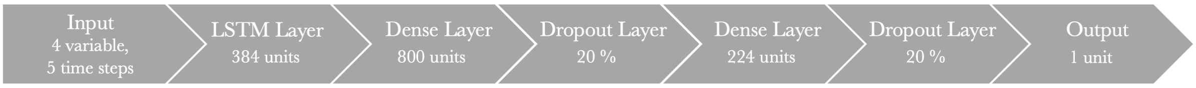
(e) Bi-GRU (0.001 LR)

Figure 22: Genmab with explanatory variables

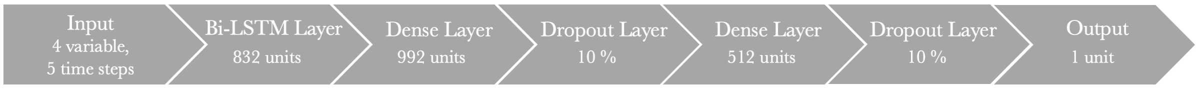


(a) FNN (0.001 LR)

(b) RNN (0.001 LR)



(c) LSTM (0.001 LR)



(d) Bi-LSTM (0.001 LR)

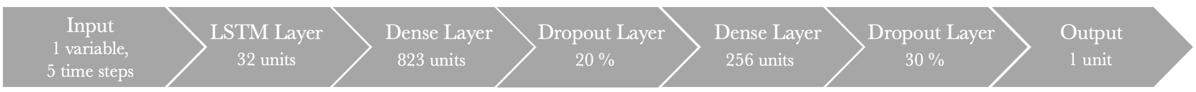
(e) GRU (0.01 LR)

(f) Bi-GRU (0.001 LR)

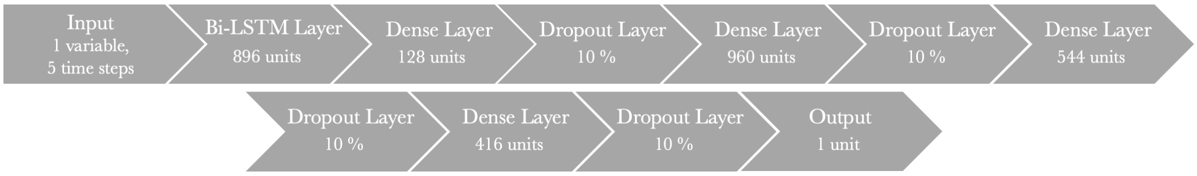
15.4.3 Structure of ANN for Jyske Bank

Figure 23: Jyske Bank without explanatory variables

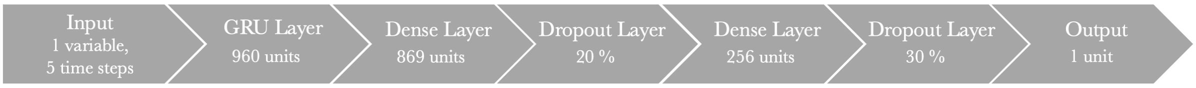
(a) RNN (0.001 LR)



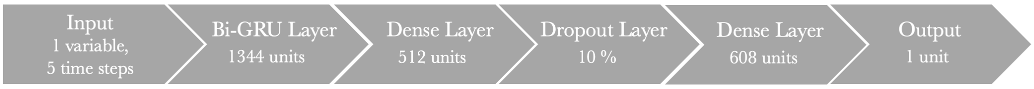
(b) LSTM (0.01 LR)



(c) Bi-LSTM (0.01 LR)

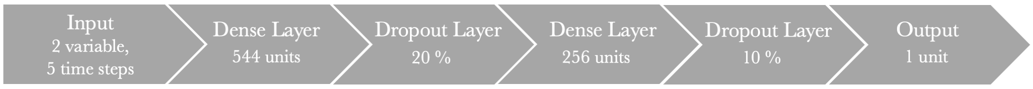


(d) GRU (0.001 LR)

