FORECASTING DURING RECESSION

-

Comparing the performance of machine learning and autoregressive models on the Swedish stock market

Author	Supervisor
Jacob Skoglund	Luca Margaritella

Lund Unviersity
Department of Economics

January 2024

Bachelor's thesis

Acknowledgements

I would like to acknowledge and express my gratitude to my supervisor Luca Margaritella. Without his insight, expertise and assistance this thesis would not have been possible.

Abstract

As the processing power of computers continuously increase so does the interest for machine learning and artificial intelligence. This thesis evaluates the forecasting performance of both machine learning models and common auto-regressive models on the Swedish stock market index OMXS30 on the Stockholm stock exchange during the 2008 financial crises. Forecasts are performed 3, 6 and 12 months ahead. The results indicate that machine learning models perform noticeably better when forecasting 6 and 12 month ahead, while the result for the machine learning models are comparable to those of the autoregressive models when forecasting 3 months ahead.

Keywords: forecasting, machine learning, stock market, Sweden.

Abbreviations

ML	Machine Learning
AI	Artificial Intelligence
AR	Auto Regressive
ARIMA	Auto Regressive Integrated Moving Average
SARIMA	Seasonal Auto Regressive Integrated Moving Average
MA	Moving Average
VAR	Vector Auto Regressive
NN	Neural Network
RNN	Recurrent Neural Network
ANN	Artificial Neural Network
DNN	Deep Neural Network
KNN	K-Nearest Neighbors
NLP	Natural Language Processing
RF	Random Forest
LSTM	Long Short-Term Memory
XGB	Extreme Gradient Boosting
SVM	Support Vector Machine
DF	Dickey-Fuller
ADF	Augmented Dickey-Fuller
AIC	Akaike Information Criterion
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
CPI	Consumer Price Index
Adam	Adaptive Moment Estimation

List of Figures

1 Graphical illustration of the OMXS30 before the forecasted period. . . 3

2 Graphical illustration of time series exhibiting weak-stationarity, seasonality and long term trend. 6

3 Structure of a single node in a neural network where each input value $x_{i,t}$ is multiplied with a corresponding weight w_i and summarised to give the output value y_{t+1}. 9

4 A schematic overview of an ANN employing an input layer, three hidden layers and a single output layer. 10

5 Schematic overview of a RNN where the hidden state h is passed along updating the network. 11

6 Graphical illustration of the Sigmoid- and Hyperbolic tangent function. 12

7 A schematic illustration of a LSTM cell consisting of the three gates: Forget-, Input-, and Output gate. 13

8 Illustration of a decision-tree evaluating the risk for a participant to suffer from an heart attack based on specific features. 14

9 Illustration of a regression-tree and corresponding graphical illustration as partitioned areas. 14

10 AIC score for the AR models using different number of lags. 18

11 AIC score for the VAR models using different number of lags. 18

12 The loss function plotted for the LSTM model using different number of neurons, layers and epoch. 20

13 The loss function plotted depending on the number of estimators for the RF model using different hyperparametares. 22

List of Tables

1 Overview of dependent (OMXS30) and independent (CPI, Export and Oil price) variables used in this thesis. 16

2 Overview of each model in relation to used dependent and independent variables. 17

3 Overview of the results from the four models when forecasting three months ahead. The results are given as the RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) between the forecasted value and the observed value for the OMXS30. 23

4 Overview of the results from the four models when forecasting six months ahead. The results are given as the RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) between the forecasted value and the observed value for the OMXS30. 24

5 Overview of the results from the four models when forecasting twelve months ahead. The results are given as the RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) between the forecasted value and the observed value for the OMXS30. 24

Contents

Acknowledgements i

Abstract i

Abbreviations ii

List of Figures iii

List of Tables iv

1 Introduction 1

1.1 Background 1

1.2 Purpose and aim 2

1.3 Data limitations 3

1.4 Previous research 3

1.5 Disposition 4

2 Theoretical background 6

2.1 Stationarity 6

2.2 Autoregressive models 7

2.2.1 AR 7

2.2.2 VAR 8

2.3 Neural Networks 9

2.4 Machine Learning models 11

2.4.1 LSTM 11

2.4.2 Random Forest 13

2.5 Evaluation Metrics 15

2.5.1 MAE 15

2.5.2 RMSE 15

3 Method 16

3.1 Variables 16

3.2 Models and hyperparameters 17

3.2.1 AR 17

3.2.2 VAR 18

3.2.3 LSTM 18

3.2.4 Random Forest 21

3.3 Software 21

4 Empirical results **23**

5 Discussion and concluding remarks **26**

6 Future Research **28**

Bibliography **29**

References **29**

Appendix A: AR results **32**

Appendix B: VAR results **34**

Appendix C: LSTM results **36**

Appendix D: RF results **38**

1 Introduction

1.1 Background

The ups and downs of the stock market regularly makes headline news as the expansion and contraction of the market is a key indicator of investors aggregated expectation for the future of the economy (Merkle & Weber, 2014). Individuals in general therefore have a vested interest in the stock market as changes in market value have far broader global implications on the economy, impacting more than just stockholders. Among private and corporate investors, the ability to accurately forecast changes within the market would allow for investor to accrue large amounts of wealth, as well as for governments to accurately adjust both monetary and fiscal policies to mitigate the impact of future financial downturns.

The degree by which an investor accurately can forecast future prices on the stock market using historical data is still debated. Some argue for the efficient market hypothesis, stating that all known information is accounted for in the pricing of stocks on the market, while any future information emerges at random, allowing for no single investor to outperform the market (Dahlquist & Knight, 2022). Those who argue that the future of the stock market is completely random are also implicitly stating that past information has no bearing on future stock prices. This view of changes in the stock market as completly random is echoed in Burton Malkiels influential book *A Random Walk Down Wallstreet* from 1973, where Burton argues that any short-term change in the stock market is completely random, going so far as to say that ”a blindfolded monkey throwing darts at the stock listings could select a portfolio that would do just as well as one selected by the experts” (Malkiel, 2016). Yet, with the advent of digitalization and increased computational capacity, the principles of the efficient market hypothesis have been challenged with many researchers arguing that statistic, behavioural and rational models based on historical data do exhibit certain predictive powers (Kumbure, Lohrmann, Luukka, & Porras, 2022). But if Burton Malkiel is correct, and forecasting models are as effective at forecasting future movement on the stock market as flipping a coin, newer more advanced machine learning (ML) models should be equally incapable at market forecasting as the simplest and most primitive models.

Today, financial forecasting and modeling is a key part of any technical market analysis. Yet, it is generally agreed that while stock market forecasting is a relevant and sought after endeavour, the ability of conventional models based on historical data to accurately and consistently predict changes in the stock market is much in doubt (Chen & Hao, 2018). Most previous studies concerned with financial forecasting applies conventional statistical autoregressive time series models such as ARCH (autoregressive conditional heteroscedasticity), ARIMA (autoregressive integrated moving average) or VAR (Vector autoregressive) models. But, with the introduction of artificial intelligence (AI) and machine learning (ML) new statistical methods have garnered wide spread attention within the field of financial forecasting. Unlike conventional time series models, these new models are often described as far better equipped at handling noisy and chaotic nonlinear data and should therefore be better suited to forecasting financial data than conventional models (Kumbure et al., 2022). If ML models are better equipped at handling large, noisy and chaotic data, do ML models outperform simpler statistical models during periods characterised by chaotic and sudden changes, such as recessions?

As interest in the field of ML has increased drastically during the last fifteen years and significant research has been devoted to improving and develop more complex models, there exists somewhat of a noticeable optimism concerning the future ability of both ML and AI amongst the public. Yet, many do argue that literature within the field of ML forecasting tend to claim satisfactory results without comparing them with other simpler statistical models such as autoregressive models. Doing so creates false presumptions regarding the accuracy of ML methods according to some researchers, creating a false apprehension that ML models due to their increased sophistication and complexity are superior to simpler statistical methods (Makridakis, Spiliotis, & Assimakopoulos, 2018).

1.2 Purpose and aim

With the rise of ML models, artificial intelligence and increased computational power we have new reasons to question the principles of the efficient market hypothesis. But also whether these newer models can outperform more conventional statistical time series models. Yet, if the efficient market hypothesis holds true, any forecasting models, regardless of complexity, should be equally incapable at forecasting future changes in the market.

This thesis aim at chiefly discerning whether ML model, such as Recurrent Neural Network (RNN) or Random Forest (RF), can outperform traditional autoregressive models during periods characterised by chaotic and sudden changes such as recessions. Given the previous introduction and discussion, the following question is formulated:

- Can ML models more accurately predict price changes on the Swedish stock market during periods of recession compared to conventional statistical models such as autoregressive models?

1.3 Data limitations

This thesis considers forecasting on the Swedish stock market, specifically the market index OMXS30 (OMX Stockholm 30) which is a weighted average index including the 30 most traded stocks on the NASDAQ Stockholm stock exchange. Although granular data is available down to a frequency of minutes for the OMXS30, this thesis considers a monthly frequency (end of month) as other key economic indicators used as independent variables for both the VAR and RF model are only available on a monthly basis. The same restrictions concerning available data limits the studied time span to a time period starting in 1990.

The studied period contains four recessions: The 90s crises, the IT-bubble, the financial crises of 2008 and the 2020 stock market crash. Due to the limited time span this thesis is limited to the financial crises of 2008 as earlier recessions do not give a sufficiently large training set for the studied models.

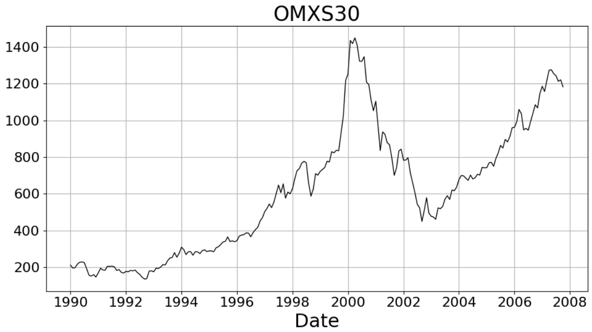

Figure 1: Graphical illustration of the OMXS30 before the forecasted period.

1.4 Previous research

Laymen, corporations and researchers have all taken a keen interest in ML models and their potential uses within the financial market. While many do consider ML models to be the next development in time series modeling, some argue that the increased sophistication of such models does not necessarily have to translate to increased accuracy (Bhattacharjee & Bhattacharja, 2019). When evaluating the accuracy of ML models such as KNN, RNN, LSTM and RF on historical stock data researchers have on occasion found that ML models do outperform traditional forecasting models (Bhattacharjee & Bhattacharja, 2019). Yet, when Makridakis et al. (2018) compared the forecasting ability of simple statistic and ML models on a data set of over a thousand time series from the business and finance sector, the researchers found that the forecasting accuracy of ML models were lower than that of conventional forecasting models, such as the ARIMA model.

A core problem discussed in relation to forecasting financial markets is the inherit non-linearity, volatility and unpredictability of financial data. Proponents of ML models argue that these models are better adapted to handling non-linearity and therefore more suited to forecasting on financial data (Vijh, Chandola, Tikkiwal, & Kumar, 2019). When Kavinnilaa et. al. (2021) evaluated the forecasting accuracy of RNN models in relation to common statistical models the authors concluded that the LSTM model achieved the highest accuracy. Similar results were achieved by Sonkavde et al. (2023) when forecasting changes on the European stock market, also stressing the importance of accurately tuning model hyperparameters.

