LUT University
School of Business and Management
Master's Programme in Strategic Finance and Analytics

Markus Kauppinen

Economic Growth Forecasting in Nordic Countries: A Comparative Analysis Between Generalized Autoregressive Conditional Heteroscedasticity Models and Nonlinear Autoregressive Neural Network Models

Master's Thesis

1st examiner: Associate professor Sheraz Ahmed
2nd examiner: Associate professor Jan Stoklasa

ABSTRACT

Author:	Markus Kauppinen
Title:	Economic Growth Forecasting in Nordic Countries: A Comparative Analysis Between Generalized Autoregressive Conditional Heteroscedasticity Models and Nonlinear Autoregressive Neural Network Models
Faculty:	School of Business and Management
Degree:	Master of Science in Economics and Business Administration
Master's Programme:	Strategic Finance and Analytics
Year:	2021
Master's Thesis:	LUT University, 117 pages, 33 figures, 48 tables, 8 appendices
Examiners:	Associate professor Sheraz Ahmed
	Associate professor Jan Stoklasa
Keywords:	NAR, ARMA-GARCH, ANN, GDP, economic growth forecasting, comparative analysis

Advanced computational efficiency has made possible to utilize artificial neural networks (ANN) in economic forecasting. This thesis studies differences between hybrid autoregressive moving average with generalized autoregressive conditional heteroscedastic (ARMA-GARCH) -models and nonlinear autoregressive neural network (NAR) - models in predicting quarterly real growth rate of GDP in five Nordic countries. Models are fitted to the in-the-sample subsample derived from the original time series data sample and values for next 20 timesteps are predicted and compared with the actual out-of-sample values derived from original time series data. Results indicate that there are slight differences in predicting capability among the models. In general, as a whole NAR -models could not outperform ARMA-GARCH -models. In case of Iceland NAR -models produced more accurate forecasts than ARMA-GARCH models and in case of Denmark NAR -models did perform better than the majority of ARMA-GARCH -models. For Norway and Sweden, the results are mixed; neither NAR -models nor ARMA-GARCH -models outperformed each other clearly, but NAR -models performed slightly worse in general. For Finland the results could not be derived due to characteristics of the time series data.

TIIVISTELMÄ

Tekijä:	Markus Kauppinen
Aihe:	Talouskasvun ennustaminen Pohjoismaissa: vertaileva analyysi yleistettyjen autoregressiivisien ehdollisesti heteroskedastisien mallien ja epälineaarisien autoregressiivisten neuroverkkomallien välillä
Tiedekunta:	Kauppatieteiden tiedekunta
Tutkinto:	Kauppatieteiden maisteri (KTM)
Koulutusohjelma:	Strategic Finance and Analytics
Vuosi:	2021
Pro Gradu -tutkielma:	LUT yliopisto 117 sivua, 33 kuvaa, 48 taulukkoa, 8 liitettä
Tarkastajat:	Apulaisprofessori Sheraz Ahmed Apulaisprofessori Jan Stoklasa
Asiasanat**:**	NAR, ARMA-GARCH, ANN, GDP, talouskasvun ennustaminen, vertaileva analyysi

Kehittynyt laskentateho on mahdollistanut keinotekoisten neuroverkkojen hyödyntämisen talouskasvun ennustamisessa. Tämä Pro Gradu -tutkielma tutkii eroavaisuuksia yhdistettyjen autoregressiivisten liukuvan keskiarvon ja yleistettyjen autoregressiivisien ehdollisesti heteroskedastisien -mallien (ARMA-GARCH), sekä epälineaarisien autoregressiivisten neuroverkkomallien (NAR) välillä kvartaalittaisen aidon talouskasvun ennustamisessa Pohjoismaissa. Mallit sovitetaan sisäiseen dataotokseen, joka on johdettu alkuperäisestä aikasarjadatasta, jonka jälkeen ennustetaan arvot 20:lle seuraavalle aika-arvolle. Ennustettuja arvoja verrataan alkuperäisestä aikasarjadatasta muodostetun vertailudatan kanssa. Tutkimuksen tulokset osoittavat, että mallien välillä on hienoja eroavaisuuksia ennustuskyvykkyydessä. Yleisesti kokonaisuudessaan NAR -mallit eivät suoriutuneet ARMA-GARCH -malleja paremmin. Islannin osalta NAR –mallit tuottivat tarkemmat ennusteet kuin ARMA-GARCH -mallit ja Tanskan osalta NAR -mallit suoriutuivat paremmin kuin suurin osa ARMA-GARCH -malleista. Norjan ja Ruotsin kohdalla tulokset ovat hajoavia; niin ikään NAR -mallit, kuin ARMA-GARCH mallitkaan eivät suoriutuneet selkeästi toisiaan paremmin, mutta NAR -mallit suoriutuivat yleisesti hieman huonommin. Suomen osalta tuloksia ei voitu määrittää johtuen aikasarjadatan ominaisuuksista.

ACKNOWLEDGEMENTS

I would like to thank Associate professor Sheraz Ahmed for guidance throughout the process of writing this thesis and valuable teaching lessons during the studies. I also want to thank Associate professor Jan Stoklasa for giving advice on modelling techniques and principles used on conducting the empirical analysis of this thesis.

Special thanks to fellow master students for making the time spent in Lappeenranta truly unforgettable and with who I got to experience the joys of student life once more. Those memories last for a lifetime.

I'm very grateful for Simo showing support during this demanding process and cheering for me to finish my studies. Your support and care mean a lot to me.

Helsinki 19th of June2021

Markus Kauppinen

TABLE OF CONTENTS

1 INTRODUCTION 12
1.1 Motivation and research methodology of the study 13
1.2 Objectives and research questions 15
1.3 Structure of the thesis 15
2 THEORETICAL FRAMEWORK – A LITERATURE REVIEW 16
2.1 Time series 16
2.2 Time series modelling and forecasting 17
2.3 Neural Networks 18
2.4 ANNs and ARMA-GARCH -models in GDP growth forecasting – a review of previous research 21
3 TIME SERIES FORECASTING MODELS 24
3.1 Conditional mean models 24
3.1.1 AR model 24
3.1.2 MA model 25
3.1.3 ARMA model 26
3.2 Conditional volatility models 27
3.2.1 GARCH model 27
3.2.2 EGARCH model 28
3.2.3 GJR-GARCH model 29
3.3 Nonlinear Autoregressive Neural Network Model 29
3.3.1 Levenberg-Marquardt algorithm 30
3.3.2 Bayesian Regularization algorithm 32
3.3.3 Scaled Conjugate Gradient algorithm 33
3.4 Tests for stationarity 34
3.4.1 Augmented Dickey-Fuller test 34
3.4.2 KPSS test 34
3.4.3 Variance ratio test 35
3.5 Model fitness tests 35
3.5.1 Ljung-Box test for autocorrelation 35
3.5.2 Engle's ARCH test 36
3.5.3 Jarque-Bera test for normality 37
3.6 Loss functions MAE and MSE 37
4 DATA AND METHODOLOGY 39
4.1 Descriptive statistics of Quarterly Growth Rate of real GDP, Denmark 40
4.2 Descriptive statistics of Quarterly Growth Rate of real GDP, Finland 45
4.3 Descriptive statistics of Quarterly Growth Rate of real GDP, Iceland 50

4.4 Descriptive statistics of Quarterly Growth Rate of real GDP, Norway 55
4.5 Descriptive statistics of Quarterly Growth Rate of real GDP, Sweden 61
4.6 Forecasting procedure 65
5 EMPIRICAL RESULTS AND ANALYSIS 67
5.1 Results for Quarterly Growth Rate of real GDP, Denmark 68
5.1.1 Results of conditional mean and variance model estimation 68
5.1.2 Result of NAR architecture estimation 69
5.1.3 Results of model performance and forecasting 71
5.2 Results for Quarterly Growth Rate of real GDP, Finland 72
5.2.1 Results of conditional mean and variance model estimation 72
5.2.2 Result of NAR architecture estimation 74
5.2.3 Results of model performance and forecasting 76
5.3 Results for Quarterly Growth Rate of real GDP, Iceland 77
5.3.1 Results of conditional mean and variance model estimation 77
5.3.2 Result of NAR architecture estimation 79
5.3.3 Results of model performance and forecasting 80
5.4 Results for Quarterly Growth Rate of real GDP, Norway 81
5.4.1 Results of conditional mean and variance model estimation 81
5.4.2 Result of NAR architecture estimation 83
5.4.3 Results of model performance and forecasting 85
5.5 Results for Quarterly Growth Rate of real GDP, Sweden 85
5.5.1 Results of conditional mean and variance model estimation 85
5.5.2 Result of NAR architecture estimation 87
5.5.3 Results of model performance and forecasting 88
5.6 Overall ranking results of the models 89
6 CONCLUSIONS 91
6.1 Discussion 91
6.2 The limitations of the study 94
6.3 Suggestions for further research 95
REFERENCES 96
APPENDICES 101
Appendix 1 AIC matrices 101
Appendix 2 Levenberg-Marquardt performance matrices 104
Appendix 3 Scaled conjugate gradient performance matrices 107
Appendix 3 Bayesian regularization performance matrix 110
Appendix 4 Architecture or open and closed loop NARs, Denmark 113
Appendix 5 Architecture of open and closed loop NARs, Finland 114
Appendix 6 Architecture of open and closed loop NARs, Iceland 115

Appendix 7 Architecture of open and closed loop NARs, Norway..........................116
Appendix 8 Architecture of open and closed loop NARs, Sweden.........................117

LIST OF FIGURES

Figure 1. Structure of the thesis 16
Figure 2. Architecture of ANN 19
Figure 3. Architecture of NAR 20
Figure 4. Time series of DEN_growth_rate 41
Figure 5. ACF -plot of DEN_growth_rate 42
Figure 6. Histogram of DEN_growth_rate 43
Figure 7. QQ -plot of DEN_growth_rate 43
Figure 8. Time series of FIN_growth_rate 46
Figure 9. ACF -plot of FIN_growth_rate 47
Figure 10. Histogram of FIN_growth_rate 48
Figure 11. QQ -plot of FIN_growth_rate 48
Figure 12. Time series of ISL_growth_rate 51
Figure 13. ACF -plot of ISL-Growth_rate 52
Figure 14. Histogram of ISL_growth_rate 53
Figure 15. QQ -plot of ISL_growth_rate 53
Figure 16. Time series of NOR_growth_rate 56
Figure 17. ACF -plot of NOR_growth_rate 57
Figure 18. Histogram of NOR_growth_rate 58
Figure 19. QQ -plot of NOR_growth_rate 58
Figure 20. Time series of SWE_growth_rate 61
Figure 21. ACF -plot of SWE_growth_rate 62
Figure 22. Histogram of SWE_growth_rate 63
Figure 23. QQ -plot of SWE_growth_rate 63
Figure 24. ACF and PACF -plots of fitted ARMA(8,8) -model for DEN_growth_rate 68
Figure 25. MSE of defined delay parameters for NARs; DEN_growth_rate 70
Figure 26. ACF and PACF -plots of fitted ARMA(5,8) -model for FIN_growth_rate 73
Figure 27. MSE of defined delay parameters for NARs; FIN_growth_rate 75
Figure 28. ACF and PACF -plots of fitted ARMA(7,7) -model for ISL_growth_rate 77
Figure 29. MSE of defined delay parameters for NARs, ISL_growth_rate 79
Figure 30. ACF and PACF -plots of fitted ARMA(7,7) -model for NOR_growth_rate 82
Figure 31. MSE of defined delay parameters for NARs, NOR_growth_rate 84
Figure 32. ACF and PACF -plots of fitted ARMA(6,2) -model for SWE_growth_rate 86
Figure 33. MSE of defined delay parameters for NARs, SWE_growth_rate 88

LIST OF TABLES

Table 1. Information of the time series 40
Table 2. Summary statistics of DEN_growth_rate 44
Table 3. Result of Jarque-Bera test for DEN_growth_rate 44
Table 4. Result of ADF test for DEN_growth_rate 45
Table 5. Result of KPSS test for DEN_growth_rate 45
Table 6. Result of Variance ratio test for DEN_growth_rate 45
Table 7. Summary statistic for FIN_growth_rate 49
Table 8. Result of Jarque-Bera test for FIN_growth_rate 49
Table 9.Result of ADF test for FIN_growth_rate....................50
Table 10. Result of KPSS test for FIN_growth_rate 50
Table 11. Result of Variance ratio test for FIN_growth_rate....................50
Table 12. Summary statistics of ISL_growth_rate 54
Table 13. Result of Jarque-Bera test for ISL_growth_rate 54
Table 14. Results of ADF test for ISL_growth_rate 55
Table 15. Result of KPSS test for ISL_growth_rate....................55
Table 16. Result of Variance ratio test for ISL_growth_rate 55
Table 17. Summary statistic for NOR_growth_rate 59
Table 18. Result of Jarque-Bera test for NOR_growth_rate....................59
Table 19. Result of ADF test for NOR_growth_rate 60
Table 20. Result of KPSS test for NOR_growth_rate 60
Table 21. Result of Variance ratio test for NOR_growth_rate. 60
Table 22. Summary statistics for SWE_growth_rate 64
Table 23. Result of Jarque-Bera test for SWE_growth_rate....................64
Table 24. Result of ADF test for SWE_growth_rate 65
Table 25. Result of KPSS test for SWE_growth_rate....................65
Table 26. Result of Variance ratio test for SWE_growth_rate 65
Table 27. Result of Ljung-Box test for ARMA(8,8) -model residuals 68
Table 28. Result of Engle's ARCH test for ARMA(8,8) -model residuals....................69
Table 29. Result of Jarque-Bera test for ARMA(8,8) -model residuals....................69
Table 30. Model performance ranking for DEN_growth_rate 72
Table 31. Result of Ljung-Box test for ARMA(5,8) -model residuals 73
Table 32. Result of Engle's ARCH test for ARMA(5,8) -model residuals....................74
Table 33. Result of Jarque-Bera test for ARMA(5,8) -model residuals....................74
Table 34. Model performance ranking for FIN_growth_rate 76