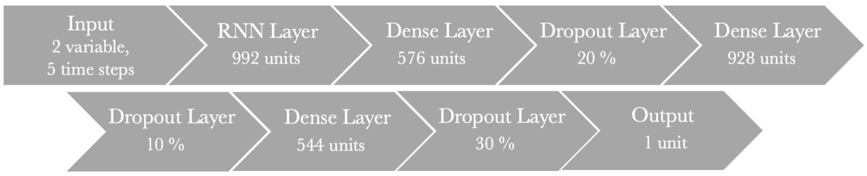


(e) Bi-GRU (0.001 LR)

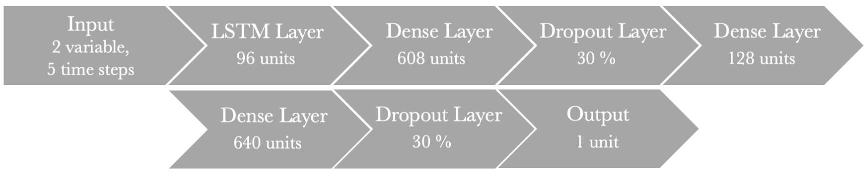
Figure 24: Jyske Bank with explanatory variables



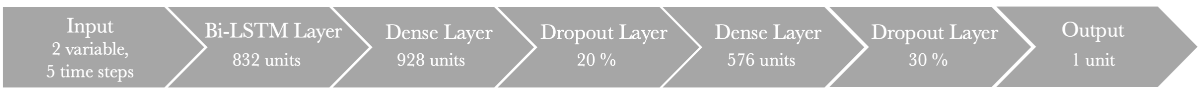
(a) FNN (0.001 LR)



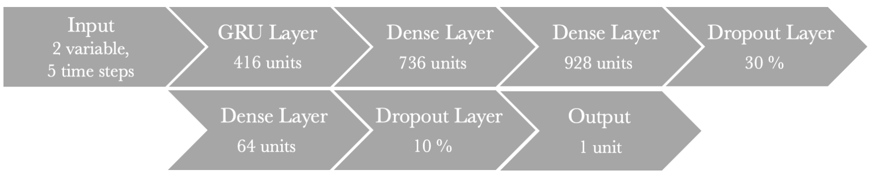
(b) RNN (0.001 LR)



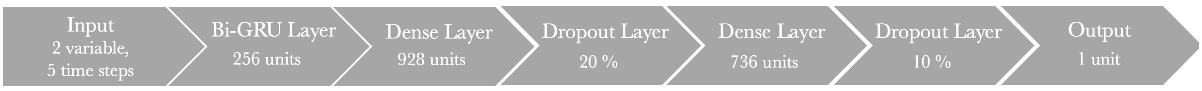
(c) LSTM (0.01 LR)



(d) Bi-LSTM (0.001 LR)



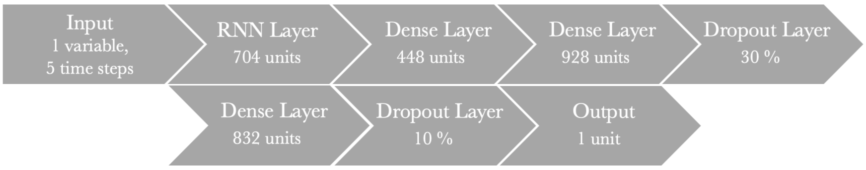
(e) GRU (0.001 LR)



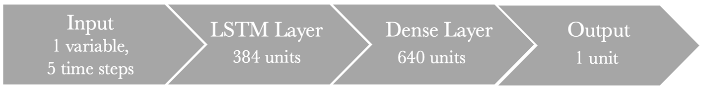
(f) Bi-GRU (0.01 LR)

15.4.4 Structure of ANN for Mærsk B

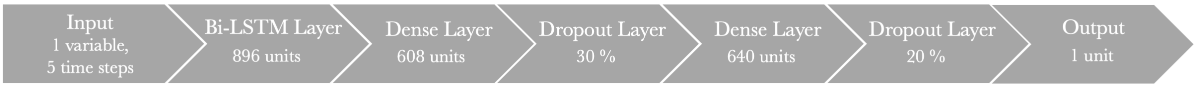
Figure 25: Mærsk B without explanatory variables



(a) RNN (0.001 LR)