The performance of forecasting models on the Swedish stock market has been evaluated previously, chiefly in academic papers written by students. Most of these thesis, such as Hellman and Pilerot (2017) or Andréasson and Blomquist (2020), indicates that Recurrent neural network (RNN) in general outperform or perform as well as autoregressive models when tested on data from the Swedish stock market (Pilerot & Hellman, 2017; Andréasson & Blomquist, 2020). These thesis all employ an ARIMA (autoregressive integrated moving average) model as a reference when evaluating the result of different ML models. Unlike previous papers this thesis considers a vector autoregressive (VAR) model as a benchmarking model used to compare with the results of the ML models. While the ARIMA model forecast future values of the time series based on previous observations and trends within the same series, the VAR model is a form of multivariate analysis which models the behaviour of the studied time series based on its statistical relationship to historical data of other time series (Shumway & Stoffer, 2017).

1.5 Disposition

- Chapter one introduces the thesis background, purpose and previous research.
- Chapter two considers the theoretical background of the used models as well as a background regarding time series, time series modeling and neural networks.
- Chapter three describes the underlying hyperparameters of each model along with information regarding dependent and independent variables used for the

models.

- Chapter four presents the results for each model when forecasting 3, 6 and 12 months ahead during the studied period of November 2007 and October 2008.
- Chapter five discusses the results and thesis in general as well as giving concluding remarks.
- Chapter six discusses future research.

2 Theoretical background

2.1 Stationarity

A key assumption of any autoregressive models is that of *weak-stationarity* as the model assumes that the computed time series represents a realisation of a stochastic process centered around a constant mean value (μ) independent of time. For any time series to be considered weakly stationary it needs to exhibit the same statistical behaviour and a constant probability distribution regardless of analysed time period. Therefore time series that display long term trends, such as GDP, or seasonality, such as local temperature, would not be considered weakly stationary (Montgomery, Jennings, & Kulahci, 2008), for graphical illustration of different time series see Figure 2.

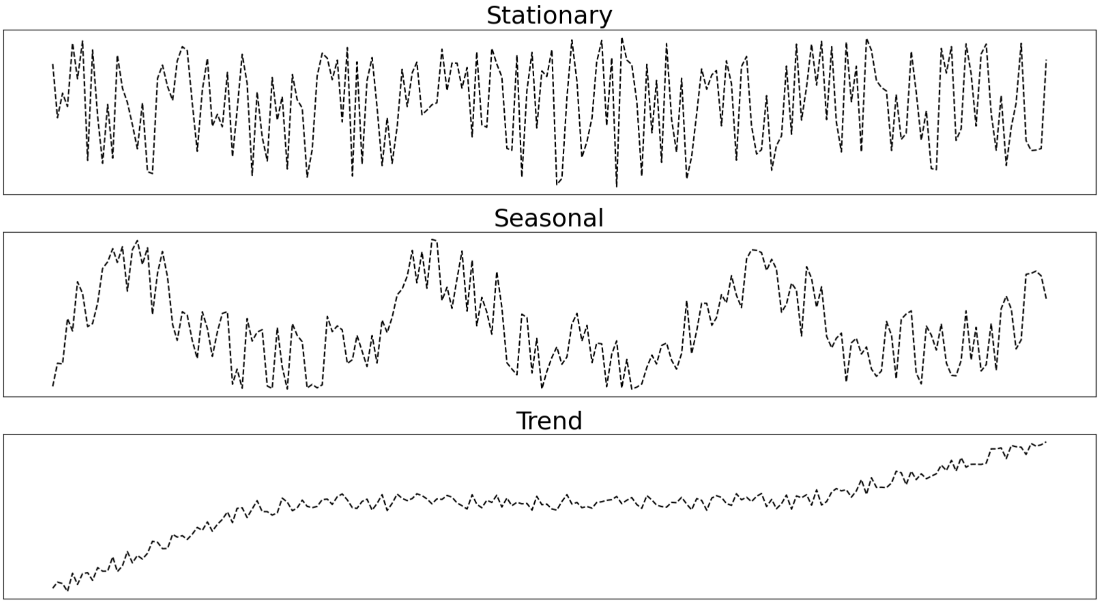

Figure 2: Graphical illustration of time series exhibiting weak-stationarity, seasonality and long term trend.

Yet, ascertaining whether a time series is weakly stationary or if it exhibit different statistical properties dependent on time is not obvious. If a time series y_t is weakly stationary it needs to fullfill the following criteria for any $t, h \in \mathbb{N}$:

$$\begin{array}{r|l} E(y_t) = \mu & \text{Mean is independent on time} \\ \text{VAR}(y_t) = \sigma^2 & \text{Variance is independent on time} \\ \text{COV}(y_t, y_{t+h}) = p_h & \text{Constant Auto-correlation independent on time} \end{array}$$

Generally, when testing for stationarity the Dickey-Fuller test (DF) is employed which considers a simple AR(1) (autoregressive) model without time trend or drift given by

Equation 2.1 where ε_t denotes white noise.[1] See Section 2.2.1 for further discussion of the AR model.

$$y_t = \phi y_{t-1} + \varepsilon_t \tag{2.1}$$

Subtracting the lagged value of the time series, y_{t-1} on both sides:

$$\underbrace{y_t - y_{t-1}}_{\Delta y_t} = \underbrace{(\phi - 1)}_{\rho} y_{t-1} + \varepsilon_t \tag{2.2}$$

$$\Delta y_t = \rho y_{t-1} + \varepsilon_t \tag{2.3}$$

As the term ε_t represents white noise we know $E(\varepsilon_t) = 0$, $Var(\varepsilon_t) = \sigma^2$ and $Cov(\varepsilon_t, \varepsilon_s) = 0 \ \forall \ t \neq s$. As a result, any time series with $\rho < 0$ exhibiting the same statistical properties independent of time would result in a stationary model. Therefore the following null hypothesis is formulated and tested using OLS-regression for the estimator ϕ:

$H_0 : \rho$ =0, time series contains a unit root and is not stationary
$H_1 : \rho$ <0, time series has no unit root and is stationary

For time series data exhibiting higher order correlation, the so called Augmented Dickey-Fuller test (ADF) is used which, unlike the DF test, considers an AR(p) model:

$$y_t = \mu + \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + \varepsilon_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t \tag{2.4}$$

Subtracting the lagged value of the time series, y_{t-1} on both sides:

$$\Delta y_t = \mu + \rho y_{t-1} + \sum_{i=1}^{p} \phi_i \Delta y_{t-i} + \varepsilon_t \tag{2.5}$$

The hypothesis testing for the ADF is the same as for the regular DF test.

2.2 Autoregressive models

2.2.1 AR

An autoregressive (AR) model describes a process in which the output variables of the model is solely determined by its own previous values and an error term. The order p of the autoregressive model indicates the number of lags of the model, i.e, the number of lagged parameters the output of the model is dependent on (Peixeiro,

[1] The Dickey-Fuller unit root test commonly considers three AR(1) models, including and or excluding drift and linear time trend, see Dickey-Fuller 1976.

2022). Mathematically the autoregressive model AR(p) is described as:

$$y_t = \mu + \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + \varepsilon_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t \qquad t = 1, ..., T \tag{2.6}$$

Where μ is a constant, $\phi_1 ... \phi_p$ are the autoregressive coefficients and ε_t indicates a realisation of a white noise process.

2.2.2 VAR

The vector autoregressive model (VAR) is commonly used for evaluating the relationship between time series. The degree by which the VAR model can predict future values of the time series is both dependent on the selected variables included within the model and their correlation. Unlike the AR model, the VAR model is a multivariate extension considering multiple time series (Shumway & Stoffer, 2017). Consider a multivariate system consisting of three time series y_t, x_t and z_t, dependent on p lags of themselves and the other variables:

$$\begin{aligned} y_t &= \mu_y + \phi^1_{11} y_{t-1} + \phi^1_{12} x_{t-1} + \phi^1_{13 t-1} + ... + \phi^p_{11} y_{t-p} + \phi^p_{12} x_{t-p} + \phi^p_{13} z_{t-p} + \varepsilon_{y,t} \\ x_t &= \mu_x + \phi^1_{21} y_{t-1} + \phi^1_{22} x_{t-1} + \phi^1_{23 t-1} + ... + \phi^p_{21} y_{t-p} + \phi^p_{22} x_{t-p} + \phi^p_{23} z_{t-p} + \varepsilon_{x,t} \\ z_t &= \mu_z + \phi^1_{31} y_{t-1} + \phi^1_{32} x_{t-1} + \phi^1_{33 t-1} + ... + \phi^p_{31} y_{t-p} + \phi^p_{32} x_{t-p} + \phi^p_{33} z_{t-p} + \varepsilon_{z,t} \end{aligned}$$

Where μ is the mean of the time series, ϕ are the autoregressive coefficient and ε is the white noise of the model. The VAR model written in matrix form:

$$\begin{bmatrix} y_t \\ x_t \\ z_t \end{bmatrix} = \begin{bmatrix} \mu_y \\ \mu_x \\ \mu_z \end{bmatrix} + \begin{bmatrix} \phi^1_{11} & \phi^1_{12} & \phi^1_{13} \\ \phi^1_{21} & \phi^1_{22} & \phi^1_{23} \\ \phi^1_{31} & \phi^1_{32} & \phi^1_{33} \end{bmatrix} \begin{bmatrix} y_{t-1} \\ x_{t-1} \\ z_{t-1} \end{bmatrix} + ... + \begin{bmatrix} \phi^p_{11} & \phi^p_{12} & \phi^p_{13} \\ \phi^p_{21} & \phi^p_{22} & \phi^p_{23} \\ \phi^p_{31} & \phi^p_{32} & \phi^p_{33} \end{bmatrix} \begin{bmatrix} y_{t-p} \\ x_{t-p} \\ z_{t-p} \end{bmatrix} + \begin{bmatrix} \varepsilon_{y,t} \\ {\scriptstyle x,t} \\ {\scriptstyle z,t} \end{bmatrix}$$

Written in a reduced form as:

$$\boldsymbol{Y}_t = \boldsymbol{\delta} + \sum_{i=1}^{p} \boldsymbol{\Phi}_i \boldsymbol{Y}_{t-i} + \boldsymbol{\varepsilon}_t \tag{2.7}$$

Where:

p	Number of lags
T	Number of variables
$\boldsymbol{Y}_t$	n×1 vector containing the variables
$\boldsymbol{\delta}$	n×1 vector containing the intercepts
$\boldsymbol{\Phi}_i$	n×n matrix of coefficients
$\boldsymbol{Y}_{t-i}$	n×n matrix of i lagged variables
$\boldsymbol{\varepsilon_t}$	n×1 vector of error terms

With any autoregressive model it is critical to determine the optimal lag length for the model. A common mean of lag length selection is through the use of selection criteria such as *Aiaike Information Criterion* (AIC) or *Bayesian information criterion* (BIC). These criteria evaluates the sum of squared residuals for including additional parameters in the model, with the model achieving the lowest value being considered the best fitting model (Chakrabarti & Ghosh, 2011). According to Chakrabarti and Ghosh (2011) the AIC is more appropriate when evaluating the best model to use for future predictions, while the BIC is more useful when selecting a more correct model in relation to known data.

$$AIC = 2k - 2ln(\hat{L})$$
$$BIC = k \cdot ln(n) - 2 \cdot ln(\hat{L})$$

$Where:$

n	Number of observations
k	Number of parameters
$\hat{L}$	Maximized value of the likelihood function

2.3 Neural Networks

A neural network (NN) is a structure of interconnected neurons arranged in layers. Each neuron acts as a function which given an input returns a calculated output which is weighted and summarised together with the output of the remaining neurons within the layer (Peixeiro, 2022). In a simple linear model each underlying feature x is multiplied by a corresponding weight w and summarised to give the prediction of the next value in the time series, see Figure 3. The mathematical expression for the next value in the time series using a simple linear model can therefore be described as:

$$y_{t+1} = w_1 x_{1,t} + w_2 x_{2,t} + w_3 x_{3,t} + w_4 x_{4,t} + w_5 x_{5,t} \tag{2.8}$$

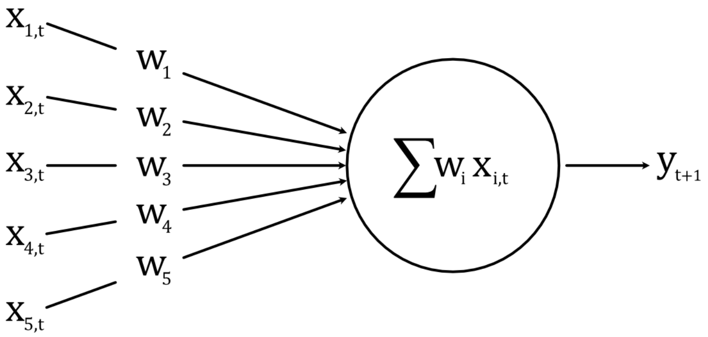

Figure 3: Structure of a single node in a neural network where each input value $x_{i,t}$ is multiplied with a corresponding weight w_i and summarised to give the output value y_{t+1}.