Table 35. Result of Ljung-Box test for ARMA(7,7) -model residuals 78
Table 36. Result of Engle’s ARCH test for ARMA(7,7) -model residuals 78
Table 37. Result of Jarque-Bera test for ARMA(7,7) -model residuals..............................78
Table 38. Model performance ranking for ISL_growth_rate 81
Table 39. Result of Ljung-Box test for ARMA(7,7) -model residuals 82
Table 40. Result of Engle's ARCH test for ARMA(7,7) -model residuals...........................83
Table 41. Result of Jarque-Bera test for ARMA(7,7) -model residuals..............................83
Table 42. Model performance ranking for NOR_growth_rate..85
Table 43. Result of Ljung-Box test for ARMA(6,2) -model residuals 86
Table 44. Result of Engle's ARCH test for ARMA(6,2) -model residuals...........................87
Table 45. Result of Jarque-Bera test for ARMA(6,2) -model residuals..............................87
Table 46. Model performance ranking for SWE_growth_rate..89
Table 47. Final ranking of the models according to MAE 90
Table 48. Final ranking of the models according to MSE 90

ABBREVIATIONS

ACF	Autocorrelation Function
ADF	Augmented Dickey-Fuller test
ANN	Artificial Neural Network
AR	Autoregressive
ARCH	Autoregressive Conditional Heteroscedastic
ARMA	Autoregressive Moving Average
ARMA-GARCH	Autoregressive Moving Average with Generalized Autoregressive Conditional Heteroscedastic
BR	Bayesian Regularization
EGARCH	Exponential Generalized Autoregressive Conditional heteroscedastic
GARCH	Generalized Autoregressive Conditional Heteroscedastic
GDP	Gross Domestic Product
GJR-GARCH	Glosten-Jagannathan-Runkle- Generalized Autoregressive Conditional Heteroscedastic
KPSS	Kwiatowski-Phillips-Schmidt-Shin test
LM	Levenberg-Marquardt
MA	Moving Average
MAE	Mean Absolute Error
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NAR	Nonlinear Autoregressive Neural Network
NN	Neural Network
OLS	Ordinary Least Squares
PACF	Partial Autocorrelation Function
RMSE	Root Mean Square Error
SCG	Scaled Conjugate Gradient
SVM	Support Vector Machine

1 INTRODUCTION

Recent financial crises across continents have risen concerns whether policymakers have all the necessary tools to maintain financial stability. Capability to predict fluctuations in an economy makes it possible for policymakers to take precautionary actions to minimize the effects of economic disturbances (Hall et. al. 2008). Artificial neural network (ANN) techniques have been increasingly used in recent years for a variety of applications where traditionally statistical methods have been employed. Due to advances in computational capabilities, utilization of these nonparametric models has become more easily applicable (Medeiros et. al. 2006). While problems such as economic forecasting, financial modelling and stock market prediction are being more and more exposed to machine learning algorithms and neural network techniques to solve for, traditional econometric approaches are still being used (Zukime 2004).

One of these traditional approaches is time series predicting, which allows to discover, with some margin of error, the future values of a series from its past values. These approaches have been successfully applicated in fields such as finance and economics and are based on mathematical linear models. In late 70’s Box and Jenkins made crucially important work in developing these linear approaches. However, practical experiments have shown that these approaches are not always suitable in predicting, since many time series seem to follow rather nonlinear behavior than linear behavior (Tealab 2018). Including neural network techniques in predicting has two main advantages. First, these techniques do not require making any previous assumptions of underlying distribution of the data, second; neural network prediction is especially useful in situations where inputs are missing or highly correlated, or in situations where the dependence of the systems is nonlinear (Zukime 2004).

Gross domestic product (GDP) is a measure of performance for an economy. GDP is the value of all goods and services produced in a country within a given certain time period, for example one year or one quarter. Forecasting economic performance is essential part of a country’s economic decision-making. Therefore, GDP data is widely used in the field of economic time series modelling and analysis (Elsayir 2018). Recently, research in GDP growth forecasting using ANNs has increased. Research done by Tkacz (2001), shows that neural network (NN) models produce statistically lower errors in forecasting yearly GDP growth rate than univariate time series and linear models. Zhang et. al. (2018) conducted a study of forecasting accuracy between of ARNN, ARIMA and GARCH models in predicting the Baltic dry index. They found out that ANN models produce more accurate forecast for long-term time horizon but not for short-term time horizon. Study by Samir (2016) in comparing ANN and time series models the author found out ANNs outperform time series and regression models in forecasting quarterly GDP for Palestine. According to these studies, there seems to be evidence of ANN models being able to predict GDP changes more accurately than traditional time series and linear models, which motivates to investigate the performance of nonlinear autoregressive neural networks compared to traditional time series modelling and predicting methods in economic growth forecasting.

Jahn (2018) performed a research in demonstrating the capabilities of ANN regression models in estimating time trends of GDP growth rates. The research included multiple West-European countries as well as Japan and United States. From Nordic countries, the research included Denmark, Finland and Sweden. As a result, it was found out that the prediction errors of ANN model were much lower than those of linear model. This gives evidence that ANN models could also produce better forecasts for GDP growth rate in some of the Nordic countries, but since not all of the Nordic countries were involved in the research, it cannot be stated that this is the case for all of the Nordic countries. Nordic countries have generally similar infrastructure and size of an economy than some European countries in Jahn’s (2018) research. Because ANNs produce better forecasts for a Middle East country Palestine according to Samir (2016) and for the United States and Japan (Jahn 2018), it is intriguing to see how ANNs perform when utilized in forecasting of contradictory countries’ economic growth. These are the main reasons why Nordic countries are chosen to be in the focus of this study.

According to these studies, there seems to be evidence of NN models being able to predict GDP changes more accurately than time series and linear models. Some of the studies are done for contradictory economies to Nordic countries and some for similar countries and economies. There is no evidence of this being the case for all of the Nordic countries but rather for some of them. This is the main justification and motivation to investigate the performance of nonlinear autoregressive neural networks compared to traditional time series modelling and predicting methods in forecasting quarterly growth rate of real GDP in the Nordic countries.

1.1 Motivation and research methodology of the study

Motivation for conducting this study arises from the authors interest towards time series forecasting and modelling and the desire to learn more about economic forecasting with

ANNs. ANN techniques are quite new in a field of economic research and there seem to be a quite large gap in research in this field for the authors knowledge. The fact that there are not lot of empirical research done on this matter in the Nordics was the main reason for doing this research. That also affected the chosen methodology of the study to be comparative analysis, since not lot research has been done on investigating the predictive capabilities on NAR -models in economic forecasting.

Challenging problem of improving forecasts usually arises from the failure to account for large autocorrelations, trend and seasonality in data leading to a lack of accuracy in forecasting. Time series models such as ARMA have been used in literature to take into account these kind of patterns in time series data (Abdullah & Tayfur 2004). ANNs are an effective way to simulate nonlinear patterns and they can find hidden patterns that are independent from any mathematical model (Baghirli 2015). ANNs are also used for nonlinear regression models and time series modelling and forecasting because of their universal approximation properties. (Benrhmach et. al. 2020). Therefore, both modelling techniques ANNs and ARMA-GARCH -models can be used to forecast univariate nonlinear time series. Capabilities to model and forecast nonlinear characteristics of economic growth data with these two different kinds of time series modelling families motivates the purpose of this thesis. Also, the fact that there are only few similar studies done, where GDP growth rate is modelled and forecasted as a univariate time series with different modelling techniques derives the motivation to conduct this study. The choice to focus on univariate modelling techniques were made by the fact that in general, the models such as ANNs, which use more information (external variables), do produce more accurate forecasts because of their dynamics. The key conceptual aim of this thesis is to find out are ANNs able to produce more accurate forecasts than ARMA-GARCH -models when univariate time series is considered.

The purpose of this thesis is to examine capabilities of ANNs and ARMA-GARCH models in modelling and forecasting univariate nonlinear time series data of GDP growth rate. This is done with comparative analysis in order to find out how the models perform in modelling and forecasting in relation to actual values of GDP growth rate. The goal is to give more insight of modelling and forecasting economic growth as univariate time series with different time series modelling techniques, especially with using ANNs.

1.2 Objectives and research questions

The goal of this research is to compare forecasting performance between hybrid autoregressive moving average with generalized autoregressive conditional heteroscedastic (ARMA-GARCH) models and nonlinear autoregressive neural network (NAR) models in predicting quarterly growth rate of real GDP in the Nordic countries. This done through comparative analysis with three objectives. First objective of the research is to explore is it possible to use hybrid ARMA-GARCH and NAR in predicting economic growth. The objective is reached by reviewing previous research related economic growth predicting using ARMA-GARCH and NAR models. Second objective is to find out are NAR -models more accurate in predicting economic growth than hybrid ARMA-GARCH -models. Second objective is reached by doing comparative empirical analysis on these models and interpreting the results in terms of loss functions. Third objective of this thesis is to figure out if there is difference in predicting capabilities among the models. This objective is also reached through empirical analysis.

Based on the objectives of the research, four different research questions were formed:

1. *Can ARMA-GARCH and NAR -models be used to predict quarterly growth rate of real GDP?*
2. *Are NARs able to generate more accurate forecasts for predicting quarterly growth rate of real GDP in Nordic countries than hybrid ARMA-GARCH models?*
3. *Is there difference in predicting capability among ARMA-GARCH -models?*
4. *Is there difference in predicting capability among NAR-models?*

Finding out answers to these research questions brings a lot of new insights into predicting quarterly growth rate of real GDP as a time series in Nordic countries since a lot of research have not been done on this field of economic growth forecasting.

1.3 Structure of the thesis

The structure of this thesis consists of three main parts. The first part presents the aim of the research, research questions and theoretical framework of the study. Second part of the thesis focuses on describing the data and presenting the results of empirical analysis. In the last part the findings of the empirical analysis are discussed and suggestions for further research are made. The structure of the thesis is illustrated in figure 1.

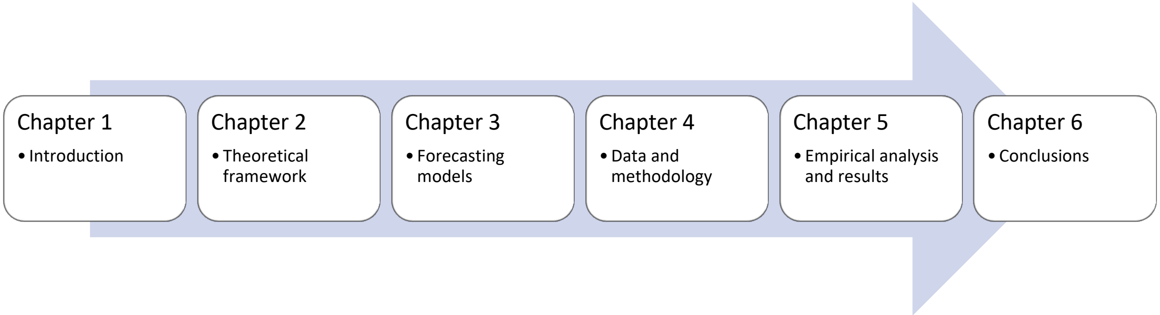


Figure 1. Structure of the thesis

In chapter 1 motivation and background of the study, focus of the study, research questions and objectives are presented. In chapter 2 the theoretical framework of the study is gone through with explaining basic concepts related to the study and reviewing previous research done on the topic of this research. Chapter 3 presents the concepts of the forecasting models utilized in this research and the tests used for data analysis in order to utilize and evaluate the performance of the forecasting models. In chapter 4 the data used in this research is presented and descriptive analysis of the data is conducted. Chapter 5 regards the empirical analysis of this research presents the result of empirical analysis. In chapter 6 conclusions from empirical analysis and limitations of the research are discussed with suggestions for further research.

2 THEORETICAL FRAMEWORK – A LITERATURE REVIEW

2.1 Time series

Time series is a set of observations of values ordered according to their indices. It is a sort of a parameter that changes over time, for example price, cost, turnover or rate (Benrhmach et. al. 2020). Theoretical and empirical aspects of time series analysis is an integral part of the study of financial markets and economics (Terence et. al. 2008). The time series perspective, where economic cycles are being determined by various random shocks propagated throughout the economy over time is central part how modern macroeconomists nowadays view economic fluctuations (Whelan 2016). Therefore, time series analysis and prediction are major scientific challenges in a field of finance and economics. With time series analysis it is possible to describe and explain phenomenon over time and draw conclusions for decision-making. Prediction is seen as one of the main objectives of time series study, which means predicting the future values of the series from its' previous observed values (Benrhmach et. al., 2020). For successful time series analysis, the notion of stationarity is essential. If stationary time series Y_t with $t = 1, \ldots, n$ and $n \in \mathbb{N}^*$ is a series whose properties are not changed over time we are led to following definition (Benrhmach et. al. 2020):

Definition 1. A stochastic process $(Y_t, t \in \mathbb{Z})$ is weakly stationary, if for any finite sequence of instants $t_1, \dots, t_k, k \in \mathbb{N}^*$, and for any integer t, the joint law of $Y_{t_1+t}, \dots, Y_{t_k+t}$does not depend on t.

Definition 2. A process is stationary $(Y_t, t \in \mathbb{Z})$, if:

1. $\forall t \in \mathbb{Z}, E[Y_t] = \mu$ (independent of t),
2. $\forall t \in \mathbb{Z}, E[Y_t^2] < \infty$ (independent of t),
3. $\forall t \in \mathbb{Z}, \forall k \in \mathbb{Z}, cov(Y_t, Y_{t+k} = \gamma(k)$(independent of t).

If the statistical characteristics of the stochastic process Y_t vary during the observation period it is said to be nonstationary. Therefore, stationarity can be summarized as temporal homogeneity (Benrhmach et. al. 2020).

2.2 Time series modelling and forecasting

The main objective of time series modelling, and forecasting, is to study techniques and measures for drawing conclusions from past data. The time series models can be utilized to not only describe and analyze the sample data, but to make forecasts for the future. Handling any persistent patterns in data is one of the main advantages of time series models (Abdullah & Tayfur 2004). Traditional and more simple, linear time series modelling techniques, such as moving average (MA), autoregressive (AR) and autoregressive moving average (ARMA) models, operate under the assumption of constant variance and is used for modelling and predicting the mean behavior. Therefore, these models are rarely concerned with the effects of conditional variance (Würtz et. al. 2006).

Autoregressive conditional heteroskedastic (ARCH) and generalized autoregressive conditional heteroskedastic (GARCH) models have become key models in the analysis of financial time series data and especially in financial applications where the goal is to analyze and estimate volatility of the time series (Würtz et. al. 2006). ARCH/GARCH -type of models originate from econometrics and can capture volatility clustering of econometric data, which means a phenomenon where small changes tend to follow small changes, and large changes tend to follow large changes. This phenomenon is well recognized in financial and econometric time series and is called conditional heteroskedasticity (Pahlavni & Roshan

2015). Where conditional mean models, such as ARMA, are not able capture autoregressive conditional heteroscedastic effects of time series data, ARCH -model proposed by Engle (1982) and its later extension; GARCH -model proposed by Bollerslev (1986), are able to express these characteristics of time series data. The ARCH/GARCH-type models are nonlinear models which include past variances in the explanation of future variances. ARCH/GARCH -type models can generate accurate forecasts of future volatility over short horizons and are therefore crucial part of mean modelling and forecasting the future values of time series (Wang et. al. 2005).