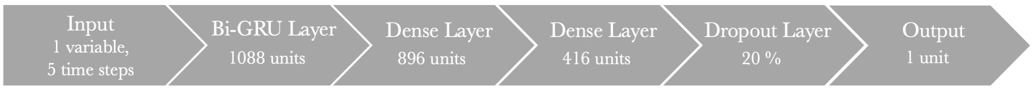


(b) LSTM (0.01 LR)



(c) Bi-LSTM (0.001 LR)

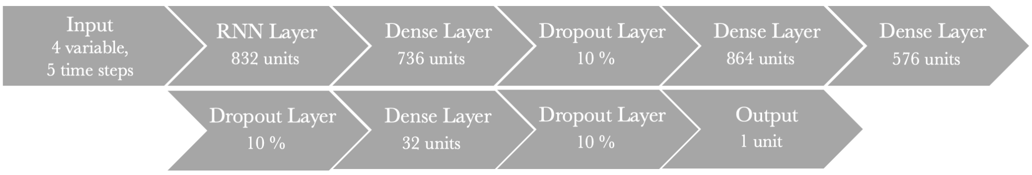
(d) GRU (0.01 LR)



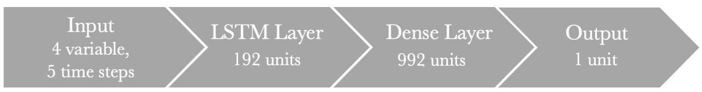
(e) Bi-GRU (0.001 LR)

Figure 26: Mærsk B with explanatory variables

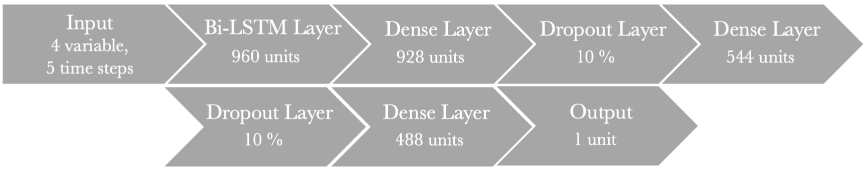
(a) FNN (0.001 LR)



(b) RNN (0.001 LR)

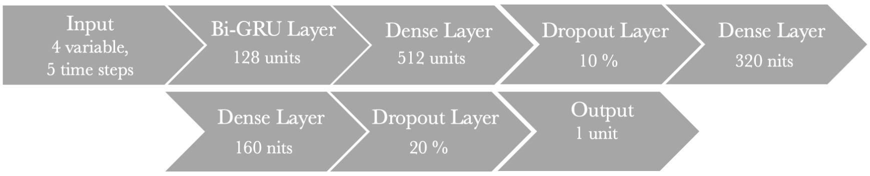


(c) LSTM (0.01 LR)



(d) Bi-LSTM (0.001 LR)

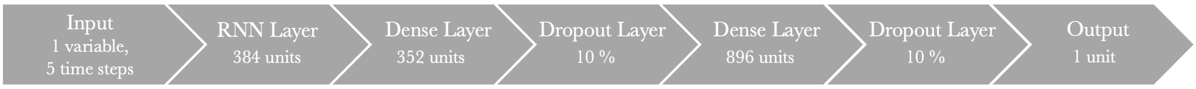
(e) GRU (0.001 LR)



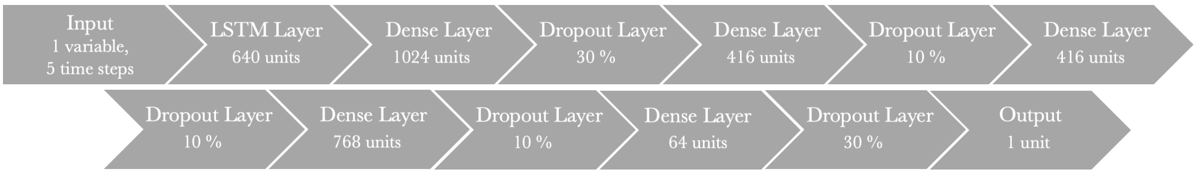
(f) Bi-GRU (0.001 LR)