More complex neural networks employ so called hidden layers and are generally referred to as *deep neural networks* (DNN) or *artificial neural networks* (ANN) depending on the number of hidden layers or depth of the model. Artificial neural network consist structurally of three types of layers: the input, hidden and output layer, see Figure 4. The hidden layers allow for a non-linear relationship between the input data and the output prediction, resulting in a more complex but generally more accurate model. The added benefit of employing additional layers is that the model is given more opportunities to learn and thus improve its performance (Peixeiro, 2022). In general it is better to include a large number of hidden neurons rather than too few, as a model with an insufficient amount of neurons may not be able to adequately capture the non-linearity of the data. Yet, NN which employ too many neurons may result in an overfitted and computationally expensive model which accurately captures the trend of known training data, but fails at making accurate prediction for future values (Hastie, Tibshirani, & Friedman, 2009).

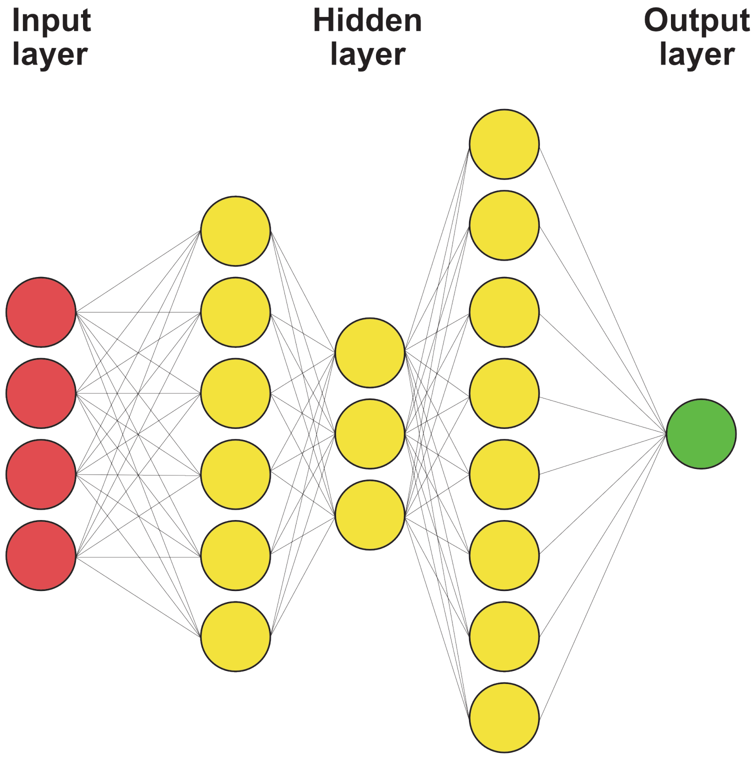

Figure 4: A schematic overview of an ANN employing an input layer, three hidden layers and a single output layer.

A common way to assess the optimal number of neurons for a given model is through cross-validation. A process where the known sample data is portioned into two data sets, a training set used to train the data on known values and a test set which is used to calculate the error of the models predictions. As the number of neurons increases, the error of the predictions decreases until a point where the error term converges, indicating the optimal number of neurons (Hastie et al., 2009).

A recurrent neural network (RNN) is a form of deep learning architecture which is especially adapted to processing sequences of data such as time series, natural language processing (NLP) or audio where the order of the data matters. Unlike common NN, the RNN effectively uses previous information from the sequence to inform the output for the next element, creating a form of memory which allows for previous elements to have an impact on future predictions. The RNN employs a so called hidden state, denoted as h in Figure 5, which is computed and updated before it is passed on to the next element in the sequence.

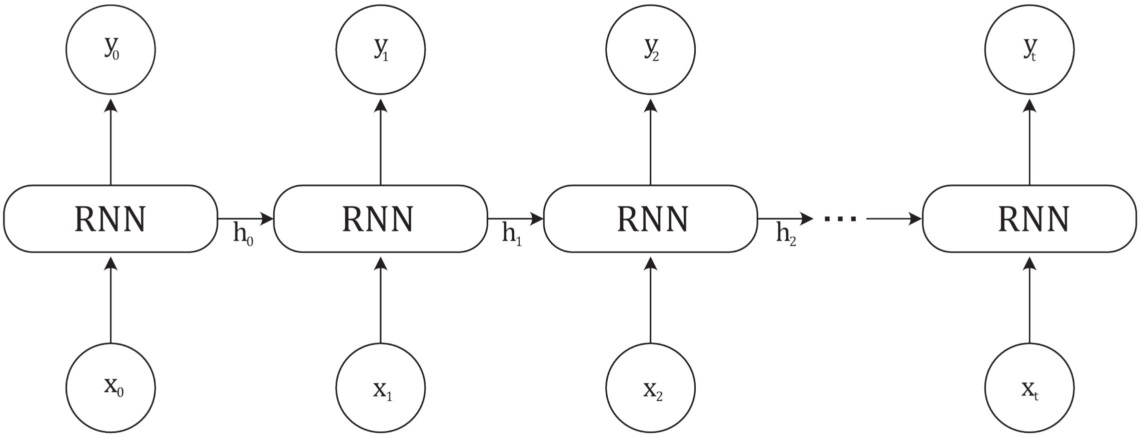

Figure 5: Schematic overview of a RNN where the hidden state h is passed along updating the network.

A problem generally associated with RNN is that of the *vanishing gradient problem.* The gradient is the function which changes the weights of historical data in the network, if the gradient is very small the weights of the network seize to be updated for older historical information. As a result the network only learns from relatively recent information, creating a short-term and less effective neural network (Mandic & Chambers, 2001). The problem of vanishing gradient is generally solved through long short-term memory (LSTM) architecture which is explained in Section 2.4.1.

2.4 Machine Learning models

2.4.1 LSTM

The long short-term memory (LSTM) network is a type of RNN with a complex system architecture which allows for the individual cells to consider what information to store and what information to forget. The LSTM network was created with the intention of addressing the vanishing gradient problem common amongst RNN models by adding the cell state which allows for the network to keep past information within the network for longer periods of time (Sherstinsky, 2020).

The LSTM cell acts as a point in the flow of information where new and historical information is evaluated and filtered before it is passed along the network. A core part of any ANN, and especially LSTM cells, are the so called activation functions. The activation function transforms the input value into a output value within a limited range which allows for a more logical processing of data within the network (Hastie et al., 2009). The LSTM cell considers two separate non-linear activation functions: the sigmoid function and the hyperbolic tangent function (tanh).

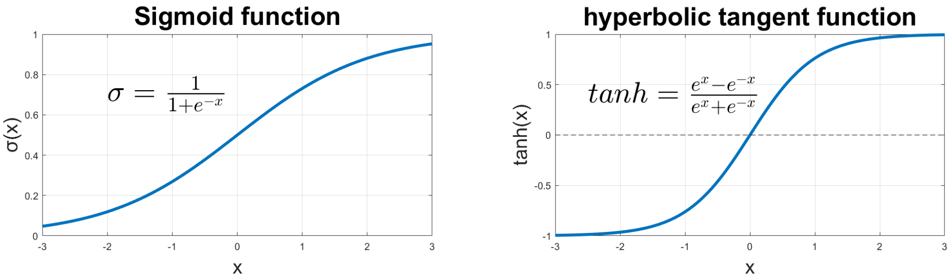

Figure 6: Graphical illustration of the Sigmoid- and Hyperbolic tangent function.

The sigmoid function maps any input value within the range (0,1) and can be viewed as a form of probability in decimal form given by Equation 2.9. For the LSTM network the sigmoid function indicates to what degree previous information within the network is to be included, with zero resulting in completely discarding the information and a one indicates keeping all of the information.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.9}$$

The hyperbolic tangent function (tanh) is a non-linear activation function just like the sigmoid function. But, unlike the sigmoid funtion, the hyperbolic tangent function maps the input value within a range between (1,-1) and is given by equation 2.10. See Figure 6 for a visual representation of both the sigmoid and tanh activation function.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.10}$$

Figure 7 outlines the architecture of a common LSTM cell divided into three separate processes called *gates*, each with a separate function.

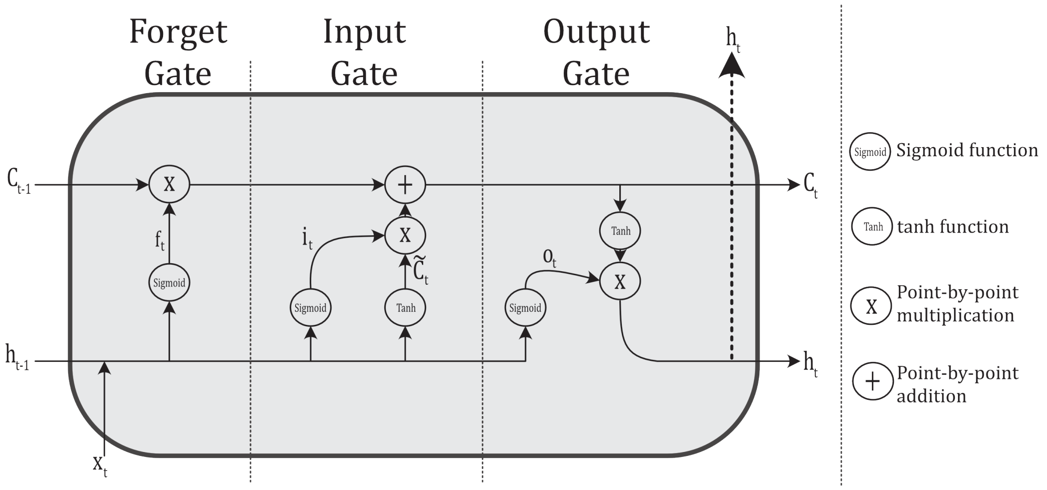

Figure 7: A schematic illustration of a LSTM cell consisting of the three gates: Forget-, Input-, and Output gate.