Even though ARMA models are powerful and flexible in forecasting conditional mean, they are not able to handle the volatility and nonlinearity of the time series. Previous studies have shown that hybridization of potential univariate time series ARMA -models with GARCH -family models can be an effective way to overcome limitations of each components model and able to improve forecast accuracy. In recent years, hybrid forecasting models such as ARMA -model implemented with GARCH -model have been proposed to be applied to time series data, to produce better performing models and forecasts. (Pahlavni & Roshan 2015)

2.3 Neural Networks

ANN can be described in multiple ways. At one extreme, ANNs can be seen as a class of mathematical algorithms, since neural networks can essentially be regarded as a graphic notation for a large set of algorithms. These algorithms produce solutions to a number of different specific scientific and other problems. On the other hand, ANNs can be seen as synthetic neural networks that mimic the biological neural networks found in living organisms such as human brains (Batra 2014). As an intermediate conclusion ANNs can be described as large-scale parallel-distributed information processing systems which are composed of many internally connected nonlinear computational units, i.e., neurons (Baghirli 2015).

ANN learning happens as the weights of the network are adjusted along the layers, according to the relationship between the inputs and the desired outputs. Multilayer perceptron (MLP) is one of the most basic ANN models that is widely used in the approximation of nonlinear functions which describe complicated between independent and dependent variables (Baghirli 2015). MLP is a feed-forward neural network with one or more layers between input and output layers and it is trained with the backpropagation learning algorithm (Batra 2014). MLP was first developed to solve complex classification problems but were quickly used for nonlinear regression models and then for time series modelling and forecasting because of their universal approximation property (Benrhmach et. al. 2020). ANNs have been shown to be effective way to simulate nonlinear patterns. From the training data sets, hidden patterns that could be independent from any mathematical model, can be found easily. ANNs produce result with minimum mean squared error (MSE), if the same or similar patterns are recognized (Baghirli 2015). The estimation and identification of MLP models use advanced techniques and the determination of correct architecture is not easy. Therefore, MLP models are overparametrized by definition and error functions to be minimized have multiple local minima which lead to difficulties in implementation (Benrhmach et. al. 2020). Figure 2 shows an illustration of the architecture of ANN.

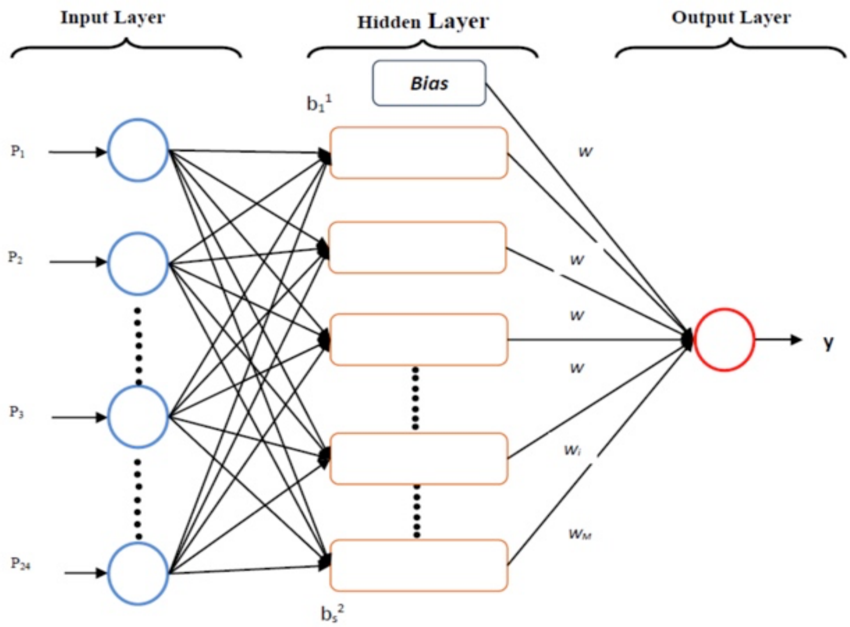


Figure 2. Architecture of ANN

The nonlinear autoregressive neural network (NAR) is a type of ANN that can be trained to predict future values of a time series from a set of that series past values $Y(t-1), Y(t-2), \ldots, Y(t-d)$ which are called feedback delays, where d is the time delay parameter. The network is first created and trained in open loop, utilizing the real values as target values as a response and making sure that the greater approximation being very close to the real values in training (Benrhmach et. al. 2020). Afterwards the network is converted into closed loop and the predicted values are used as new response inputs to the network (Benmouiza & Cheknan 2013). Figure 3 present the architecture of NAR.

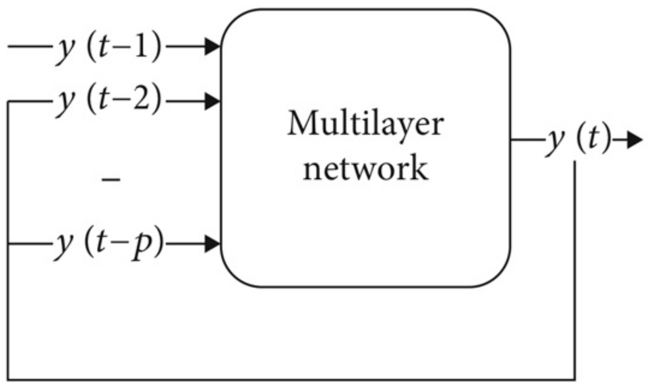


Figure 3. Architecture of NAR

The optimization of the neural network architecture aims at reducing the number of synapses (weights) and neurons as much as possible to reduce the complexity of the network and maintaining the generalization capabilities as well as improve the computing times (Benrhmach et. al. 2020). NAR networks are based on training algorithms that are used to adjust the weight values to get a desired output when certain inputs are introduced for the network (Benmouiza & Cheknan 2013). Two main approaches are presented in the existing literature concerning the optimization of the network (Benrhmach et. al. 2020):

1. Selection approach, which consists of starting the construction of a complex network which contains a large number of neurons and then reducing the number of unnecessary neurons and remove redundant connections or at the end of learning
2. Incremental approach, where the construction starts with the simplest possible network structure, and then neurons or layers are added until the optimal architecture is reached

An effective way for building NAR architecture is to estimate the prediction error using a set of data which was not used to construct the predictor meaning not used for learning. This set of data is called a test set. The dataset should be divided into three subsets of target timesteps as follows (Benrhmach et. al. 2020):

1. Training: this dataset is presented to the network during training and the network is adjusted in order to its error
2. Validation: this dataset is used for measuring network generalization and to stop the training when the generalization stops improving

3. Testing: this dataset has no effect on training and therefore provide an independent measure of network performance during and after training.

2.4 ANNs and ARMA-GARCH -models in GDP growth forecasting – a review of previous research

Quite few studies have been done comparing linear and non-linear forecasting models to ANN models in forecasting GDP growth. There is however some evidence, that ANN models are useful alternatives to econometric models in econometric time series modelling and in forecasting. Also there seems to be quite scarce number of studies done on forecasting GDP growth with hybrid ARMA-GARCH -models. However, some evidence of using ARMA-GARCH -models in economic growth, and other macroeconomic variable forecasting can be found.

Nkwatoh (2016) investigated the possibility to utilize ARIMA-GARCH models in modelling and forecasting GDP growth rate of Cameroon. The purpose of the research was to find out what time series model would be best to model and forecast Cameroon’s economic growth and help to achieve aspiration to become like China by 20135 in terms of economic growth. As the result of the research, the author found out that the best model for projecting Cameroon’s future economic growth rates is ARIMA(0,1,3)-GARCH(1,2) (Nkwatoh 2016).

Studies using ARMA-GARCH models in predicting other macroeconomic variables than GDP has also been done. Research by Floros (2005) shows that MA(4)-ARCH(1) -model outperform regular MA(4) model in forecasting unemployment rate in the United Kingdom. Kamil and Noor (2006) made a study of comparing ARMA-GARCH model with simple ARCH model and found out that the hybrid ARMA-GARCH model produce more accurate forecast for the price of raw palm oil in Malaysia. Hennani (2013) performed a study of using ARMA-GARCH -model in financial forecasting. The study focuses on modelling and forecasting S&P 500, CACC40 and FTSE 100 daily stock indexes on the time horizon of 420 time-steps using ARMA-GARCH -models and comparing the results with those of support vector machine learning (SVM) algorithms. The result of the study showed that ARMA-GARCH -models could not beat SVM -algorithm.

One informational study in using ANN models in GDP forecasting is concluded by Greg Tkacz. In his research “Neural network forecasting of Canadian GDP Growth” Tkacz (2001) examines differences of forecasting errors between univariate time series models, linear models and neural network NN models. The goal of his research was to improve the accuracy of financial and monetary forecasts of Canadian GDP using NN models. As dependent variables in his research Tcakz (2001) used financial and monetary variables such as one and four-quarter cumulative growth rates of real Canadian GDP. Explanatory variables in his research were US and Canadian interest rate yields, the real 90-day Corporate Paper rate, the growth rates of real narrow and broad monetary aggregates and Toronto Stock Exchange 300 index as representee of the growth rate of real stock prices. He found out that NN models produce statistically lower forecast errors in forecasting yearly growth rate of real GDP compared to univariate time series and linear models. On the other hand, he found out that differences in forecasting errors are not so significant when forecasting growth rate of quarterly real GDP and that NN models are unable beat a naive no-change model. Tcakz's (2001) conclusion is that at the one-quarter horizon none of the models performs exceptionally well compared to no-change models and that the chosen monetary and financial variables yield very large forecast errors for growth rate of real output. When examining four-quarter horizon he found out that NN models yields a forecast error on average about 0.25 percent lower than the best linear model and that the chosen variables seem to be much better predictors for the output growth rate in this time horizon. The author justifies his decision to focus specifically to forecast performance of NN models with three explanations. First argument is that NNs are data-driven models which can learn from and adapt to underlying relationships which is useful if there are not any previous beliefs about functional forms. Secondly, he argues that NNs are universal functional approximators and could approximate functional form to given level of accuracy if NNs are properly specified. Final argument is based on earlier findings that macroeconomic data follows non-linear processes and since NNs are non-linear in nature they would be good in forecasting such data (Tcakz 2001). The justification to focus on NN models in forecasting are well argued. The chosen variables for the research seem to work somewhat as good economic predictors, but the decision to focus on only financial and monetary variables seems a bit incoherent since based on the research in economic growth theories we know that the real output growth rate is not based only on macroeconomic contributors. The author seems to reflect the effect of human capital in real growth rate of national GDP.

Tcakz's informational research has inspired also other researcher to investigate utilizing ANNs in GDP forecasting. In 2018, Malte Jahn demonstrates the capabilities of ANN regression models in forecasting GDP growth of 15 industrialized economies between 1996 and 2016 in his working paper "Artificial neural network regression models: Predicting GDP growth". The author presents theoretical framework and precise algorithm which is used in training the ANN. Jahn (2018) investigates the capability of ANN regression model to generate more accurate predictions in GDP growth forecasting of those 15 industrialized countries than corresponding linear models. The results of his research come to close to those of Tcakz's (2001). However, the author criticizes Tcakz's model being unable to recognize negative growth rate. Jahn succeeds in building a better performing regression model using ANN than linear regression model in GDP growth forecasting. As a result, the root mean squared error (RMSE) of the built ANN model is 0.555, which is lower than RMSE of built linear regression model which was 1.833 (Jahn 2018). They key foundation of the Author's research is that ANN model gives more realistic forecasts compared to those of linear models. As an example, the author demonstrates capability of ANN see the economy recovering after 2009 financial crisis with increasing GDP growth rates after the crisis, whereas linear models only suggest general decline after 2009 financial crisis in GDP growth and is not able to forecast the recovering (Jahn 2018).

Zukime (2004) studied possibility to predict GDP growth of Malaysia with neural networks using knowledge-based economy indicators. The author used the backpropagation technique, a delta-learning rule and a sigmoid transfer function as learning and pattern recognition methods for neural networks. As the result, the best performing network models obtained consisted of 4 input units, one hidden layer with one hidden unit, and on output unit. The performance of the models was measured with Root Mean Squared Error and Mean Absolute Error and the result was that these values for built neural network models were better than those of used econometric model in comparison. used econometric model in comparison was regression model of consisting of those same input variables than in neural network model (Zukime 2004). The study is conducted well and seem valid. However, the time series data was collected from time period of 1995 to 2000, so the number of observations included in the procedure was quite low, which may be argued to affect the validity of the research.

Zhang et. al. (2018) performed a study of forecasting accuracy between of ARNN, ARIMA and GARCH models in predicting the Baltic dry index. The Baltic dry index is a sort of "barometer" used to evaluate the shipping industry, international trade and the global economy. The comparison between econometric models and ANN- based algorithms were made of using daily, weekly, and monthly data. The result of the study was that ANN technique predicts the most accurate weekly and monthly values for the index. It was also found out that ARIMA and GARCH models produce better forecasts for short-term forecast and especially for daily forecast. Author points out that ANNs are sensitive to input data and that ANN model predictions vary a lot so that no particular model is best for all of the scenarios (Zhang et. al. 2018).

Samir (2016) made a study in comparing ANN and time series models for predicting quarterly GDP in Palestine. The author used simulation method and real data for GDP. Result of the study was that ANNs outperform time series models, which were ARIMA and regression -models. The performance of the models was measured with Root Mean Squared Error (RMSE) and the final conclusion is that ANNs perform better that traditional methods in forecasting GDP in Palestine (Samir 2016).

3 TIME SERIES FORECASTING MODELS

This chapter presents the mathematical foundation behind different time series modelling and forecasting techniques and their evaluation. At first, traditional conditional mean modelling techniques, such as AR, MA and ARMA models are presented. For second, different GARCH -models for conditional volatility modelling and forecasting of time series are introduced. After that, three different training algorithms for training NAR are presented. The last part of this chapter focuses on mathematical theory behind goodness-of-fit tests and model performance evaluation.