15.4.5 Structure of ANN for SimCorp

Figure 27: SimCorp without explanatory variables



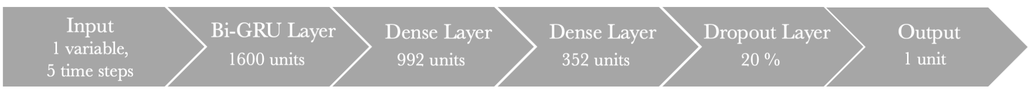
(a) RNN (0.001 LR)



(b) LSTM (0.001 LR)

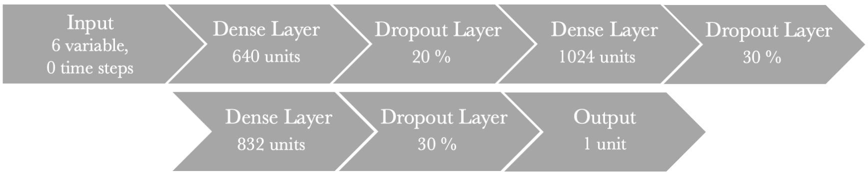
(c) Bi-LSTM (0.001 LR)

(d) GRU (0.001 LR)

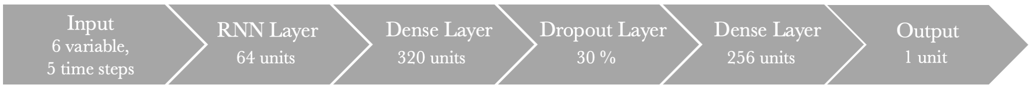


(e) Bi-GRU (0.001 LR)

Figure 28: SimCorp with explanatory variables

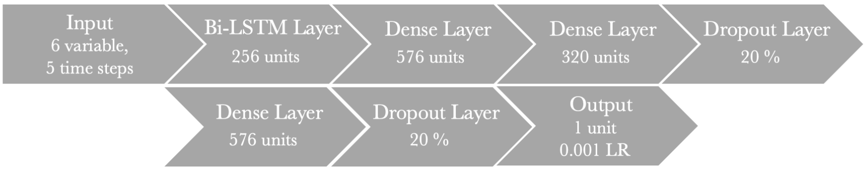


(a) FNN (0.001 LR)

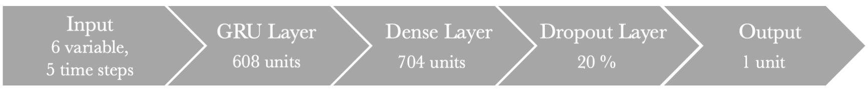


(b) RNN (0.0001 LR)

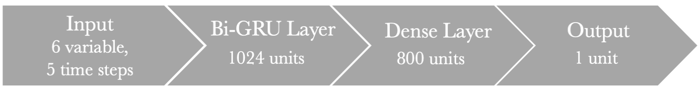
(c) LSTM (0.001 LR)



(d) Bi-LSTM (0.001 LR)



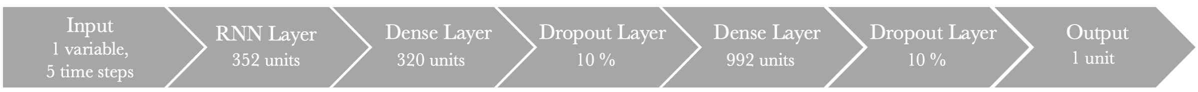
(e) GRU (0.001 LR)



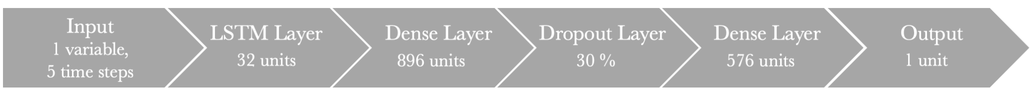
(f) Bi-GRU (0.01 LR)

15.4.6 Structure of ANN for Vestas

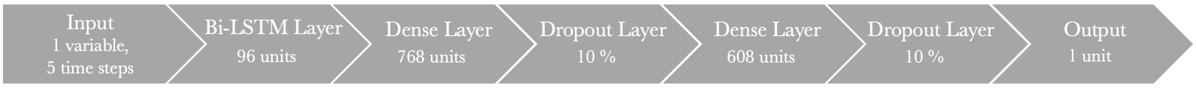
Figure 29: Vestas without explanatory variables