- The **forget gate** determines what information from previous steps are still relevant and should be passed along the sequence. The present element x_1 is combined with previous information h_{t-1} and duplicated being sent both to the input gate and to the sigmoid activation function. The information is then pointwise multiplied with the previous cell state ,C_{t-1}, generating a new updated cell state C_t.
- The **input gate** evaluates what information from the current step is to be kept within the network. The information $h_{t-1} + x_t$ is passed along from the forget gate and once again duplicated and sent to the sigmoid function which determine whether the information is to be kept or discarded, while the hyperbolic tangent function ($tanh$) regulates the efficiency of the network. The resulting information is pointwise multiplied to one another and then pointwise added to the cell state, C_t resulting in an updated and final cell state.
- The **output gate** is the point where the new hidden state, h_t, and the cell state, C_t, is constructed and passed along to the next LSTM neuron in the next stage. The information $h_{t-1} + x_t$ is passed through the sigmoid function and point-by-point multiplied with the output of the updated cell state C_t and then passed through the hyperbolic tangent function (tanh) creating a new hidden state h_t.

2.4.2 Random Forest

RF is a form of ML method which uses decision-trees to solve problems concerning classification and regression. A decision-tree is a form of supervised learning consisting of a hierarchical structure which acts like a decision process where a given input is processed resulting in a specific output. The input value flows from the root of the decision-tree along branches to internal nodes which represents a decision based on specific features (Ho, 1995), see Figure 8 for illustration of a common decision tree.

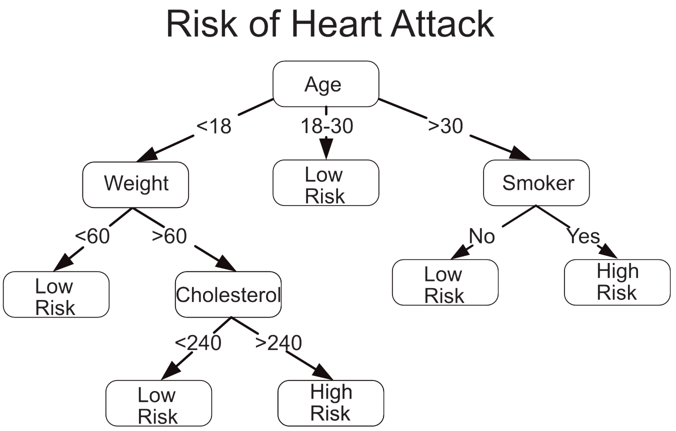

Figure 8: Illustration of a decision-tree evaluating the risk for a participant to suffer from an heart attack based on specific features.

Regression based problems considers a regression tree which, unlike the decision tree, handles continuous variables instead of classes where each leaf (potential output) represents a numeric value. Each leaf of the regression tree considers a partitioned space or region represented by a constant value (Sammut & Webb, 2010), see Figure 9.

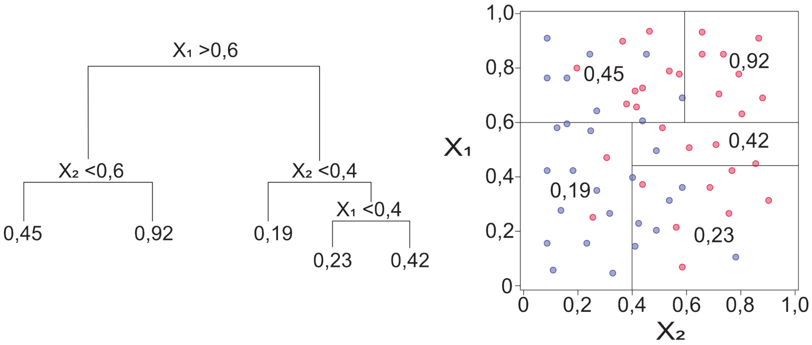

Figure 9: Illustration of a regression-tree and corresponding graphical illustration as partitioned areas.

Single decision- or regression trees have one aspect which prevents them from being ideal methods for predicting or forecasting, namely inaccuracy. While single trees do exhibit a high degree of accuracy when evaluated on the same data used to train them, single trees perform poorly on new unknown data. Therefore, the RF considers multiple decision- or regression trees in an ensemble with trees built in parallel, utilising a resampling method called bootstrapping. Each individual tree within the RF is trained on a subset of the training data and a random selection of features resulting in a large number of trees, each individually different from the other. When forecasting on new data the RF algorithm aggregates the prediction of all individual tress within the ”forest”, creating a final aggregated prediction (Hastie et al., 2009).

2.5 Evaluation Metrics

Conventionally when evaluating the performance of any forecasting model the forecasted values are compared with historical data in an effort to asses the accuracy of a given model. This approach of evaluating models based on residuals is commonly referred to as assessing the *goodness of the fit*, with the best fitting model exhibiting the smallest error. This thesis considers two forms of error measurement commonly used when evaluating the accuracy of time series models: MAE and RMSE (Montgomery et al., 2008).

2.5.1 MAE

The mean absolute error (or MAE) consider the linear relationship between the forecasted and observed value, where the average of the sum of all residuals indicates the error of the model, see Equation 2.11.

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|e_t| \quad \text{where:} \begin{array}{r|l} n & \text{Number of observations} \\ e_t & \text{Residual} \end{array} \tag{2.11}$$

2.5.2 RMSE

While the MAE indicates the linear relationship between the forecasted and observed value, the root mean squared error (or RMSE) punishes models with large individual residuals. Unlike the MAE, the relationship between forecasted and observed value is quadratic, premiering forecasting models with less noisy residuals, see Equation 2.12.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(e_t)^2} \quad \text{where:} \begin{array}{r|l} n & \text{Number of observations} \\ e_t & \text{Residuals} \end{array} \tag{2.12}$$

3 Method

3.1 Variables

As both the VAR and RF model use independent variables, these variables need to be selected with care. Table 1 describes all variables considered in this thesis starting from the year 1990 with a monthly frequency.

Variable	Description	Data Source
OMXS30	Index consisting of a weighted portfolio including the 30 largest traded stocks on the NASDAQ Swedish stock market.	(NASDAQ OMX NORDIQ, 2023)
CPI	Yearly average consumer price index in Sweden on monthly basis	(Statistiska Centralbyrån, 2023a)
Export	Total Swedish export on a mothly basis	(Statistiska Centralbyrån, 2023b)
Oil price	Crude oil price ($ per barrel)	(Federal Reserve Bank of St Louis, 2023)

Table 1: Overview of dependent (OMXS30) and independent (CPI, Export and Oil price) variables used in this thesis.

As it is common for many economic indicators to be reported on a quarterly or yearly basis, this thesis only considers economic indicators which are reported at least at a monthly basis. Another aspect which impacts the selection of variables is the need for the selected variable to have available data stretching back to the 1990s. In selecting the relevant independent variables used for the multivariate models, this thesis considers arguments from relevant academic litterateur concerned with evaluating the relationship between shifts in economic indicators and changes on relevant financial markets.

The *consumer price index* (CPI) considers the relative difference in total cost of a weighted average basket of a typical urban consumer between two different time periods. The CPI is a common metric to measure inflation, i.e the relative price increase in goods and services for a country. During periods of increased inflation it is common for the central bank to increase interest rates, resulting in decreased investment, spending and output, risking a potential future recession (Blanchard & Johnson, 2013). Most articles evaluating the relationship between inflation and short-term stock return find a clear negative correlation, such as Anari and Kolari (2014) study on the Canadian, US, UK, French, German and Japanese stock market.

Since *export* is a key component in any countries GDP, export is generally regarded as a key economic indicator when determining the state of a nations economy (Blanchard & Johnson, 2013). An increased export would indicate an expanding economy resulting in a positive shift on the stock market (Dahlquist & Knight, 2022). When studying the relationship between exports and stock market performance of European countries Mads Asprem (1989) found strong correlation between lagged export values and stock prices. The same strong correlation between export and changes in the stock market was observed by Celebi and Köning (2019) when evaluating the impact of macro economic factors on the German stock market.

The impact of sudden changes in oil prices on global financial markets have been much discussed since the 1970s energy crisis. During recent decades the number of academic papers discussing the topic has increased noticeably, with most papers, such as Degiannakis, Filis & Arora (2018), indicating a strong correlation between increased crude oil price volatility with increased volatility on financial markets. In relation to the European market, it is especially sudden shocks in crude oil price which seems to have a statistically significant impact on stock market return (Park & Ratti, 2008).

See Table 2 for an overview of models and corresponding dependent and independent variables.

	AR	VAR	LSTM	RF
Dependent variable	OMXS30	OMXS30	OMXS30	OMXS30
Independent Variables		CPI Export Crude Oil		CPI Export Crude Oil

Table 2: Overview of each model in relation to used dependent and independent variables.

3.2 Models and hyperparameters

3.2.1 AR

As previously discussed in Section 2.2.1 the number of lags (or order p) for the AR model describes the number of previous values considered in the forecast. To determine the optimal number of lag, the AIC for the AR model has been calculated using different number of lags, see Figure 10. From the figure the optimal number of lags is determined as 9, since the AR(9) model exhibit the lowest AIC score. As mentioned in Section 2.2.2 the AIC score is generally regarded as a better metric for evaluating the predictive performance of a model, which is why this thesis considers the AIC and not the BIC for time lag selection.

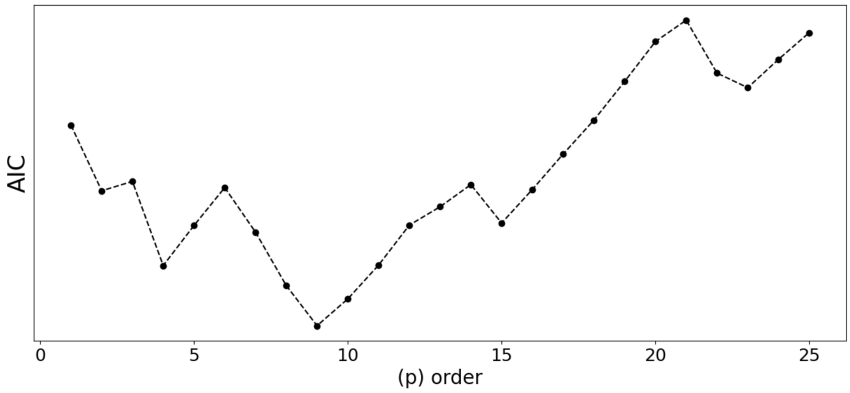

Figure 10: AIC score for the AR models using different number of lags.

3.2.2 VAR

The number of lags for the VAR model is determined using the same AIC metric as used for the AR model. Just like with the AR model, the VAR model with the lowest AIC score is considered to use the optimal number of lags. According to the AIC plot displayed in Figure 11 a VAR model using 14 lags was determined to be optimal.

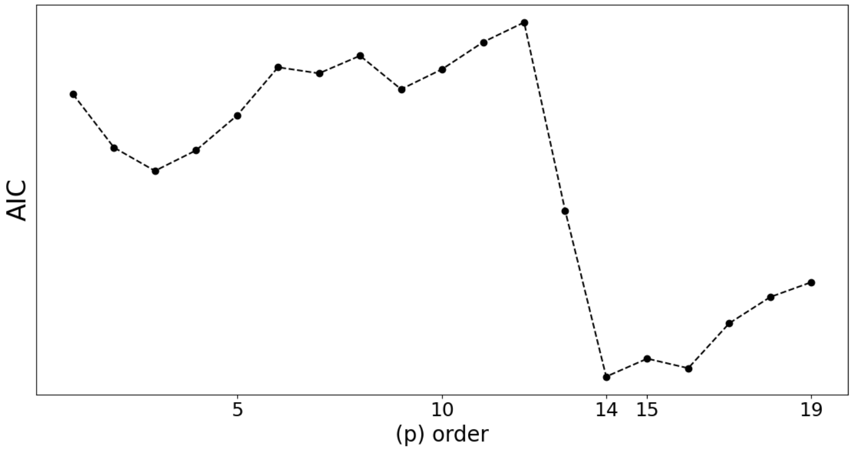

Figure 11: AIC score for the VAR models using different number of lags.