3.1 Conditional mean models

3.1.1 AR model

In an AR -model for time series modelling the current value of a variable y depends on only the values that the variable took in previous periods added with an error term. An autoregressive model denoted as AR(p) for order p is written as:

$$y_t = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + u_t \qquad (1)$$

where the white noise error term is u_t. To demonstrate the properties of an autoregressive model, the equation above can be rewritten more compactly using sigma notation (Brooks 2008):

$$y_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-1} + u_t \qquad (2)$$

or by utilizing the lag operator as:

$$y_t = \mu + \sum_{i=1}^{p} \phi_i L^i y_t + u_t \tag{3}$$

or alternatively

$$\phi(L) y_t = \mu + u_t \tag{4}$$

where;

$$\phi(L) = \left(1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p\right) \tag{5}$$

3.1.2 MA model

MA model is one of the simplest modelling techniques for time series modelling. MA model is a linear combination of white noise processes, where y_t depends on the current values and previous values of white noise error term. When u_t with $(t = 1,2,3, \dots)$ is white noise process with $E(u_t) = 0$ and $var(u_t) = \sigma^2$, then (Brooks 2008):

$$y_t = \mu + u_t + \theta_1 u_{t-1} + \theta_2 u_{t-2} + \cdots + \theta_q u_{t-q} \tag{6}$$

is a qth order moving average process denoted as MA(q). This can be expressed using sigma notation as (Brooks 2008):

$$y_t = \mu + \sum_{i=1}^{q} \theta_i u_{t-1} + u_t \tag{7}$$

If the equation above is manipulated by introducing the lag operator, then then it could be expressed as $Ly_t = y_{t-1}$ to denote that y_t is lagged once. To demonstrate that that the value y_t took i periods ago the notation would be written $L^i y_t = y_{t-i}$. If using the lag operator, then the equation above can be written as (Brooks 2008):

$$y_t = \mu + \sum_{i=1}^{q} \theta_i L^i u_t + u_t \tag{8}$$

or alternatively

$$y_t = \mu + \theta(L) u_t \tag{9}$$

where;

$$\theta(L)01 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q. \tag{10}$$

The characteristics of MA(q) process of shown above are:

1. $E(y_t) = \mu$ (11)
2. $var(y_t) = \gamma_0 = \left(1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2\right)\sigma^2$ (12)
3. covariances γ_s

$$= \begin{Bmatrix} \left(\theta_s + \theta_{s+1}\theta_1 + \theta_{s+2}\theta_2 + \cdots + \theta_q\theta_{q+s}\right)\sigma^2 & for\ s = 1,2,\ldots,q \\ 0 & for\ s > q \end{Bmatrix} \tag{13}$$

Therefore, MA process has constant mean and variance as well as autocovariances which may differ from zero up to lag q and will be zero always after that (Brooks 2008).

3.1.3 ARMA model

Brooks (2008) explains that an ARMA -model for time series modelling, is a model which explains that the current value of time series y is dependent linearly on its own previous values added with a combination of current and previous values of white noise error term. Thus, ARMA(p, q) model is a combination of AR(p) and MA(q) models where the characteristics of an ARMA process will be a combination of those from the autoregressive and moving average parts. The ARMA model could be written as (Brooks 2008):

$$\phi(L) = \mu + \theta(L)u_t \tag{14}$$

where;

$$\phi(L) = 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p \tag{15}$$

and

$$\theta(L) = 1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_p L^p \tag{16}$$

or alternatively

$$y_t = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \theta_1 u_{t-1} + \theta_2 u_{t-2} + \cdots + \theta_q u_{t-q} + u_t \quad (17)$$

with;

$$E(u_t) = 0; E(u_t^2) = \sigma^2; E(u_t u_s) = 0, t \neq s \quad (18)$$

Since ARMA process has characteristics of both AR and MA parts will partial autocorrelation function (PACF) be extremely useful in this context because autocorrelation function (AFC) alone can only distinguish between a pure autoregressive and a pure moving average process. Because an ARMA process has geometrically declining ACF as do also a pure AR process, will PACF be helpful for distinguishing between an AR(p) process and an ARMA(p, q) process, since the latter has both declining ACF and PACF while the former has only declining ACF and PACF cutting off to zero after p lags. The mean of ARMA process series can be written as (Brooks 2008):

$$E(y_t) = \frac{\mu}{1-\phi_1-\phi_2-\cdots-\phi_p} \quad (19)$$

3.2 Conditional volatility models

3.2.1 GARCH model

GARCH -model is enhanced version of ARCH -model developed by Bollerslev (1986) and Taylor (1986). Weaknesses of ARCH model are possible violation of non-negative constraints and required high number of lagged squared residuals in order to capture all dynamics of time series' conditional variance, so GARCH -model was developed to invalidate some of these weaknesses of ARCH model. As conditional variance depends on model's previous lags and squared lagged residuals in GARCH -model, it is more parsimonious compared to ARCH -model because the lagged conditional variance in the model requires less lagged squared residuals to capture the volatility dynamics in the time series data. This leads to the fact that the developed GARCH -model has less parameters to estimate. Compared to ARCH -model, GARCH -model is less probable to indicate overfitting, so therefore GARCH -model is less likely to violate non-negative constraints (Sutelainen 2019). Since financial data has tendency to have leptokurtic characteristics GARCH -model is able to capture these characteristics of the data (Brooks 2008). GARCH -model with q lags of squared residuals and p lags of conditional variance can be formulated to model the conditional variance as follows (Brooks 2008):

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \, u_{t-i}^2 + \sum_{j=1}^{p} \beta_j \, \sigma_{t-j}^2 \tag{20}$$

where conditional variance is σ^2 and residual is u. Parameters ω, α_i and β_j parameters are required to have positive values to fulfill non-negative restraints of the model since negative variance is not possible. It is required also that parameters $\alpha_i + \beta_j < 1$ to make sure that long term variance always equals the predicted variance σ_t^2. This can be also formulated as $\frac{\omega}{1-\alpha_i-\beta_j}$, so therefore variance reverts to its long-term mean (Brooks 2008).

Even though GARCH -model can capture leptokurtic distributions and volatility clustering of financial time series data there are some weaknesses in the model (Sutelainen 2019). GARCH -model is not able to detect possible asymmetries in volatility size and sign bias because the sign bias is lost when the residuals of the model are squared, and positive and negative shocks have an impact to the same extent (Brooks 2008).

Brooks (2008) explains that GARCH -model with one lag of conditional variance and one lag of squared residual is most of the time sable to express volatility clustering in the time series data and no higher order GARCH models are usually implemented to economic time series data in the current literature. For this reason, no higher order than GARCH(1,1) -model is used in this thesis as a part of hybrid ARMA-GARCH -model.

3.2.2 EGARCH model

One of the first models developed to capture asymmetric volatility characteristics in the data, such as sign and size bias was done by Nelson (1991). This model was EGARCH it is asymmetric extension to the standard GARCH -model. When EGARCH -models has q lagged squared residuals and p lags of conditional variance, the model takes formulation of (Brooks 2008):

$$\ln(\sigma_t^2) = \omega + \sum_{i=1}^{q} (\alpha_i u_{t-1} + \gamma_i(|u_{t-i}| - E|u_{t-1}|)) + \sum_{j=1}^{p} \beta_j \ln(\sigma_{t-j}^2) \tag{21}$$

In the equation γ_i denotes the size bias and α_i denotes the sign bias (Sutelainen 2019). Even though the parameters would be negative, the model will not violate the constraints of non-negative parameters as the natural logarithm of conditional variance is modelled, so therefore there are no reason to make sure that the non-negativity constraints are violated by parameters (Brooks 2008).

Previous research has shown that no higher order models than EGARCH(1,1) are commonly been utilized in conditional volatility modelling and forecasting, since those are not performing sufficiently enough (Brooks 2008). Therefore, no higher order than EGARCH(1,1) -model is used in this thesis as a part of hybrid ARMA-GARCH -model.

3.2.3 GJR-GARCH model

GJR-GARCH model is also asymmetric extension of the standard GARCH model. GJR-GARCH was developed by Glosten, Jagannathan and Runkle (1993). The model takes into account the asymmetric size and sign effects with additional term. If the model has q lags of squared residuals and p lags of conditional variance, then the GJR-GARCH(p,q) can be formulated as follows (Brooks 2008):

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i u_{t-1}^2 + \sum_{i=1}^{q} \gamma_i u_{t-i}^2 l_{t-1} + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 \tag{22}$$

Where the dummy variable is denoted by indicator l_{t-1} taking value of 1 when the value of lagged residual u_{t-i} is below zero and value of 0 in other situations. Leverage effects are captured by coefficient γ_i and when $\gamma_i > 0$ positive shocks have smaller effect on the conditional variance than negative shock of the same size. Some parameters of the model have non-negativity constraints. When $\omega > 0$, $\alpha_i > 0$, $\beta_j \geq 0$ and $\alpha_i + \gamma_i \geq 0$ these constraints are not violated (Sutelainen 2019). Even though coefficient γ_i is below zero, GJR-GARCH model is still sufficient enough as long as condition of $\alpha_i + \gamma_i \geq 0$ is fulfilled. GJR-GARCH model transform to standard GARCH(p,q) model if the leverage term $\gamma_i = 0$ (Brooks 2008).

Similarly, with GARCH and EGARCH -models, higher order variants other than GJR-GARCH(1,1) -model are not commonly used in previous research in volatility forecasting (Brooks 2008). This seem to be a general practice with volatility forecasting models, and because of this practice only GJR-GARCH(1,1) -model will be utilized in this thesis as a part of hybrid ARMA-GARCH-model.

3.3 Nonlinear Autoregressive Neural Network Model

NAR is based on linear autoregressive model with feedback connections, including several layers of the network, and is therefore a recurrent dynamic network. NAR is commonly used in multi-step ahead prediction of time series (Benrhmach et. al. 2020). NAR which is applied to the time series forecasting describes a discrete nonlinear autoregressive model that can be written as (Benrhmach et. al. 2020):

$$Y_t = h(Y_{t-1}, Y_{t-2}, \dots, Y_{t-d}) + \varepsilon_t \tag{23}$$

where the function $h(\cdot)$ is unknown in advance. The training of NAR aims to approximation of the function by means of the optimization of the weights and neuron bias of the network. NAR can be defined precisely by the following equation (Benrhmach et. al. 2020):

$$Y_t = \alpha_0 + \sum_{j=1}^{k} \alpha_j \phi \left(\sum_{i=1}^{a} \beta_{ij} Y_{t-1} + \beta_{0j} \right) + \varepsilon_t \tag{24}$$

where the number of entries is α, the number of hidden layers is k with activation function ϕ and the parameter corresponding to the weight of the connection between the input unit i and the hidden unit j is β_{ij}. The constants that correspond to the hidden unit j and the output unit are β_{0j} and α_0, respectively (Benrhmach et. al. 2020).

NAR networks are based on training algorithms that are used to adjust the weight values to get a desired output when certain inputs are introduced for the network. (Benmouiza & Cheknan 2013). The next sub-chapters introduce three different training algorithms used in this thesis in NAR architectures.

3.3.1 Levenberg-Marquardt algorithm

In the beginning of 1960's Levenberg-Marquardt (LM) algorithm was developed to solve nonlinear least squares problems. Gavin (2020) states that, "Least squares problems arise in the context of fitting a parameterized mathematical model to a set of data points by minimizing an objective expressed as the sum of the squares of the errors between the model function and a set of data points." The least squares' objective in the parameters is quadratic, if a model is linear in its parameters. With the solution to a linear matrix equation this objective may be minimized with a respect to the parameters. If the function to be fitted is not linear in its parameters, an iterative algorithm is required to solve the least squares problem. Through an implication of well-chosen updates to the model parameters' values these algorithms reduce the sum of the squared values of the errors between the model function and data points (Gavin 2020).

According to Gavin (2020) “The Levenberg-Marquardt algorithm combines two numerical minimization algorithms: the gradient descent method and the Gauss-Newton method.” The parameters are updated with the steepest-descent direction in the gradient descent method resulting to reduction of the sum of the squared errors. The Gauss-Newton method assumes that the least squares function is locally quadratic in the parameters. The sum of the errors is reduced by finding the minimum of this quadratic. When the parameters are far from their optimum value, the LM method behaves more like a gradient-descent method and when the values are close to their optimum values more like Gauss-Newton method (Gavin 2020).

The LM algorithm adjustably alters the parameter updates between the gradient descent update and the Gauss-Newton update according to following equation (Gavin 2020):

$$[J^T W J + \lambda I] h_{lm} = J^T W (y - \hat{y}) \tag{25}$$

where the local sensitivity of the function $\hat{y}$ to variation in the nonlinear function parameters p is represented by the $m \times n$ Jacobian matrix $[\partial \hat{y} / \partial p]$ and $[\partial \hat{y} / \partial p]$ is represented by the variable J for notational simplicity. Weighting matrix W is diagonal with $W_{ii} = 1/\sigma_{y_i}^2$ where measurement error for datum $y(t_i)$ is σ_{y_i}. Small values of the damping parameter λ result in Gauss-Newton update while large values result in gradient descent update. First updates are small towards the steepest-descent direction because the damping parameters λ is initially set to be large. If result of any iteration happens to worse approximation $(\chi^2(p + h_{lm} > \chi^2(p))$ then dumping parameter λ increases. in other cases, when the solution improves the dumping factor λ decreases and the LM method approaches the Gauss-Newton method resulting the solution to accelerating to the local minimum. The values of dumping factor λ are normalized to the values of $J^T W J$ in Marquardt’s update relationship with equation (Gavin 2020):

$$[J^T W J + \lambda \, diag(J^T W J)] h_{lm} = J^T W (y - \hat{y}) \tag{26}$$

For more detailed mathematical explanation of Levenberg-Marquardt algorithm, see article Gavin, P. (2020) “The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems”. In this thesis, LM algorithm is used as a one type of neural network training function among others in comparing different NAR models.