(a) RNN (0.001 LR)

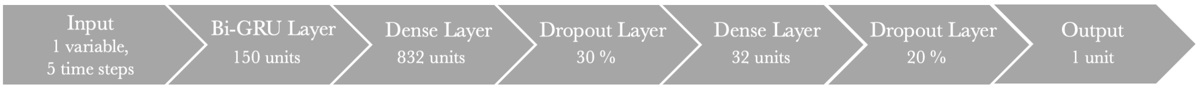


(b) LSTM (0.01 LR)



(c) Bi-LSTM (0.001 LR)

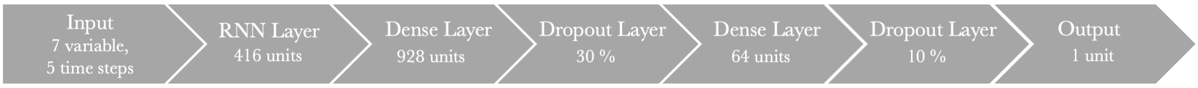
(d) GRU (0.01 LR)



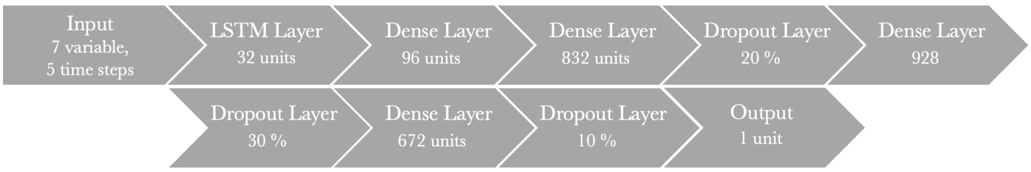
(e) Bi-GRU (0.001 LR)

Figure 30: Vestas with explanatory variables

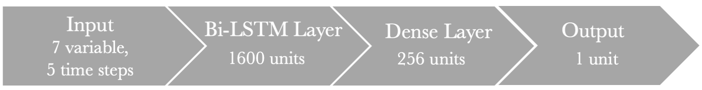
(a) FNN (0.01 LR)



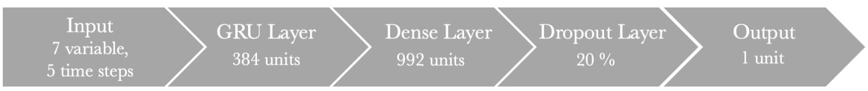
(b) RNN (0.0001 LR)



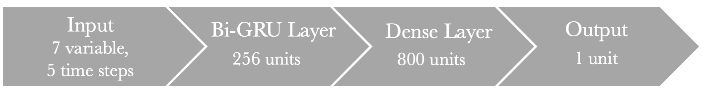
(c) LSTM (0.0001 LR)



(d) Bi-LSTM (0.0001 LR)



(e) GRU (0.001 LR)



(f) Bi-GRU (0.01 LR)

15.5 Appendix 5 - Abbreviation used in the thesis

ADF	Augmented Dickey–Fuller	**AI**	Artificial Intelligence
AIC	Akaike Information Criterion	**ANN**	Artificial Neural Network
AR	AutoRegressive	**ARIMA**	AutoRegressive Integrated Moving Average
ARIMAX	AutoRegressive Integrated Moving Average with external regressors	**ARIMA(X)**	Both ARIMA and ARIMAX
ARMA	AutoRegressive Moving Average	**Bi-GRU**	Bidirectional Gated Recurrent Unit
Bi-LSTM	Bidirectional Long-short-term memory	**CNN**	Convolutional Neural Networks
DM	Diebold-Mariano	**DT**	Decision Tree
ELM	Extreme Learning Machine	**FNN**	Feed-forward Neural Network
GRU	Gated Recurrent Unit	**LSTM**	Long-short-term memory
MA	Moving Average	**MAPE**	Mean Absolute Percentage Error
ML	Machine Learning	**MSE**	Mean Square Error
NA	Missing value	**ReLU**	Rectified Linear Unit
RF	Random Forrest	**RMSE**	Root Mean Square Error
RNN	Recurrent Neural Network	**SML**	Supervised Machine Learning