3.2.3 LSTM

The LSTM model considers a multitude of adjustable hyperparameters:

- *Neurons*: The number of neurons within each hidden layer.
- *Hidden Layers*: The number of hidden layers of neurons between the output and input layer.

- *Epochs*: The number of times the training set is run through the model allowing the internal parameters to ”learn” and optimise to more adequately fit the training data.

When selecting the hyperparameters of the model a form of grid search is performed where models using different hyperparameters are trained and tested on historical data. The so called loss function is commonly used for hyperparameter estimation which considers the difference between the estimated and true value (Hastie et al., 2009). Although there exist a wide variety of loss functions with different purposes, the LSTM model in this thesis is assessed using the *mean squared error* (MSE) which is mathematically described in Equation 3.1.

$$MSE = \frac{1}{n}\sum_{t=1}^{n} e_t^2 \quad \text{where:} \quad \begin{array}{r|l} n & \text{Number of observations} \\ e_t & \text{Residual} \end{array} \tag{3.1}$$

To improve the performance of the LSTM models and minimise the loss function different optimizers are selected depending on the nature of the data set and model. This thesis considers the optimizer *Adaptive Moment Estimation* commonly referred to as *Adam*, which according to Kingma and Ba (2015) is an appropriate optimizer to use with non-stationary and noisy data. The Adam optimizer also has the added benefit of being quick to converge, less computationally intensive and commonly used with LSTM models which makes it a fitting optimizer for this thesis (Peixeiro, 2022).

When selecting model hyperparameters it is important not to select too few hidden layers or neurons as a too simple model may fail to capture the flexibility and non-linearity of the data. Yet, creating a far to complex model may result in an overfitted model less accurate when forecasting future values. Figure 12 illustrates the mean squared error for the model using the Adam optimizers depending on number of hidden layers, number of neurons in each layer and number of epochs. As indicated by the figure the MSE quickly converges around 10 epochs but the total error slightly decreases by each new epoch. In general it can be stated that the models using more neurons within each hidden layer performs better than those with less neurons. This thesis considers a LSTM model with the following hyperparameters:

Hyperparameter	**Value**
Hidden layers	3
Neurons per layer	100
Epochs	30
Optimizer	Adam

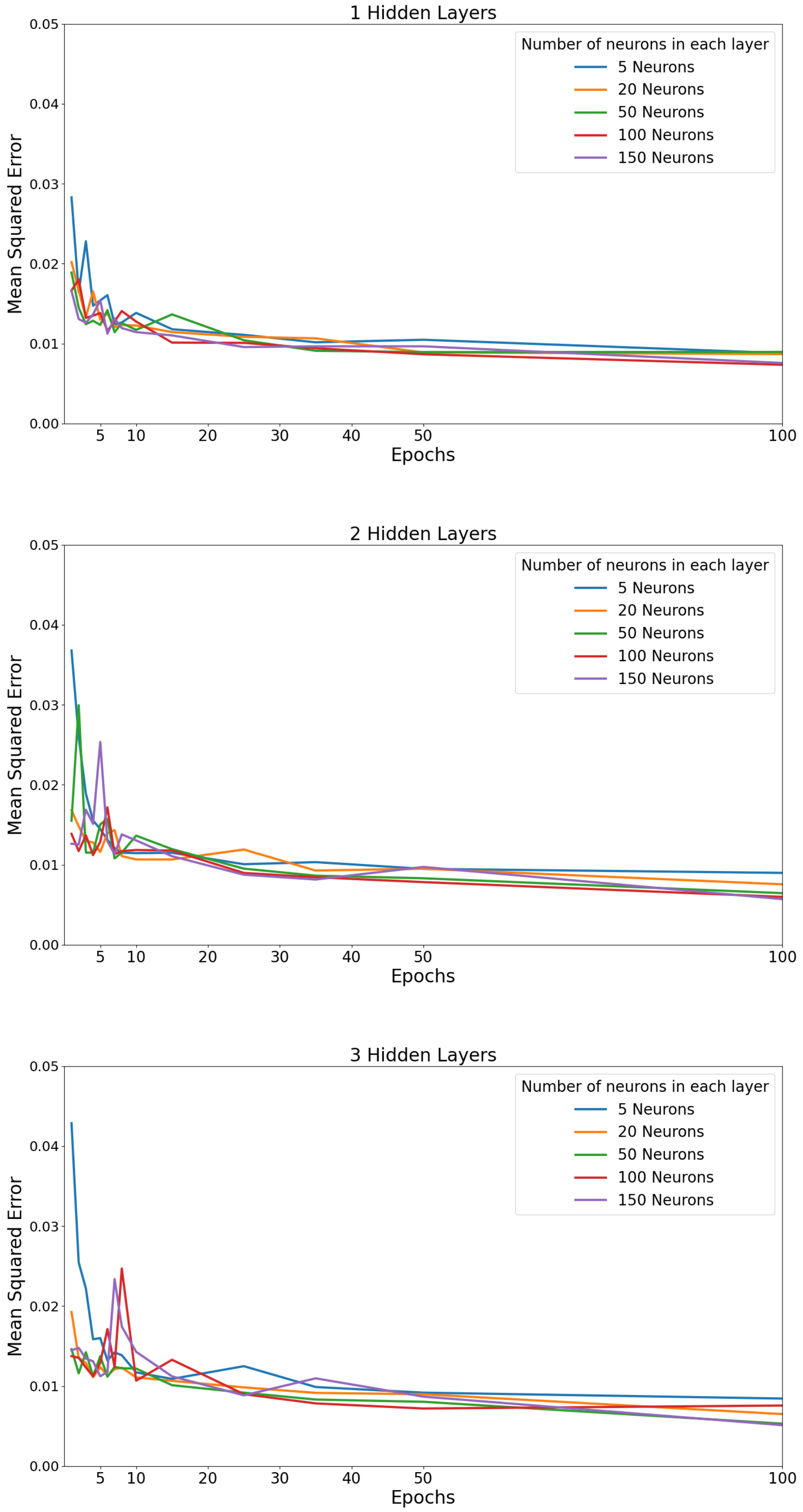

Figure 12: The loss function plotted for the LSTM model using different number of neurons, layers and epoch.

3.2.4 Random Forest

As with any ML model tuning hyperparameters is a critical aspect in optimising the performance of any RF model (Probst, 2019). Failure to sufficiently tune hyperparamaters may result in the model over- or underfitting the data, resulting in a model unable to generalise and make accurate predictions. The following hyperparamaters have been considered for the RF model in this thesis:

- The *max depth* considers the maximum total depth or number of steps in each individual tree. The *max depth* is an important hyperparameter as a too long tree risks overfitting the model while a short tree might lack the necessary depth to capture complex patterns or non-linearity within the data.
- The *number of estimators* indicates the total number of regression trees constructed in the RF. A large number of estimators generally increases the accuracy and performance of the model, but increases computational cost.
- The *max number of features* indicates the number of features considered when splitting a node in each individual decision-tree. A high number of features has the benefit of improving the performance of the model, yet a too high number of features decreases the diversity of individual trees within the RF. It is common to either select the square-root or binary logarithm of the total number of features as the max number of feature for the model.

To optimally select the hyperparamaters for the model a grid-search was performed for a variety of hyperparamaters illustrated in Figure 13. This thesis considers the following hyperparamaters for the RF model:

Hyperparameter	**Value**
Max depth	10
Number of estimators	200
Max features	sqrt

3.3 Software

The models used in this thesis were written in Python 3.9 using the following open-source software libraries: Keras 3.0, TensorFlow 2.13, statsmodels 0.14.1 and sklearn 1.3. Illustrative and schematic figures have all been designed in Affinity Designer.

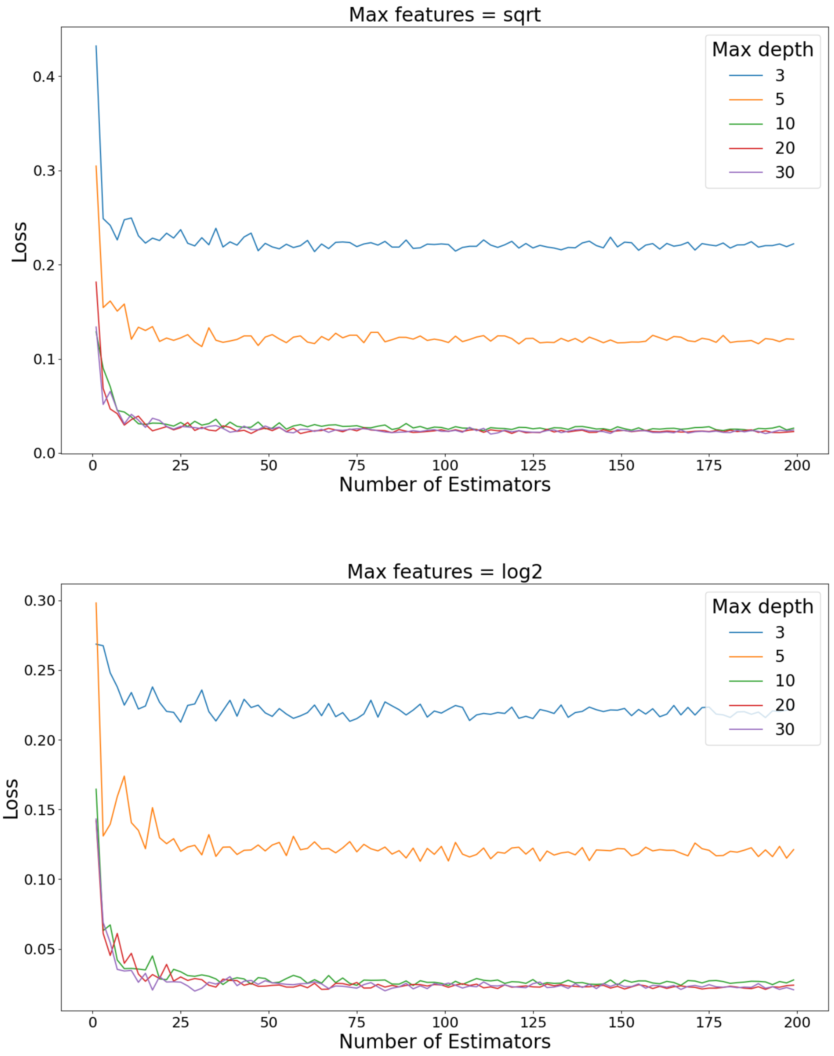

Figure 13: The loss function plotted depending on the number of estimators for the RF model using different hyperparametares.

4 Empirical results

This section considers the results of the four forecasting models: AR, VAR, RF and LSTM. The results considers the forecasted period between November 2007 and the end of October 2008, evaluating the accuracy of the models when forecasting 3, 6 and 12 months ahead using the metrics discussed in Section 2.5. The results for the four models are given in Table 3, 4 and 5. For graphical illustration of the forecasted results of the AR, VAR, LSTM and RF models see Appendix A, B, C and D.