3.3.2 Bayesian Regularization algorithm

Bayesian regularization (BR) algorithm is a training function which updates the weights and bias values of NN according to LM optimization. BR algorithm minimizes the combination of squared errors and weights and determines then the correct combination to produce a NN that is well generalized (Li & Shi 2012). Network weights are introduced into to the training function denoted as $F(w)$ by BR algorithm as follows (Baghirli 2015):

$$F(w) = \alpha E_w + \beta E_D \tag{27}$$

where the sum of the squared network weights is denoted as E_w and the sum of network errors is denoted as E_D. The weights of the network are seen as random variables in the BR framework and therefore the distribution of the networks weights and training set are thought as a Gaussian distribution. Objective function parameters are α and β which are factors defined using the Bayes' theorem (Li & Shi 2012). Two variables A and B are related based on their prior (or marginal) and posterior (or conditional) probabilities in the Bayes' theorem as follows (Baghirli 2015):

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{28}$$

where the posterior probability of A conditional on B is $P(A|B)$, the prior of B conditional on A and the non-zero probability of event B is $P(B)$, which acts as a normalizing constant. Objective functions need to be minimized to find the optimal weight space. This is equivalent of maximizing the posterior probability function of (Yue et. al. 2011):

$$P(\alpha, \beta|D, M) = \frac{P(D|\alpha, \beta, M)P(\alpha, \beta|M)}{P(D|M)} \tag{29}$$

in which the factors needed to be optimized are α and β, the wight distribution is D, the particular neural network architecture is M, the normalization factor is $P(D|M)$, the uniform prior density for the regularization parameters is $P(\alpha, \beta|M)$ and the likelihood function of D for given α, β, M is $P(D|\alpha, \beta, M)$. Maximizing the likelihood function $P(D|\alpha, \beta, M)$ is equivalent to maximizing the posterior function $P(\alpha, \beta|D, M)$. Optimum values for α and β for given weight are found as a result of this process. Subsequently BR algorithm moves into LM phase where Hessian matrix calculations updates the weight space to minimize the objective function. If the converge is not reached after that, the algorithm estimates new values for α and β and the process is repeated by itself until the converge is found. (Yue et. al. 2011)

3.3.3 Scaled Conjugate Gradient algorithm

The Scaled conjugate gradient (SCG) is based on conjugate directions, but in contrast, it does not perform a line search at each iteration unlike other conjugate gradient algorithms which do require a line search at each iteration. SCG was developed by Martin Moller in 1990's. Since other conjugate gradient algorithms makes systems computationally expensive because of using line search at each iteration, SCG was developed to avoid the time-consuming line search (Badani et. al. 2016). The step size in SCG algorithm is a function of quadratic approximation of the error function. This makes the SCG algorithm more robust and also independent of user defined parameters. Different approach is used for estimating the step size. The second order term is calculated as (Badani et. al. 2016):

$$\bar{s}_k = \frac{E'(\bar{w}_k + \sigma_k \bar{p}_k) - E\prime(\bar{w}_k)}{\sigma_k} + \lambda_k \bar{p}_k \tag{30}$$

where scalar λ_k is adjusted every time according to the sign of δ_k. The step size is calculated as (Badani et. al. 2016):

$$\alpha_k = \frac{-\mu_k}{\delta_k} = \frac{-\bar{p}_j^T E'_{qw}(\bar{y}_1)}{\bar{p}_j^T E\prime\prime(\bar{w})\bar{p}_j} \tag{31}$$

Where weight vector in space R^n is $\bar{w}$ and $E(\bar{w})$ is the global error function. $E'(\bar{w})$ is notation for the gradient of error and $E'_{qw}(\bar{y}_1)$ is the quadratic approximation or error function. The non-zero set of weight vectors is $\bar{p}1_1, \bar{p}_2, \dots, \bar{p}_k$ and λ_k is updated so that (Badani et. al. 2016):

$$\bar{\lambda}_k = 2\left(\lambda_k - \frac{\delta_k}{|\bar{p}_k|^2}\right) \tag{32}$$

where if $\Delta_k > 0.75$, then $\lambda_k = \frac{\lambda_k}{4}$, and if $\Delta_k > 0.25$, then $\lambda_k = \lambda_k + \frac{\delta_k(1-\Delta_k)}{|\bar{p}_k|^2}$. The comparison parameter Δ_k is given by (Badani et. al. 2016):

$$\Delta_k = 2\delta_k[E(\bar{w}_k) - E(\bar{w}_k + \alpha_k \bar{p}_k)]/\mu_k^2 \tag{33}$$

The values are set initially as $0 < \sigma \leq 10^{-4}$, $0 < \lambda_l \leq 10^{-6}$ and $\bar{\lambda}_t = 0$ (Badani et. al. 2016).

In this thesis, SCG algorithm is used as a one type of neural network training function among others in comparing different NAR models.

3.4 Tests for stationarity

The assumption of stationarity data for time series modelling and predicting is a crucial part of fitting time series models into the data. If the data is not stationary it can't be modelled using traditional time series modelling techniques and has to be therefore transformed into stationary mode. Three different tests are used in this thesis to check the stationarity of the used data.

3.4.1 Augmented Dickey-Fuller test

Augmented Dickey-Fuller (ADF) test is used for checking the stationarity of time series data. The null hypothesis of the test is that process contains a unit root. The null hypothesis is assessed using model (MathWorks 2021a):

$$y_t = c + \delta t + \phi y_{t-1} + \beta_1 \Delta y_{t-1} + \cdots + \beta_p \Delta y_{t-p} + \varepsilon_t, \tag{34}$$

where the differencing operator is Δ, so that $\Delta y_t = y_t - y_{t-1}$. The number of lagged difference terms is p which is defined by the user and ε_t is a zero mean innovation process. The null hypothesis of a unit root is $H_0: \phi = 1$ under the alternstive hypothesis of $\phi < 1$. Different growth characteristics are allowed by the variants. When $\delta = 0$ the model has no trend component. When $c = 0$ and $\delta = 0$ the model has no trend or drift. If the test fails to reject the null hypothesis the possibility of a unit root is rejected meaning that the process is stationary (MathWorks 2021a).

3.4.2 KPSS test

Kwiatkowski, Phillips, Schmidt and Shin (KPSS) test is another test used for checking the stationarity of the data. It was developed by Kwiatkowski, Phillips, Schmidt and Shin (1992). The null hypothesis of the test is that a univariate time series is trend stationary, and the alternative hypothesis is that it is a nonstationary unit root process. The structural model of the test is (MathWorks 2021b):

$$y_t = c_t + \delta t + u_{1t} \tag{35}$$

$$c_t = c_{t-1} + u_{2t}, \tag{36}$$

where δ is the trend coefficient and u_{1t} is a trend stationary process. u_{2t} with zero mean and σ^2 variance. The null hypothesis of $\sigma^2 = 0$ indicates that the random walk term c_t acts as a model intercept and is constant. The alternative hypothesis is $\sigma^2 > 0$ meaning the unit root in the random walk. The test statistic is (MathWorks 2021b):

$$\frac{\sum_{t=1}^{T} S_3^2}{s^2 T^2}, \tag{37}$$

where the sample size is T, the Newey-West estimate of the long-run variance is s^2 and $S_t = e_1 + e_2 + \cdots + e_t$. The test uses regression to find the ordinary least squares (OLS) fit between the null model and the data. The test statistic follows a nonstandard distribution under the null hypothesis (MathWorks 2021b).

3.4.3 Variance ratio test

The last test used to check the stationarity of the time series data is variance ratio test for random walk. The null hypothesis of the test is that a univariate time series y is a random walk and the alternative hypothesis is that the time series follow nonrandom walk. The model for the null hypothesis is (MathWorks 2021c):

$$y_t = c + y_{t-1} + \varepsilon_t, \tag{38}$$

where a drift constant is c and uncorrelated innovations with zero mean is e_t. The test statistic is based on a ratio of variance estimates of returns $r_t = y_t - y_{t-1}$ and period q return horizons $r_t + \cdots + r_{t-q+1}$ (Mathworks 2021c).

3.5 Model fitness tests

3.5.1 Ljung-Box test for autocorrelation

Ljung and Box (1978) developed a test for serial correlation in data. This test is used in this thesis to check if the residuals of fitted ARMA model exhibit serial correlation. The test evaluates are all autocorrelations simultaneously equal to zero up to certain number of lags instead testing individual lags for autocorrelation. The equation for Ljung-Box test statistic is (Ljung & Box 1978):

$$Q_h = n(n+2)\sum_{k=1}^{h}\frac{\hat{\rho}_k^2}{n-k} \sim \chi_h^2 \tag{40}$$

Where size of the sample is n, number of tested lags is k and the sample autocorrelation of lag k is $\hat{\rho}_k$ which is squared to make sure that coefficient of different sign does not counterbalance each other. The test statistic follows chi-squared distribution with h degrees of freedom under the null hypothesis of no serial autocorrelation in k lags (Ljung & Box 1978).

In this thesis the number of lags used is 4 with the confidence level of 95 percent to test whether the null hypothesis of no serial autocorrelation is rejected. The number of used lags is 4 because of quarterly data. The null hypothesis of no serial autocorrelation at confidence level of α is rejected if $Q_h > \chi_{1-\alpha,h}^2$ (Ljung & Box 1978).

3.5.2 Engle's ARCH test

Engle's ARCH test for conditional heteroscedasticity was developed by Engle (1982) to test for the existence of ARCH behavior based on the regression since the ARCH model has the form of an autoregressive model (Wang et. al. 2005). The test assesses the null hypothesis that a series (r_t) of residuals exhibits no conditional heteroscedasticity effects. The alternative hypothesis of the test is ARCH(L) model describes the series.

The ARCH(L) model has the following equation (Wang et. al. 2005):

$$r_t^2 = a_0 + a_1 r_{t-1}^2 + \cdots + a_L r_{t-L}^2 + e_t \tag{41}$$

where there is at least one of the following: $a_j \neq 0, j = 0, \ldots, L$.

The resulting test statistic is the Lagrange multiplier statistic TR^2, where the sample size is T and R^2 is the coefficient of determination from fitting the ARCH(L) model for a number of lags (L) through regression. The asymptotic distribution of the test statistic is chi-square with L degrees of freedom under the null hypothesis (Wang et. al. 2005). Bollerslev (1986) suggested that the test should also have power against GARCH alternatives.

Engle's ARCH test is used in this thesis to detect the conditional heteroscedastic effects in time series residuals after fitting the conditional mean model. This is done to check the possibility to implement GARCH -models with ARMA -model.

3.5.3 Jarque-Bera test for normality

Financial time series data is rarely normally distributed and has leptokurtic characteristics such as fatter tails and over-peaking mean. GARCH -models can be built using different distributions, which allows models to fit this kind of time series data, and that is why normality of time series residuals after fitting conditional mean model must be tested. To test normality of residuals of ARMA -model, Jarque-Bera test for normality is used in this thesis. This test is also used in estimating how the original time series data is distributed. The test indicates whether the data has no excess skewness and if the kurtosis of the data is zero. Equation for calculating the test statistic for Jarque-Bera test is (Stephanie 2016):

$$JB = \left(\frac{n}{6}\right) \times \left[s^2 + \left(\frac{k^2}{4}\right)\right] \tag{42}$$

where s is skewness and k is kurtosis. Equations for skewness and kurtosis are:

$$s = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{(\frac{1}{n}\sum_{i=1}^{n}(x - \bar{x})^2)^{\frac{3}{2}}} \tag{43}$$

and

$$k = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{(\frac{1}{n}\sum_{i=1}^{n}(x - \bar{x})^2)^2} \tag{44}$$

The null hypothesis of the JB test is that the sample data follows normal distribution. The JB test statistic with two degrees of freedom follows chi-squared distribution. If the null hypothesis is rejected, then the test indicates that the sample data is not normally distributed., and therefore it would be useful to use other distribution in the forecasting models. The confidence level of 95 percent is used in this thesis to determine if the sample data is normally distributed (Stephanie 2016).

3.6 Loss functions MAE and MSE

To determine performance of the forecasting models, loss functions mean absolute error (MAE) and mean squared error (MSE) are used. Forecasts are usually produced for the whole out-of-sample period and then the forecasted values are compared with the actual values to obtain the forecast errors before errors are being aggregated in some way. To obtain forecast error for observation y, the difference between the value of observation y

and the forecasted value for observation $\hat{y}$ is calculated. When defining the forecast error this way, it will be positive if the forecasted value was lower than the actual value and vice versa, if the forecasted value was higher than the actual value. it is noteworthy, that it is not possible to simply sum the forecast errors because the positive and negative errors would rule each other out. To keep the forecast errors positive, they are usually squared, or the absolute error values are taken before aggregating. (Brooks 2008)

Conclusions from the individual values of MAE and MSE can't be made. However, values of MAE and MSE of one forecasting model can be compared to those of another forecasting model and that way the forecasting models can be ranked according to their forecasting performance. In contrast to MAE, MSE is quadratic loss function and therefore particularly advantageous in situations where larger forecast errors are more serious. This may be also disadvantage of MSE in situations where large value of forecast error is not more serious than lower value of forecast error. (Brooks 2008)

When denoting s-step-ahead forecasts of a variable at time t as $f_{t,s}$ and the actual value of that variable at time t as y_t, the equation for MSE is (Brooks 2008):

$$MSE = \frac{1}{T-(T_1-1)} \sum_{T=T_1}^{T} \left(y_{t+s} - f_{t,s}\right)^2 \tag{45}$$

where the total sample size (in-sample + out-of-sample) is T and the first out-of-sample forecast observation is T_1. In this way, estimation of the in-sample model runs from observation 1 to $(T-1)$ initially, and observations T_1 to T are available for out-of-sample estimation (Brooks 2008).

According to Brooks (2008) the average absolute forecast error is measured with MAE and the equation for the measure with same denotations is:

$$MAE = \frac{1}{T-(T_1-1)} \sum_{t=T_1}^{T} \left|y_{t+s} - f_{t,s}\right| \tag{46}$$

In this thesis, both MAE and MSE loss functions are used for evaluating and ranking the model performance of forecasting models. These loss functions are used as a main basis for decision criteria in ranking the models. Depending on the results, the ranking of the models may differ between the values of the loss functions. Final scores of each model are based on MAE and MSE loss functions.

4 DATA AND METHODOLOGY

The data being used to conduct the empirical research in this thesis is quarterly GDP growth rate of five Nordic countries: Denmark, Finland, Iceland, Norway and Sweden. The data is quarterly GDP growth rate of these five countries covering the time period from 1.4.1960 to 31.12.2019, so there is in total of 239 observations in the data for each of the five countries. Since the data is already a relative change in GDP growth presented as percentual change over previous quarter, there is no need to transform the data to logarithmic returns. Data is in form of a time series, so the quarterly growth rate is seen in relation to time. Time frame for which the data was drawn for was based on the availability of the data for all of the countries. The data was obtained from OECD Statistic database (OECD 2021).

For the forecasting procedure, the data for each country is dived into two sub-periods; in-the-sample period and out-of-sample period. The in-the-sample period consists of 219 quarterly observations covering the time period from 1.4.1960 to 31.12.2014 and the out-of-sample period consists of 20 quarterly observations covering the time period from 1.1.2015 to 31.12.2019.

Table 1 below summarizes the information of the time series data used in this thesis for five Nordic countries

Table 1. Information of the time series

Variable	Variable name	Obs.	In-the-sample	Out-of-sample	Describtion
Growth Rate of real GDP, Denmark	DEN_growth_rate	239	1.4.1960-31.12.2014 (219 observations)	1.1.2015-31.12.2019 (20 observations)	Growth rate of real GDP compared to previous quarter in Denmark, seasonally adjusted
Growth Rate of real GDP, Finland	FIN_growth_rate	239	1.4.1960-31.12.2014 (219 observations)	1.1.2015-31.12.2019 (20 observations)	Growth rate of real GDP compared to previous quarter in Finland, seasonally adjusted
Growth Rate of real GDP, Iceland	ISL_growth_rate	239	1.4.1960-31.12.2014 (219 observations)	1.1.2015-31.12.2019 (20 observations)	Growth rate of real GDP compared to previous quarter in Iceland, seasonally adjusted
Growth Rate of real GDP, Norway	NOR_growth_rate	239	1.4.1960-31.12.2014 (219 observations)	1.1.2015-31.12.2019 (20 observations)	Growth rate of real GDP compared to previous quarter in Norway, seasonally adjusted
Growth Rate of real GDP, Sweden	SWE_growth_rate	239	1.4.1960-31.12.2014 (219 observations)	1.1.2015-31.12.2019 (20 observations)	Growth rate of real GDP compared to previous quarter in Sweden, seasonally adjusted

The choice of using growth rate of real GDP compared to previous quarter was done on the basis that growth rates offer scale-free analysis of economic growth of different sized economies. Also, the fact that changed currencies won't affect measured economic growth favored using growth rates instead of raw real GDP volumes. The choice of using seasonally adjusted data was made on the assumption of statistical models requiring stationary, not seasonally affected time series data.

The next sections introduce seasonally adjusted growth rate of real GDP compared to previous quarter in all five Nordic countries and provides comprehensive data analysis including graphical examinations of the distribution, statistical tests to verify the results of graphical examination and tests for stationarity.

4.1 Descriptive statistics of Quarterly Growth Rate of real GDP, Denmark

Growth rate of real GDP in Denmark is presented as variable named "DEN_growth_rate" and it is seasonally adjusted growth rate of real gross domestic product compared to previous quarter. Sample size consists of 239 quarterly observations of growth rate percentage for time period from 1.4.1960 to 31.12.2019 and it is divided into two sub-samples: in-the-sample period consisting of 219 quarterly observations from 1.4.1960 to

31.12.2014 and out-of-sample period consisting of 20 quarterly observations from 1.1.2015 to 31.12.2019.

Graphical evaluation starts with investigating the time series of DEN_growth_rate -variable which is presented in figure 4 showing quarterly observations plotted as a time series for the whole sample period. The quarterly real GDP growth rate seems to fluctuate around its mean which is assumed to be between zero and two percent according to the plotted time series. In the 60's and 70's Denmark's economic growth has been stable. After that, the economic growth starts to fluctuate more. The burst of the tech-bubble around 2000 can be seen slowed down the economic growth a little. Financial crisis in 2008 affected negatively economic growth in Denmark as did European debt crisis in 2011.

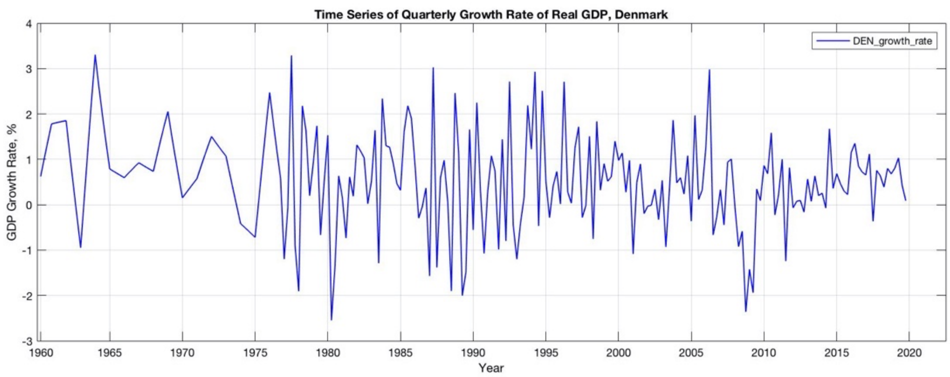


Figure 4. Time series of DEN_growth_rate

There are some alternating periods of high and low volatility in figure 4 which could imply volatility clustering indicating that the data has autocorrelation among observations since large deviations from the mean seem to be followed by large deviations of other sign from the mean. To investigate further possible autocorrelation in the data it is useful to check plotted ACF which is presented in figure 5.

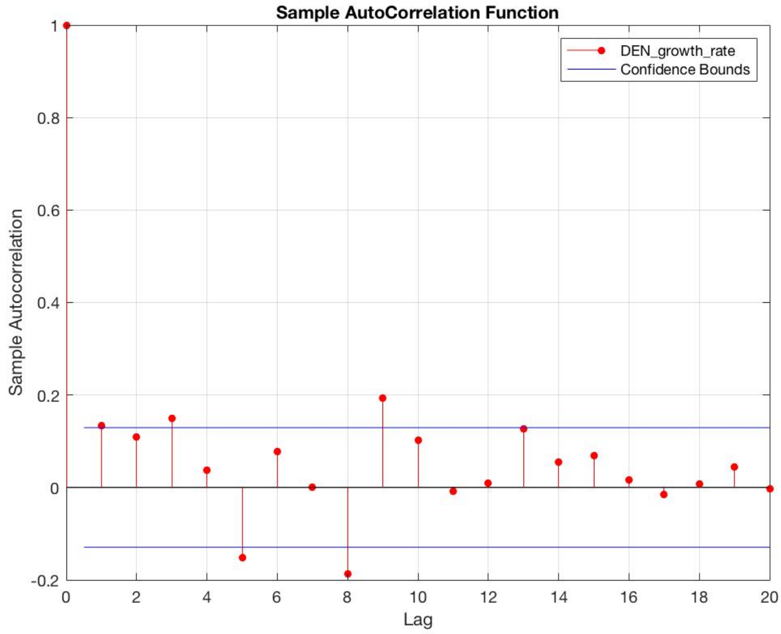


Figure 5. ACF -plot of DEN_growth_rate

Figure 5 above indicates that there are positive and negative autocorrelations in the data until up to lag 9. Lags 3,5,8 and 9 have autocorrelations above the 95% confidence bounds, which implies statistically significant autocorrelation differing from zero.

To analyze normality of the time series graphically, histogram of time series with normal distribution in figure 6, and QQ-plot in figure 7 of the time series are evaluated. From the histogram of DEN_growth_rate -variable in figure 6 it can be seen that the observations are not normally distributed since the histogram has slightly bigger tails and slight excess kurtosis around the mean compared the normal distribution. It can be also seen that the observations are skewed slightly to the left.

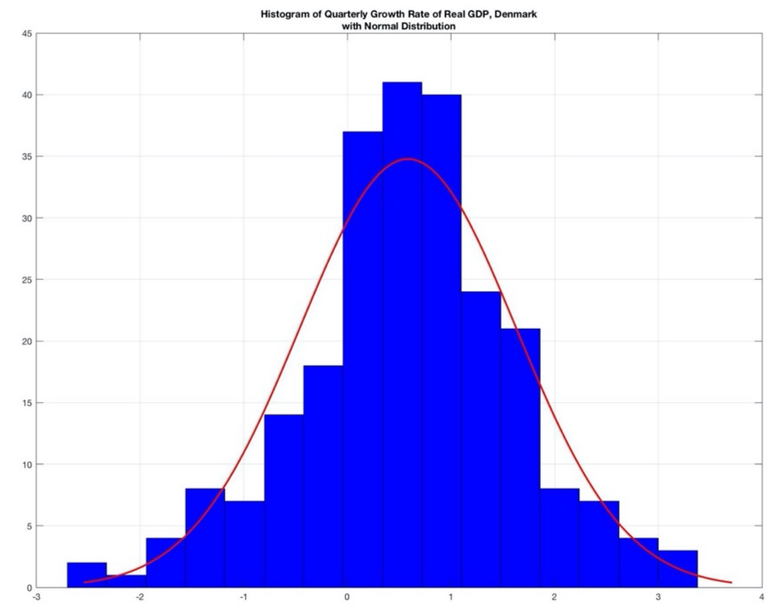


Figure 6. Histogram of DEN_growth_rate

The QQ-plot below shows some slight leptokurtic characteristics in the DEN_growth_rate time series especially for the tails of the distribution. The tails of the distribution are bigger and doesn't follow linearity in the plot indicating not normally distributed time series.

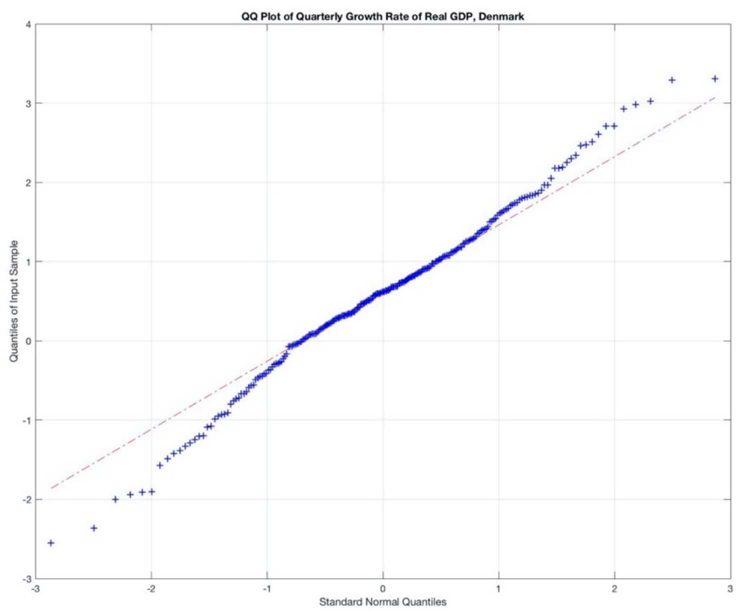


Figure 7. QQ -plot of DEN_growth_rate

To validate the results from graphical investigation of the time series, summary statistics and statistical tests are performed. Summary statistic are presented in table 2. Mean of the time series is 0.5852 while median and standard deviation being 0.6181 and 1.10419, respectively. Maximum value of the time series is 3.3064 and minimum value is -2.5489. Value of -0.1619 for skewness validates the insight from graphical evaluation that the data of the distribution is skewed more to the left of the mean and value of 3.4083 for kurtosis the result that the distribution has slightly excess kurtosis. These results support the findings that the data is not normally distributed since normally distributed data should have value of zero for skewness and value smaller than three for kurtosis.

Table 2. Summary statistics of DEN_growth_rate

Variable	Mean	Median	Standard deviation	Max	Min	Skewness	Kurtosis
DEN_growth_rate	0.5852	0.6181	1.0419	3.3064	-2.5489	-0.1619	3.4083

For verifying the results statistically, Jarque-Bera test for normality is performed for the time series. Results of the test presented in table 3. Null hypothesis that DEN_growth_rate - variable is normally distributed fails to be rejected with test statistic of 2.7041 and p-value of 0.213 with significance level of 0.050. Critical value for the test statistic is 5.7238. Failing to reject the null hypothesis suggests that the time series data for DEN_growth_rate is actually normally distributed in contrast to the graphical evaluation and summary statistics.

Table 3. Result of Jarque-Bera test for DEN_growth_rate

Jarque-Bera Test for Normality				
H_0: DEN_growth_rate is normally distributed				
Null Rejected	P-Value	Test Statistic	Critical Value	Significance Level
FALSE	0.213	2.7041	5.7238	0.050

Conditional mean modelling using ARMA models for time series data, requires the data to be stationary. To find out if the data is stationary, Augmented Dickey-Fuller (ADF) test with 4 lags is used for DEN_growth_rate because the data used is quarterly data. Null hypothesis of the test is that DEN_growth_rate contains a unit root. This null hypothesis is rejected with test statistic of -4.8671 and p-value of 0.001 with significance level of 0.050. Critical value for the test statistic is -1.9421. Rejection of the null hypothesis means that the data is stationary. Table 4 below present the statistics of ADF test.

Table 4. Result of ADF test for DEN_growth_rate

Augumented Dickey-Fuller Test					
H_0: DEN_growth_rate contains a unit root					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Significance Level
TRUE	0.001	-4.8671	-1.9421	4	0.050

To complement ADF test results KPSS test for stationarity with 4 lags is performed for DEN_growth_rate time series. Null hypothesis of the test is that DEN_growth_rate is trend stationary which fails to be rejected with test statistic of 0.0861 and p-value >0.100 with significance level of 0.050. The critical value for the test is 0.1460. The result of the test complements the result of ADF test that the time series is stationary. Table 5 presents the results of KPSS test.

Table 5. Result of KPSS test for DEN_growth_rate

KPSS Test for Stationarity					
H_0: DEN_growth_rate is trend stationary					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Signifigance Level
FALSE	>0.100	0.0861	0.1460	4	0.050

For next, DEN_growth_rate time series is tested for being random walk with Variance ratio test for random walk. In this test the number of used periods is 4 since the data used is quarterly data. The null hypothesis of the test is that DEN_growth_rate is a random walk. This null hypothesis is rejected with test statistic of -4.0533 and p-value of 0.001 with significance level of 0.050. The critical value for the test is 1.9600. The result of the test is that DEN_growth rate follows non-random walk. The results are presented in table 6.

Table 6. Result of Variance ratio test for DEN_growth_rate

Variance Ratio Test for Random Walk					
H_0: DEN_growth_rate is a random walk					
Null Rejected	P-Value	Test Statistic	Critical Value	Period	Signifigance Level
TRUE	0.001	-4.0533	1.9600	4	0.050

4.2 Descriptive statistics of Quarterly Growth Rate of real GDP, Finland

Growth rate of real GDP in Finland is presented as variable named “FIN_growth_rate” and it is seasonally adjusted growth rate of real gross domestic product compared to previous

quarter. Sample size consists of 239 quarterly observations of growth rate percentage for time period from 1.4.1960 to 31.12.2021 and it is divided into two sub-samples; in-the-sample period consisting of 219 quarterly observations from 1.4.1960 to 31.12.2014 and out-of-sample period consisting of 20 quarterly observations from 1.1.2015 to 31.12.2019.