Forecasting 3-months ahead

	RMSE			
	Autoregressive models		ML models	
Period	AR	VAR	LSTM	RF
Nov 2007 - Jan 2008	131.7	259.9	73.2	80.5
Feb 2008 - Apr 2008	108.9	58.0	90.9	68.7
May 2008 - Jul 2008	100.2	122.7	46.5	118.3
Aug 2008 - Okt 2008	121.6	129.2	282.5	104.9
Average	115.6	142.4	123.7	93.1

	MAE			
	Autoregressive models		ML models	
Period	AR	VAR	LSTM	RF
Nov 2007 - Jan 2008	114.7	243.4	57.6	77.7
Feb 2008 - Apr 2008	101.7	52.2	89.4	67.3
May 2008 - Jul 2008	86.9	111.6	41.2	89.8
Aug 2008 - Okt 2008	91.1	101.4	253.8	91.1
Average	98.6	127.2	110.5	81.5

Table 3: Overview of the results from the four models when forecasting three months ahead. The results are given as the RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) between the forecasted value and the observed value for the OMXS30.

Forecasting 6-months ahead

	RMSE			
	Autoregressive models		ML models	
Period	AR	VAR	LSTM	RF
Nov 2007 - Apr 2008	149.5	335.7	105.9	78.2
Maj 2008 - Okt 2008	178.2	201.8	98.7	123.1
Average	163.8	268.7	102.3	100.6

	MAE			
	Autoregressive models		ML models	
Period	AR	VAR	LSTM	RF
Nov 2007 - Apr 2008	139.3	319.1	97.1	59.8
May 2008 - Okt 2008	150.7	172.2	82.3	101
Average	145.0	245.6	89.7	80.4

Table 4: Overview of the results from the four models when forecasting six months ahead. The results are given as the RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) between the forecasted value and the observed value for the OMXS30.

Forecasting 1-year ahead

	RMSE			
	Autoregressive models		ML models	
Period	AR	VAR	LSTM	RF
Nov 2007 - Okt 2008	222.6	509.7	78.3	104.8

	MAE			
	Autoregressive models		ML models	
Period	AR	VAR	LSTM	RF
Nov 2007 - Okt 2008	199.1	472.3	57.3	86.4

Table 5: Overview of the results from the four models when forecasting twelve months ahead. The results are given as the RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) between the forecasted value and the observed value for the OMXS30.

From the results indicating the error of each model when forecasting 6 and 12 months ahead (see Table 4 & 5) the ML models exhibit the highest degree of accuracy with both the LSTM and RF model outperforming the autoregressive models. The results also indicate that the VAR model perform most poorly in relation to the other evaluated models for all studied periods and forecast horizons. This may be explained by poor predictive power of the independent variables selected for the VAR model.

From the results when forecasting 3 months ahead summarised in Table 3 the RF model perform on average the best. Both the AR and VAR model forecasting

3 months ahead during the initial period of the recession performs poorly, while forecasts performed for later periods exhibit far lower error. Both the RMSE and MAE indicate that both the ML and autoregressive models when forecasting 3 months ahead shows about similar results. Indicating small differences between models when forecasting short periods ahead, unlike when forecasting 6 or 12 months ahead where results indicate better performance from ML models.

5 Discussion and concluding remarks

This thesis was written with the intention of comparing the forecasting accuracy of common statistical models and more complex ML models on the Swedish stock market during periods of recession. From the results given in Section 4 the ML models clearly outperforms both autoregressive models when forecasting 6 or 12 months ahead. Indicating that both ML models (LSTM and RF) are better at long term forecasting on non-linear and sporadic data such as financial markets, something which is commonly argued for in the literature. The ML models seems notably better at capturing and predicting future sudden and unexpected shocks in stock market price which acts contrary to historical trends. This is somewhat expected as autoregressive models makes predictions on future values based on historical data, yet the increased accuracy of the ML models indicates how ML may more accurately predict future recessions.

The accuracy of the autoregressive models increases when forecasting three months ahead during later parts of the studied period. As the historical data includes recent downwards trend in the training data it is expected that these forecasts concerned with forecasting later periods should result in lower error. Never the less, the RF model performs best when forecasting three months ahead.

As both the VAR and RF uses independent variables to forecast future values within the time series the accuracy of these models are heavily dependent on the selection of independent variables. It could be argued that a larger number of independent variables would increase forecasting accuracy. Yet, it is difficult to find relevant variables which are reported at a monthly frequency while including historical data stretching back to January 1990. It should also be mentioned that variables which could explain changes on the Swedish stock market previously may not necessarily explain future changes. A deeper discussion regarding relevant economic indicators and changes on the stock market is interesting but outside the scope of this thesis.

It should also be mentioned that this thesis only considers two ML models and two common autoregressive models. There are a number of other common statistic models (SARIMA, ARIMA, MA etc) as well as other ML models (XGB, SVM etc) and variations on the LSTM and RF model which would have yielded different results. When tuning hyperparameters for the ML models only a small subset of the most relevant and commonly tuned hyperparameters were considered. This was chiefly due to time constrain but equally the large number of available hyperparameters necessitate limitation.

Although the results indicate that the RF model in general outperform the LSTM model, the results and scope of the study are insufficient to indicate which ML model is preferred above the other. Hyperparamater tuning along with a different or larger selection of independent variables would undoubtedly impact the forecast results and the outcome of the thesis would change. Therefore this thesis does not claim to make assertions regarding any specific model. Instead the results of the thesis gives useful insight into the ongoing discussion regarding model selection when forecasting on financial data.

Forecasting future values on any financial market is inherently difficult and pose challenging questions regarding model selection, hyperparamaters and modeling of complex and non-linear time series data. In relation to the OMXS30 we have ample reason to question the premise of the efficient market hypothesis as well as Burton Malkiels comparison of the likelihood of predicting future movement on the stock market with that of a coin toss. In relation to the question posed at the beginning of the thesis, the results indicates that ML models seems to more accurately predict future changes on the Swedish stock market than conventional autoregressive models during periods of recession. Giving credence to previous researchers assertion that ML are more suited for forecasting on complex and non-linear data such as financial data.

6 Future Research

- As mentioned in the previous section this thesis considers only a small subset of total hyperparameters available for tuning the ML models. Future research could consider diving deeper into optimal hyperparameter tuning for ML models when forecasting on financial data.

- While this thesis only employ a handful of independent variables used for the multivariate models. Future papers could employ far more independent variables and evaluate what economic indicators best correlate with changes on the Swedish financial market and thus evaluate which variables to consider when forecasting using multivariate ML models.

- As this thesis only considers the financial crises of 2008 for evaluating forecasting accuracy of different models, it would be interesting to see if other studies would result in the same result when evaluating different periods of recession.

References

Anari, A., & Kolari, J. (2014). Stock price and inflation. *Journal of financial research*, *24*, 587-602.

Andréasson, D., & Blomquist, J. M. (2020). *Forecasting the omxs30 – a comparison between arima and lstm* (Bachelor's Thesis). Uppsala University.

Asprem, M. (1989). Stock prices, assets portfolios and macroeconomic varaiables in ten european countries. *Journal of Banking and Finance*, *13*, 589-612.

Bhattacharjee, I., & Bhattacharja, P. (2019). Stock price prediction: A comparative study between traditional statistical approach and machine learning approach. In *4th international conference on electrical information and communication technology (eict).*

Blanchard, O., & Johnson, D. R. (2013). *Macro economics* (6th ed.). Pearson.

Celebi, K., & Hönig, M. (2019). The impact of macroeconomic factors on the german stock market: Evidence for the crisis, pre- and post-crisis periods. *International Journal of Financial Studies.*

Chakrabarti, A., & Ghosh, J. K. (2011). Aic, bic and recent advances in model selection. In P. S. Bandyopadhyay & M. R. Forster (Eds.), *Philosophy of statistics* (Vol. 7, p. 583-605). Amsterdam: North-Holland. doi: https://doi.org/10.1016/B978-0-444-51862-0.50018-6

Chen, Y., & Hao, Y. (2018). Integrating principle component analysis and weighted support vector machine for stock trading signals prediction. *Neurocomputing*, *321*, 381-402.

Dahlquist, J., & Knight, R. (2022). *Principles of finance.* OpenStax.

Degiannakis, S., Filis, G., & Arora, V. (2018). Oil prices and stock markets: A review of the theory and empirical evidence. *The Energy Journal*, *39*.

Federal Reserve Bank of St Louis. (2023). *Crude Oil Prices: Brent - Europe.* (`https://fred.stlouisfed.org/series/DCOILBRENTEU` (Accessed: 19/10/2023))

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Vol. 2). Springer.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (Vol. 1, p. 278-282 vol.1). doi: 10.1109/ICDAR.1995.598994

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic opimization. In *Proceedings of the 3rd international conference on learning representations (iclr 2015).*

Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert System With Applications*, *197*.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *Plos One*.

Malkiel, B. G. (2016). *A random walk down wall street* (11th ed.). W. W. Norton Company.

Mandic, D. P., & Chambers, J. A. (2001). *Recurrent neural networks for prediction.* John Wiley Sons Ltd.

Merkle, C., & Weber, M. (2014). Do investors put their money where their mouth is? stock market expectations and investing behavior. *Journal of Banking and Finance*, *46*, 372–386.

Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2008). *Introduction to time series analysis and forecasting.* John Wiley Sons Inc Publication.

NASDAQ OMX NORDIQ. (2023). *OMXS30 Stockholm INDEX.* (`https://www.nasdaqomxnordic.com/index/historiska_kurser?Instrument=SE0000337842` (Accessed: 10/10/2023))

Park, J., & Ratti, R. A. (2008). Oil price shock and stock markets in the us and 13 european countries. *Energy Economics*, *30*, 2587-2608.

Peixeiro, M. (2022). *Time series forecasting in python.* Manning Publications Co.

Pilerot, L., & Hellman, D. W. (2017). *Comparing forecast of armas and anns on omxs30: Examing form a economic and statistical point of view* (Master's Thesis). University of Gothenburg.

Probst, P. (2019). *Hyperparameters, tuning and meta-learning for random forest and other machine learning algorithms* (Unpublished doctoral dissertation).

Sammut, C., & Webb, G. I. (2010). *Encyclopedia of machine learning.* Springer New York, Ny.

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D*, *404*.

Shumway, R. H., & Stoffer, D. S. (2017). *Time series analysis and its application* (4th ed.). Springer.

Sonkavde, G., Dharrao, D. S., Bongale, A. M., deokate, S. T., Doreswamy, D., & Bhat, S. K. (2023). Forecasting stock market prices using machine learning and deep learning models: A systematic review, performance analysis and discussion of implications. *Internatinal Journal of Financial Studies*.