Graphical evaluation starts with investigating the time series of FIN_growth_rate -variable which is presented in figure 8 showing quarterly observations plotted as a time series for the whole sample period. The quarterly real GDP growth rate seems to fluctuate around its mean which is assumed to be between zero and two percent according to the plotted time series. The temporarily drop in the quarterly real GDP growth rate in Finland can be clearly seen during the early 90's in following the pan-Nordic financial crisis. It can be also seen that the recovery to obtain the earlier level in economic growth is not reached until in 1995. The burst of the tech-bubble around 2000 also had its negative impact on economic growth in Finland when the growth rate of quarterly real GDP lunges again although recovering sufficiently fast. However, it is clear to see that even bigger negative lunge happened in 2008 because of the global financial crisis followed by the turbulent times of European debt crisis in 2011.

Figure 8. Time series of FIN_growth_rate

High and low alternating periods of volatility in figure 8 implies volatility clustering indicating that the data has autocorrelation among observations since large deviations from the mean seem to be followed by large deviations of other sign from the mean. To investigate further possible autocorrelation in the data it is useful to check plotted autocorrelation function (ACF) which is presented in figure 9.

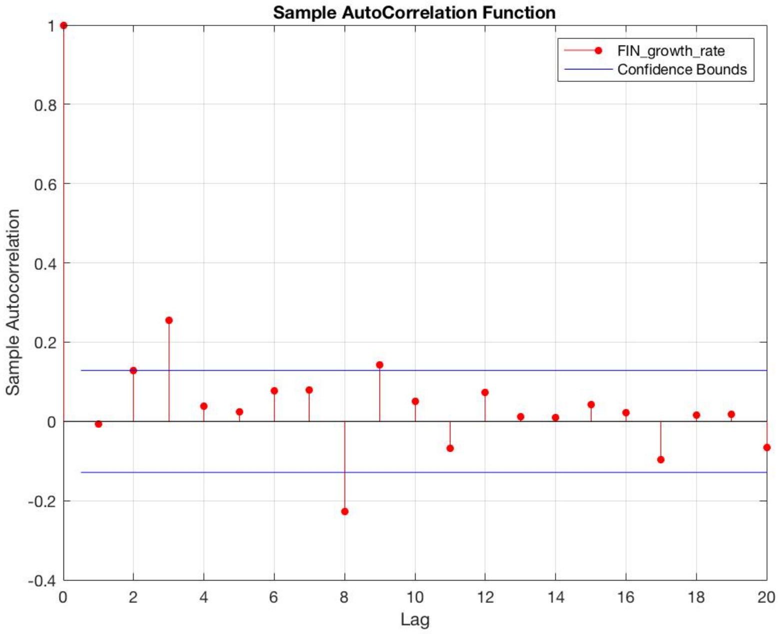


Figure 9. ACF -plot of FIN_growth_rate

Figure 9 above indicates that there are positive and negative autocorrelations in the data until up to lag 9. Some of the lags' autocorrelations are above the 95% confidence bounds, which implies statistically significant autocorrelation differing from zero.

To analyze normality of the time series graphically, histogram of timeseries with normal distribution in figure 10, and QQ-plot in figure11 of the time series are evaluated. From the histogram of FIN_growth_rate -variable in figure 10 it can be seen that the observations are not normally distributed since the histogram has excess tails and excess kurtosis around the mean compared the normal distribution. Besides that, it can be seen that the observations are skewed to left.

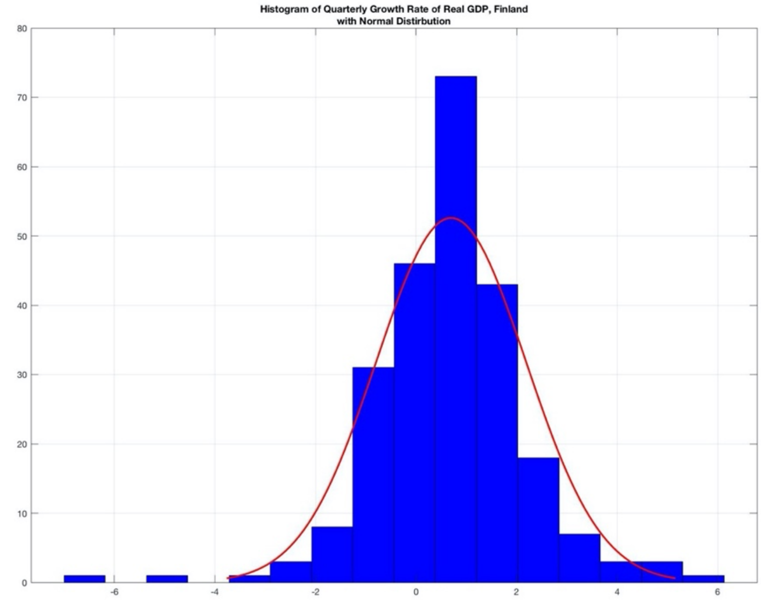


Figure 10. Histogram of FIN_growth_rate

QQ-plot below further strengthens the findings of leptokurtic characteristics in the FIN_growth_rate time series. From figure 11 below it can be seen that the tails of the distribution are bigger and doesn't follow linearity in the plot indicating not normally distributed time series.

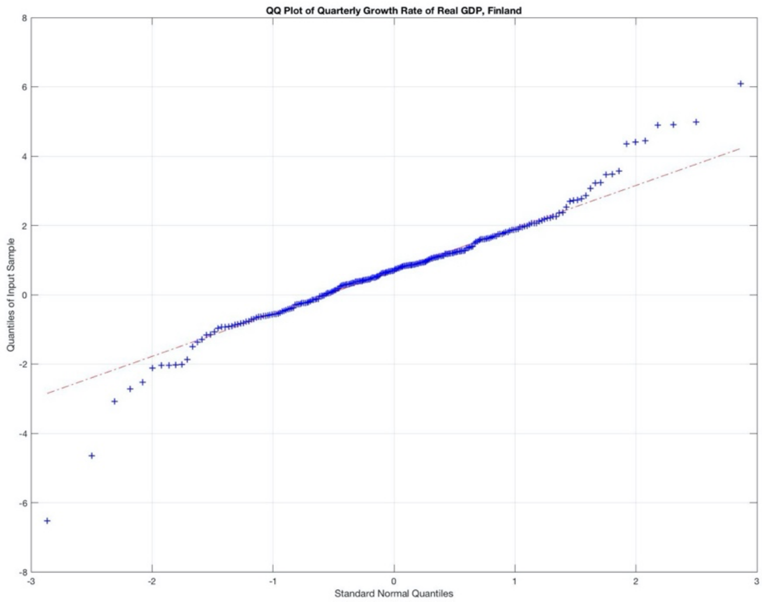


Figure 11. QQ -plot of FIN_growth_rate

To validate the results from graphical investigation of the time series, summary statistics and statistical tests are performed. Summary statistic are presented in table 7. Mean of the time series is 0.6967 while median and standard deviation being 0.7050 and 1.4865, respectively. Maximum value of the time series is 6.0945 and minimum value is -6.5216. Value of -0.1811 for skewness validates the result from graphical evaluation that the data of the distribution is skewed more to the left of the mean and value of 6.4104 for kurtosis the result that the distribution has excess kurtosis. These results support the findings that the data is not normally distributed since normally distributed data should have value of zero for skewness and value smaller than three for kurtosis.

Table 7. Summary statistic for FIN_growth_rate

Variable	**Mean**	**Median**	**Standard deviation**	**Max**	**Min**	**Skewness**	**Kurtosis**
FIN_growth_rate	0.6967	0.7050	1.4865	6.0945	-6.5216	-0.1811	6.4104

For verifying the results, Jarque-Bera test for normality is performed for the time series. Results of the test presented in table 8. Null hypothesis that FIN_growth_rate -variable is normally distributed is rejected with test statistic of 117.1325 and p-value of 0.001 with significance level of 0.050. Critical value for the test statistic is 5.7238. Rejection of the null hypothesis verifies that the time series data for FIN_growth_rate is not normally distributed and that the result is statistically significant.

Table 8. Result of Jarque-Bera test for FIN_growth_rate

Jarque-Bera Test for Normality				
H_0: FIN_growth_rate is normally distributed				
Null Rejected	P-Value	Test Statistic	Critical Value	Significance Level
TRUE	0.001	117.1325	5.7238	0.050

To use ARMA models for modelling conditional mean of time series data, the data must be stationary. To find out if the data is stationary, Augmented Dickey-Fuller (ADF) test with four lags is used for FIN_growth_rate. The value for lags is set to 4 because the used data is quarterly data. Null hypothesis of the test is that FIN_growth_rate contains a unit root. This null hypothesis is rejected with test statistic of -4.0968 and p-value of 0.001 with significance level of 0.050. Critical value for the test statistic is -1.9421. Rejection of the null hypothesis means that the data is stationary. Table 9 below present the statistics of ADF test.

Table 9.Result of ADF test for FIN_growth_rate

Augumented Dickey-Fuller Test					
H_0: FIN_growth_rate contains a unit root					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Significance Level
TRUE	0.001	-4.0968	-1.9421	4	0.050

To complement ADF test results KPSS test for stationarity with 4 lags is performed for FIN_growth_rate time series. Null hypothesis of the test is that FIN_growth_rate is trend stationary which fails to be rejected with test statistic of 0.0474 and p-value >0.100 with significance level of 0.050. The critical value for the test is 0.1460. The result of the test complements the result of ADF test that the time series is stationary. Table 10 presents the results of KPSS test.

Table 10. Result of KPSS test for FIN_growth_rate

KPSS Test for Stationarity					
H_0: FIN_growth_rate is trend stationary					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Signifigance Level
FALSE	>0.100	0.0474	0.1460	10	0.050

For next, FIN_growth_rate time series is tested for being random walk with Variance ratio test for random walk. Test is performed with 4 periods because of quarterly data. The null hypothesis of the test is that FIN_growth_rate is a random walk. This null hypothesis is rejected with test statistic of -3.6391 and p-value of <0.001 with significance level of 0.050. The critical value for the test is 1.9600. The result of the test is that FIN_growth_rate follows non-random walk. The results are presented in table 11.

Table 11. Result of Variance ratio test for FIN_growth_rate

Variance Ratio Test for Random Walk					
H_0: FIN_growth_rate is a random walk					
Null Rejected	P-Value	Test Statistic	Critical Value	Period	Signifigance Level
TRUE	<0.001	-3.6391	1.9600	4	0.050

4.3 Descriptive statistics of Quarterly Growth Rate of real GDP, Iceland

Growth rate of real GDP in Iceland is presented as variable named “ISL_growth_rate” and it is seasonally adjusted growth rate of real gross domestic product compared to previous

quarter. Sample size consists of 239 quarterly observations of growth rate percentage for time period from 1.4.1960 to 31.12.2019 and it is divided into two sub-samples; in-the-sample period consisting of 219 quarterly observations from 1.4.1960 to 31.12.2014 and out-of-sample period consisting of 20 quarterly observations from 1.1.2015 to 31.12.2019.

Graphical evaluation starts with investigating the time series of ISL_growth_rate -variable which is presented in figure 12 showing quarterly observations plotted as a time series for the whole sample period. The quarterly real GDP growth rate seems to fluctuate around its mean which is assumed to be between zero and two percent according to the plotted time series. From the 60's until mid 90's Iceland's economic growth has been stable besides one shock in economic growth in 1970. After that, the economic growth starts to fluctuate more probably due to "Nordic Tiger" economic prosperity period resulted from reformed market liberalization policies. Iceland suffered from the financial crisis similarly to other Nordic countries causing huge negative impact on economic growth rate leading to the national financial crisis ending up government taking control of the three major banks in Iceland. The European debt crisis had also negative impact on Iceland's economic growth in 2011.

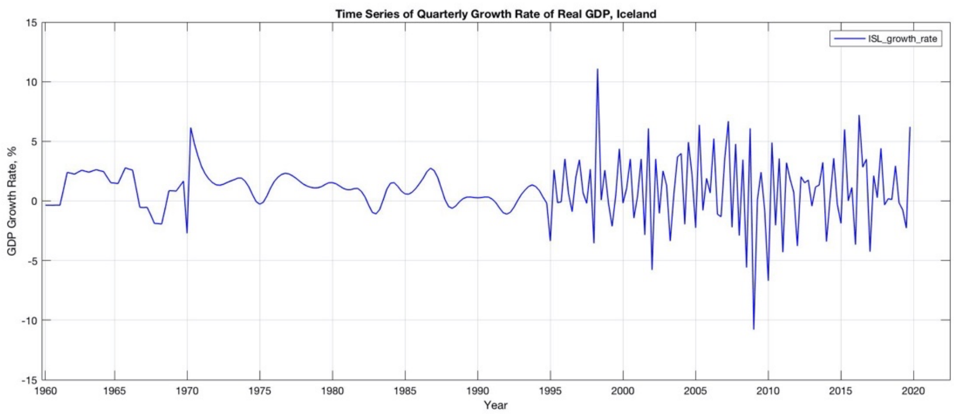


Figure 12. Time series of ISL_growth_rate

There are few alternating periods of high and low volatility in figure 12 which could imply volatility clustering indicating that the data has autocorrelation among observations since large deviations from the mean seem to be followed by large deviations of other sign from the mean. To investigate further possible autocorrelation in the data it is useful to check plotted autocorrelation function (ACF) which is presented in figure 13.

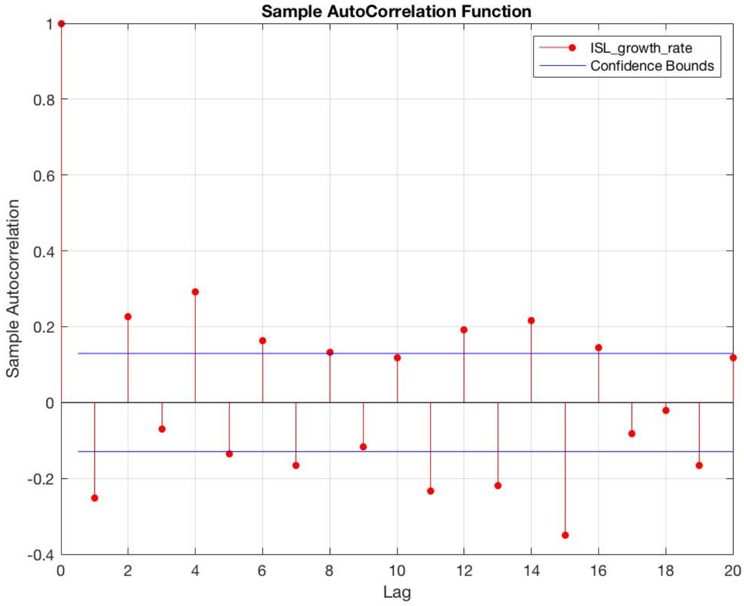


Figure 13. ACF -plot of ISL-Growth_rate

Figure 13 above indicates that there are positive and negative autocorrelations in the data until up to lag 19. Most of the autocorrelations above the 95% confidence bounds, which implies statistically significant autocorrelation differing from zero.