Statistiska Centralbyrån. (2023a). *Konsumentprisindex (KPI).* (`https://www.scb.se/hitta-statistik/statistik-efter-amne/priser-och-konsumtion/konsumentprisindex/konsumentprisindex-kpi/` (Accessed: 18/10/2023))

Statistiska Centralbyrån. (2023b). *Utrikeshandel med varor.* (`https://www.scb.se/hitta-statistik/statistik-efter-amne/handel-med-varor-och-tjanster/utrikeshandel/utrikeshandel-med-varor/#_Tabellerochdiagram` (Accessed: 18/10/2023))

Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2019). Stock closing price predictions using machine learning techniques. In *International conference on computational intelligence and data science (iccids 2019).*

Appendix A: AR results

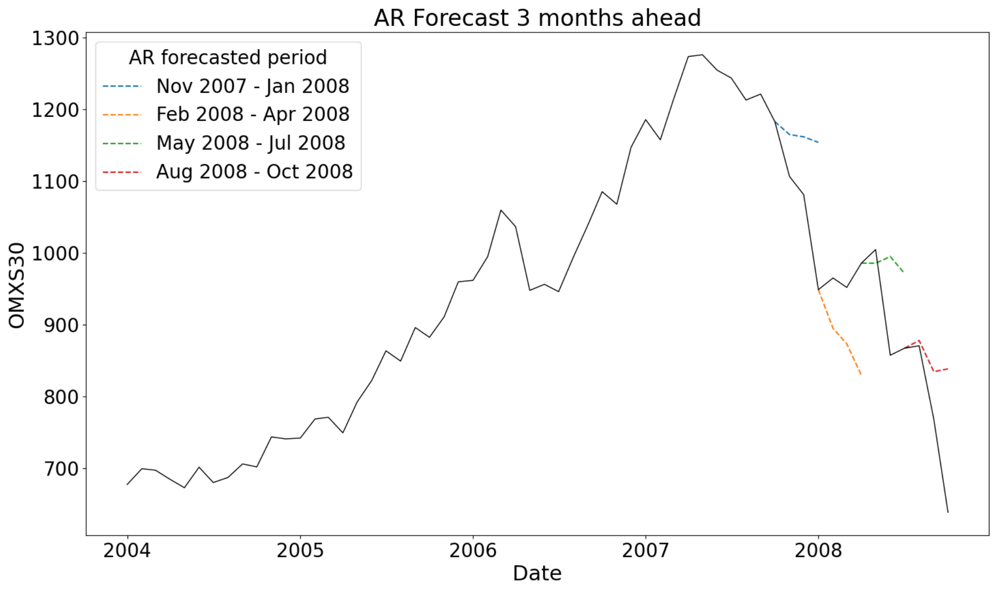

Figure 14: Results of the AR model forecasting 3-months ahead during the period Nov 2007- Oct 2008.

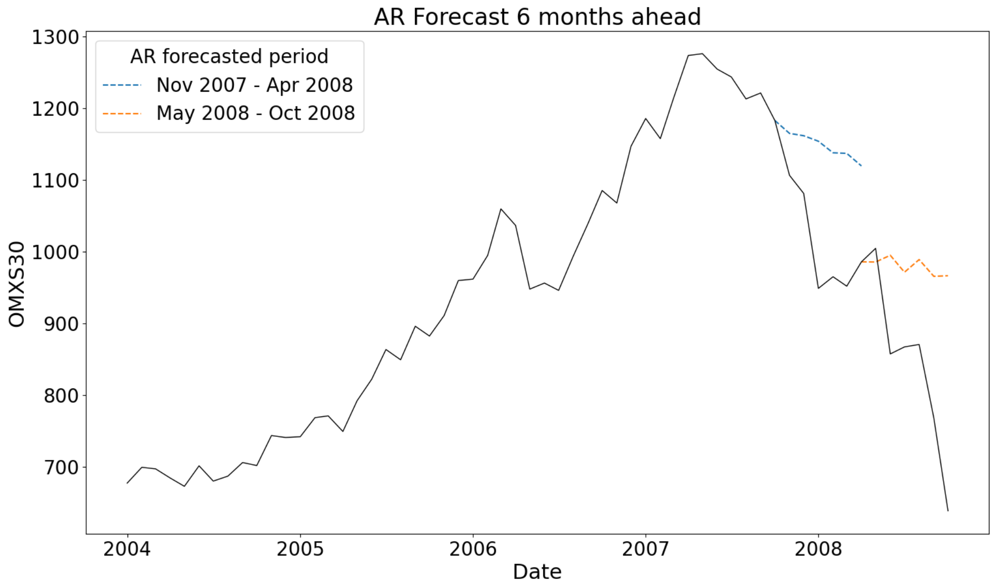

Figure 15: Results of the AR model forecasting 6-months ahead during the period Nov 2007- Oct 2008.

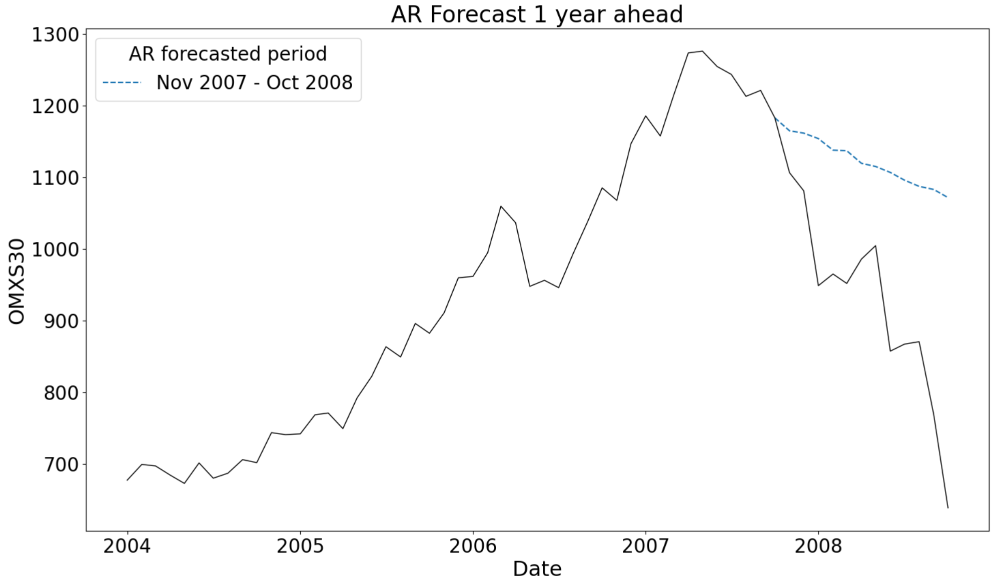

Figure 16: Results of the AR model forecasting 1-year ahead during the period Nov 2007-Oct 2008.

Appendix B: VAR results

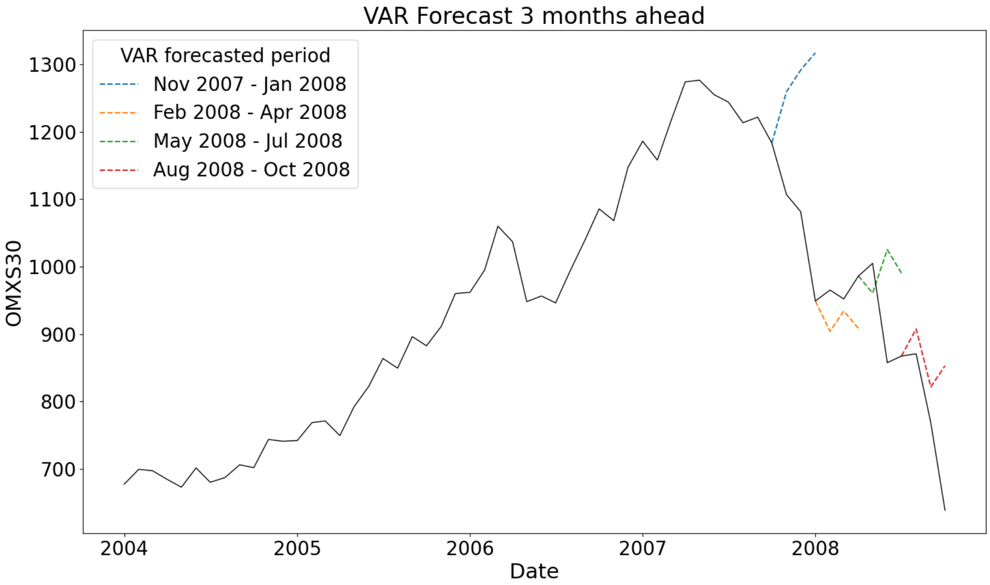

Figure 17: Results of the VAR model forecasting 3-months ahead during the period Nov 2007- Oct 2008.

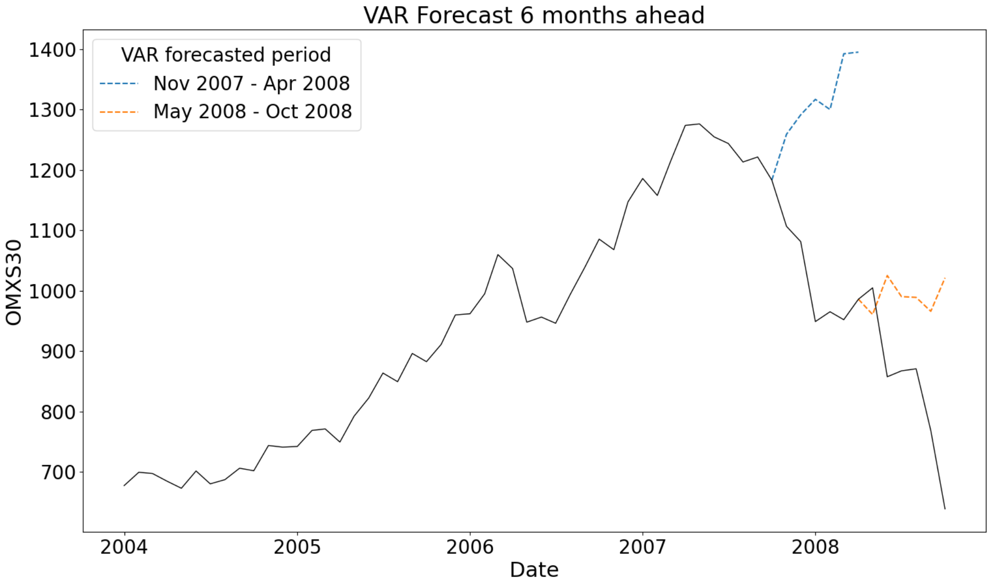

Figure 18: Results of the VAR model forecasting 6-months ahead during the period Nov 2007- Oct 2008.

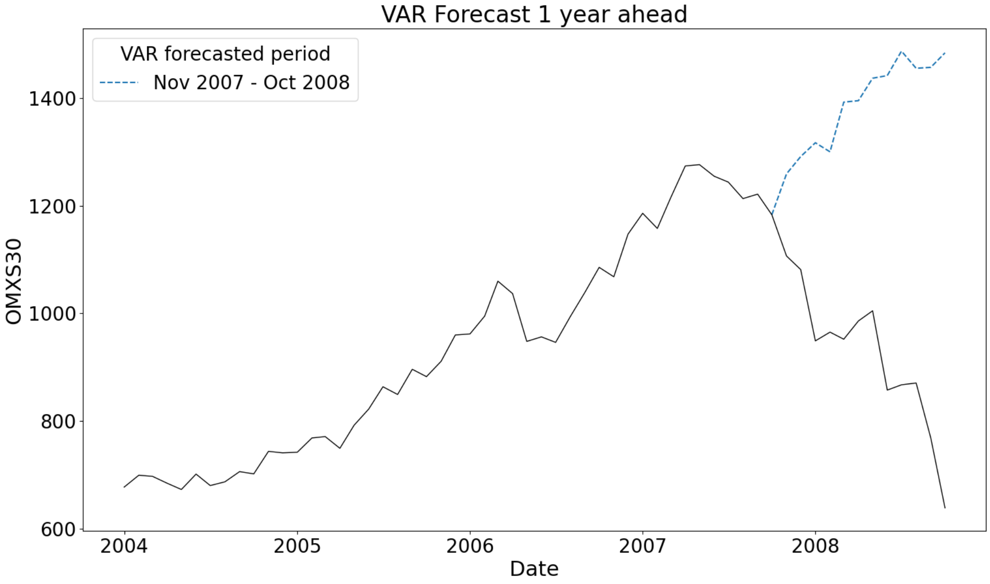

Figure 19: Results of the VAR model forecasting 1-year ahead during the period Nov 2007- Oct 2008.

Appendix C: LSTM results

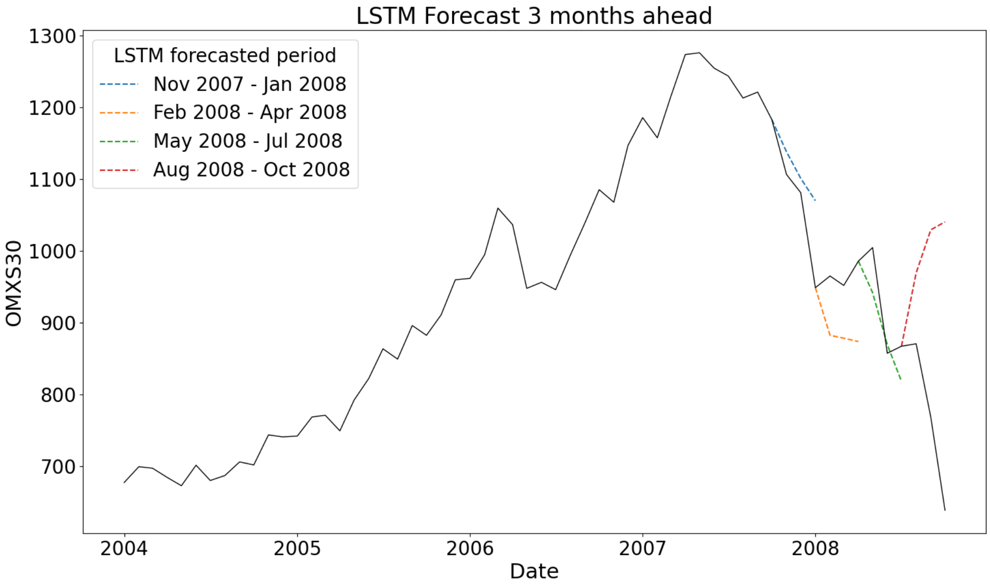

Figure 20: Results of the LSTM model forecasting 3-months ahead during the period Nov 2007- Oct 2008.

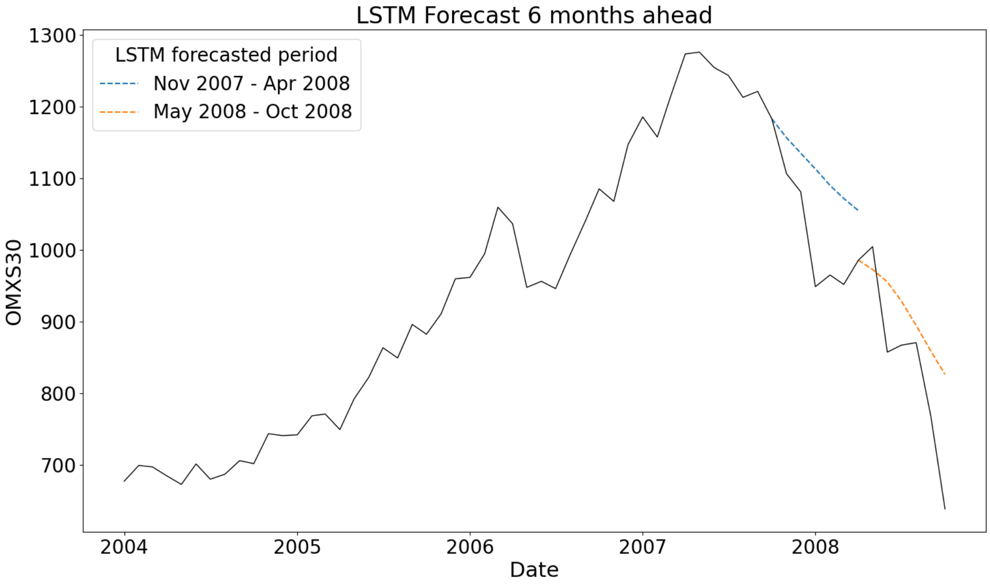

Figure 21: Results of the LSTM model forecasting 6-months ahead during the period Nov 2007- Oct 2008.

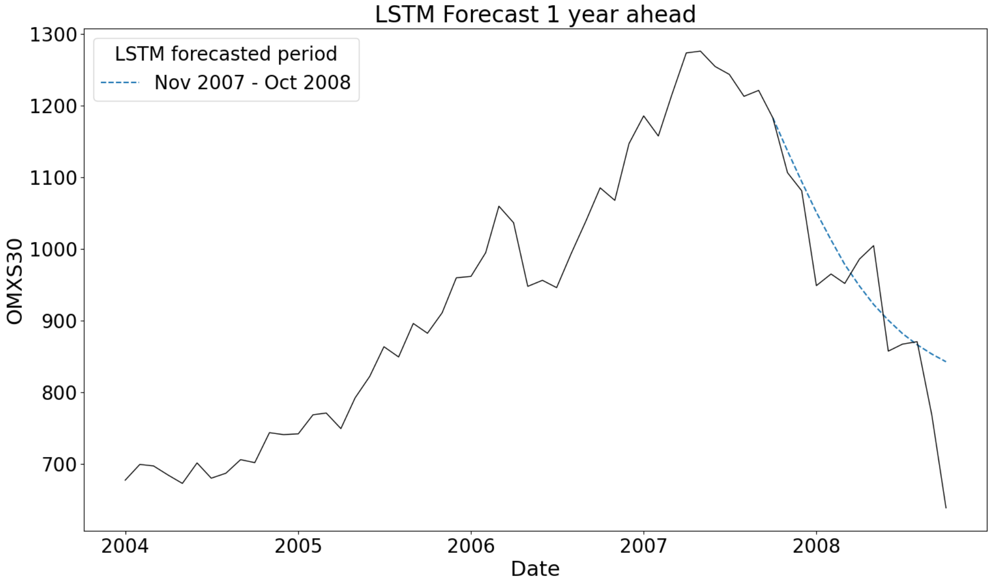

Figure 22: Results of the LSTM model forecasting 1-year ahead during the period Nov 2007- Oct 2008.

Appendix D: RF results

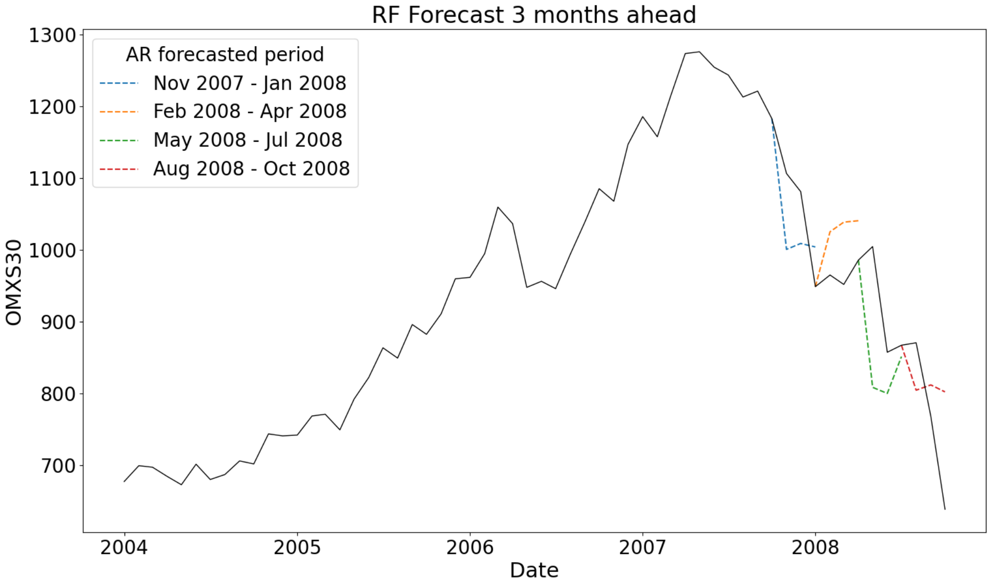

Figure 23: Results of the RF model forecasting 3-months ahead during the period Nov 2007- Oct 2008.

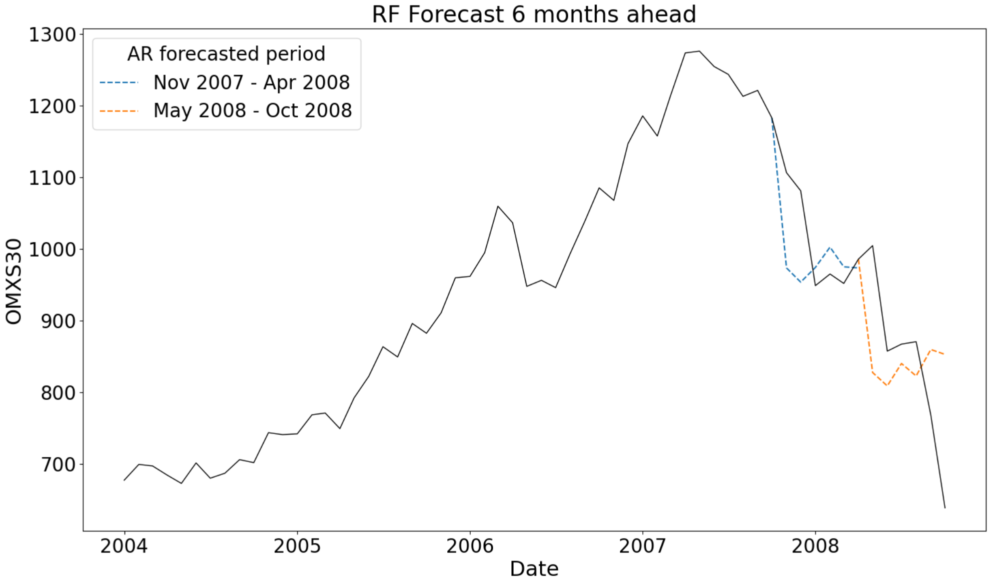

Figure 24: Results of the RF model forecasting 6-months ahead during the period Nov 2007- Oct 2008.

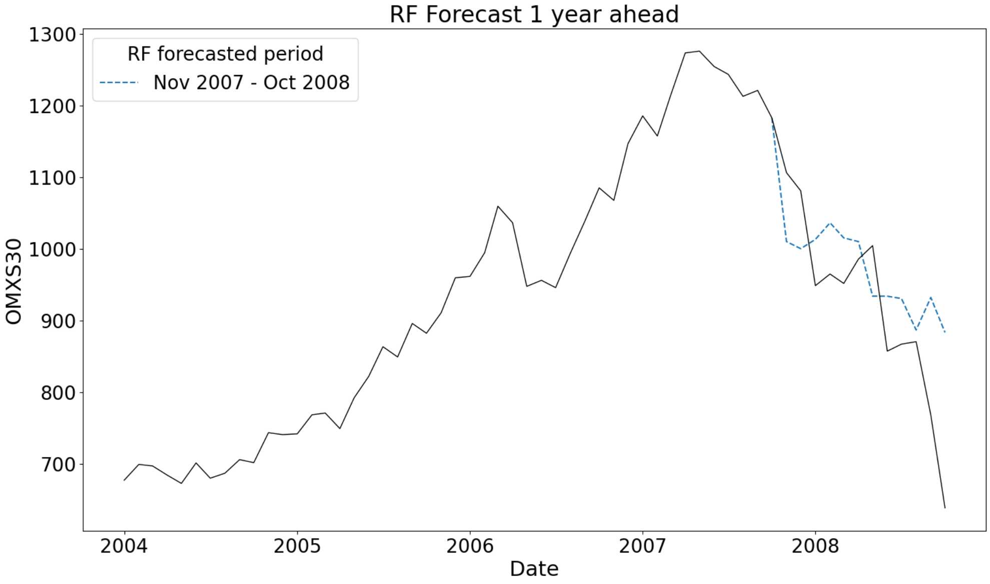

Figure 25: Results of the RF model forecasting 1-year ahead during the period Nov 2007-Oct 2008.