To analyze normality of the time series graphically, histogram of timeseries with normal distribution in figure 14, and QQ-plot in figure 15 of the time series are evaluated. From the histogram of ISL_growth_rate time series in figure 14 it can be seen that the observations are not normally distributed since the histogram has fatter tails and very high excess kurtosis around the mean compared the normal distribution. It can be also seen that the observations are skewed more to the left.

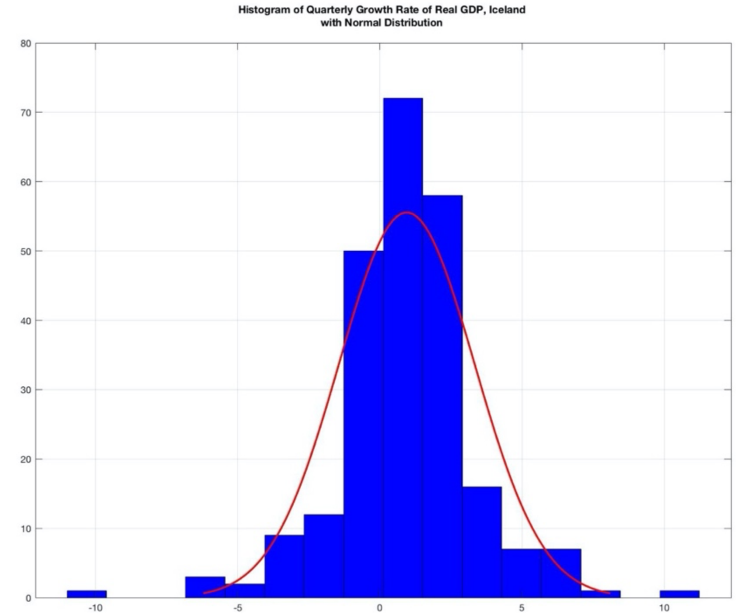


Figure 14. Histogram of ISL_growth_rate

The QQ-plot below shows clearly leptokurtic characteristics in the ISL_growth_rate time series especially for the tails of the distribution. The tails of the distribution are fatter and doesn't follow linearity in the plot indicating not normally distributed time series.

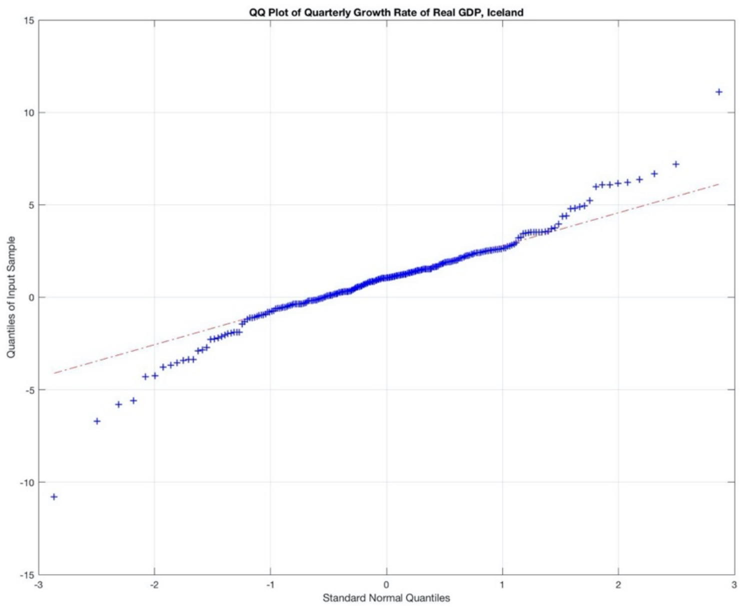


Figure 15. QQ -plot of ISL_growth_rate

To validate the results from graphical investigation of the time series, summary statistics and statistical tests are performed. Summary statistic are presented in table 14. Mean of the time series is 0.9478 while median and standard deviation being 1.0533 and 2.3860, respectively. Maximum value of the time series is 11.0915 and minimum value is -10.7929. Value of -0.3005 for skewness validates the insight from graphical evaluation that the data of the distribution is skewed more to the left of the mean and value of 6.7738 for kurtosis the result that the distribution has serious excess kurtosis. These results support the findings that the data is not normally distributed since normally distributed data should have value of zero for skewness and value smaller than three for kurtosis.

Table 12. Summary statistics of ISL_growth_rate

Variable	Mean	Median	Standard deviation	Max	Min	Skewness	Kurtosis
ISL_growth_rate	0.9478	1.0533	2.3860	11.0915	-10.7929	-0.3005	6.7738

For verifying the results, Jarque-Bera test for normality is performed for the time series. Results of the test presented in table 13. Null hypothesis that ISL_growth_rate -variable is normally distributed is rejected with test statistic of 145.4193 and p-value of 0.001 with significance level of 0.050. Critical value for the test statistic is 5.7238. Rejection of the null hypothesis verifies that the time series data for ISL_growth_rate is not normally distributed and that the result is statistically significant.

Table 13. Result of Jarque-Bera test for ISL_growth_rate

Jarque-Bera Test for Normality				
H_0: ISL_growth_rate is normally distributed				
Null Rejected	P-Value	Test Statistic	Critical Value	Significance Level
TRUE	0.001	145.4193	5.7238	0.050

In order to utilize ARMA models for modelling conditional mean of time series data, the data has to be stationary. To find out if the data is stationary, Augmented Dickey-Fuller (ADF) test with 4 lags is used for ISL_growth_rate. Yet again, we use 4 lags for this test since the data used is quarterly data. Null hypothesis of the test is that ISL_growth_rate contains a unit root. This null hypothesis is rejected with test statistic of -3.6334 and p-value of 0.001 with significance level of 0.050. Critical value for the test statistic is -1.9421. Rejection of the null hypothesis means that the data is stationary. Table 14 below present the statistics of ADF test.

Table 14. Results of ADF test for ISL_growth_rate

Augumented Dickey-Fuller Test					
H_0l: ISL_growth_rate contains a unit root					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Significance Level
TRUE	0.001	-3.6334	-1.9421	4	0.050

To complement ADF test results KPSS test for stationarity with 4 lags is performed for ISL_growth_rate time series. Null hypothesis of the test is that ISL_growth_rate is trend stationary which fails to be rejected with test statistic of 0.0580 and p-value >0.100 with significance level of 0.050. The critical value for the test is 0.1460. The result of the test complements the result of ADF test that the time series is stationary. Table 15 presents the results of KPSS test.

Table 15. Result of KPSS test for ISL_growth_rate

KPSS Test for Stationarity					
H_0: ISL_growth_rate is trend stationary					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Signifigance Level
FALSE	>0.100	0.0580	0.1460	4	0.050

Next, ISL_growth_rate time series is tested for being random walk with Variance ratio test for random walk. Number of periods used in the test is 4 because the data used is quarterly data. The null hypothesis of the test is that ISL_growth_rate is a random walk. This null hypothesis is rejected with test statistic of -3.8489 and p-value of <0.001 with significance level of 0.050. The critical value for the test is 1.9600. The result of the test is that ISL_growth_rate follows non-random walk. The results are presented in table 16.

Table 16. Result of Variance ratio test for ISL_growth_rate

Variance Ratio Test for Random Walk					
H_0: ISL_growth_rate is a random walk					
Null Rejected	P-Value	Test Statistic	Critical Value	Period	Signifigance Level
TRUE	<0.001	-3.8489	1.9600	4	0.050

4.4 Descriptive statistics of Quarterly Growth Rate of real GDP, Norway

Growth rate of real GDP in Norway is presented as variable named “NOR_growth_rate” and it is seasonally adjusted growth rate of real gross domestic product compared to previous

quarter. Sample size consists of 239 quarterly observations of growth rate percentage for time period from 1.4.1960 to 31.12.2019 and it is divided into two sub-samples; in-the-sample period consisting of 219 quarterly observations from 1.4.1960 to 31.12.2014 and out-of-sample period consisting of 20 quarterly observations from 1.1.2015 to 31.12.2029.

Graphical evaluation starts with investigating the time series of NOR_growth_rate -variable which is presented in figure 16 showing quarterly observations plotted as a time series for the whole sample period. The quarterly real GDP growth rate seems to fluctuate around its mean which is assumed to be between zero and one percent according to the plotted time series. From the 80's Norway's economic growth starts to fluctuate more than previously. The burst of the tech-bubble in 2000, and global financial crisis in 2008 followed by European debt crisis in 2011 has all had negative impact on economic growth in Norway.

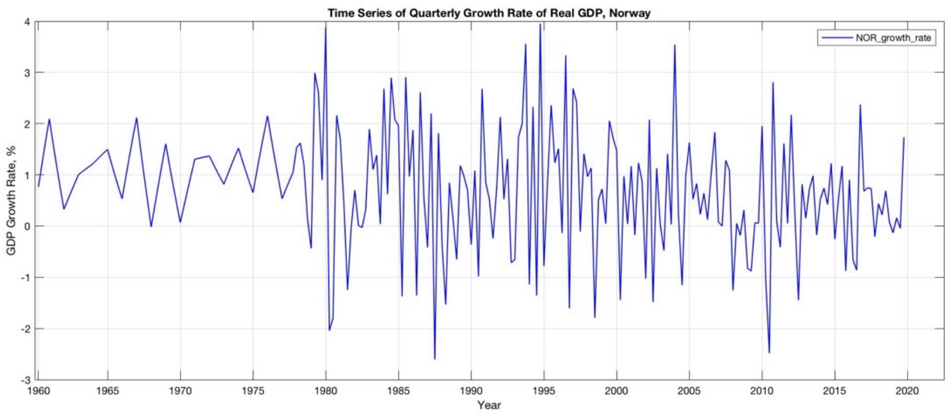


Figure 16. Time series of NOR_growth_rate

There are clear alternating periods of high and low volatility in figure 16 which imply volatility clustering indicating that the data has autocorrelation among observations since large deviations from the mean seem to be followed by large deviations of other sign from the mean. To investigate further possible autocorrelation in the data it is useful to check plotted autocorrelation function (ACF) which is presented in figure 17.

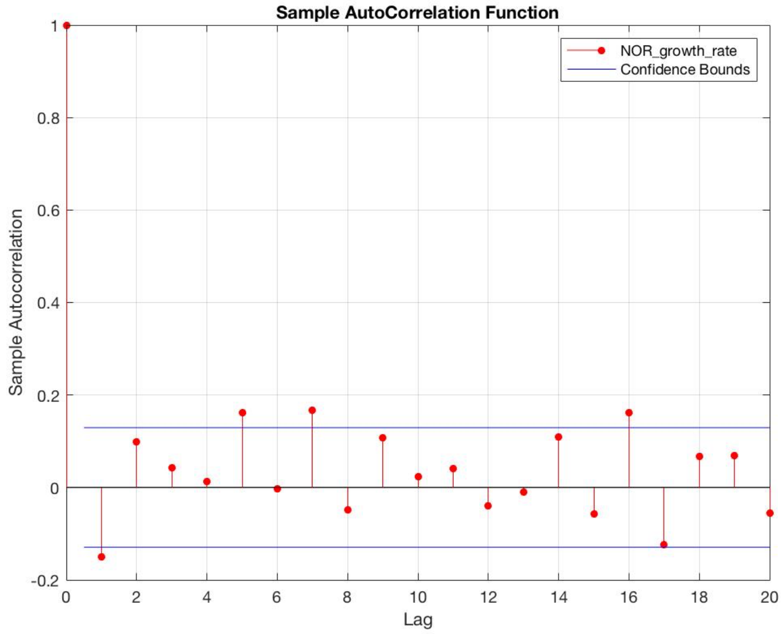


Figure 17. ACF -plot of NOR_growth_rate

Figure 17 above indicates that there are positive and negative autocorrelations in the data until up to lag 16. Lags 1, 5,7 and 16 seem to have autocorrelations above the 95% confidence bounds, which implies statistically significant autocorrelation differing from zero.

In figure 18 histogram of time series with normal distribution is plotted and figure 19 shows QQ-plot of the time series to evaluate normality of the time series. From the histogram of NOR_growth_rate time series it can be seen that the observations are not normally distributed since the histogram has fatter tails and high excess kurtosis around the mean compared the normal distribution. It can be also seen that the observations are skewed more to the left.

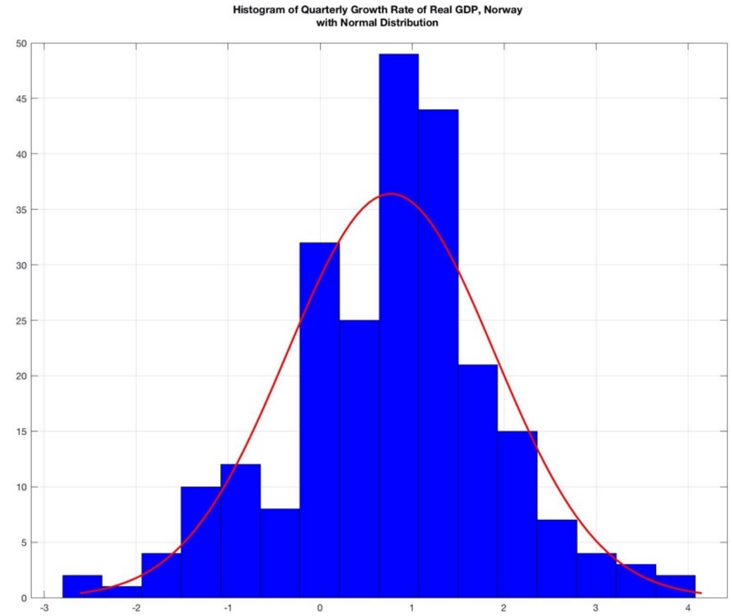


Figure 18. Histogram of NOR_growth_rate

The QQ-plot below shows clearly leptokurtic characteristics in the NOR_growth_rate time series especially for the tails of the distribution. The tails of the distribution are fatter and doesn’t follow linearity in the plot indicating not normally distributed time series.

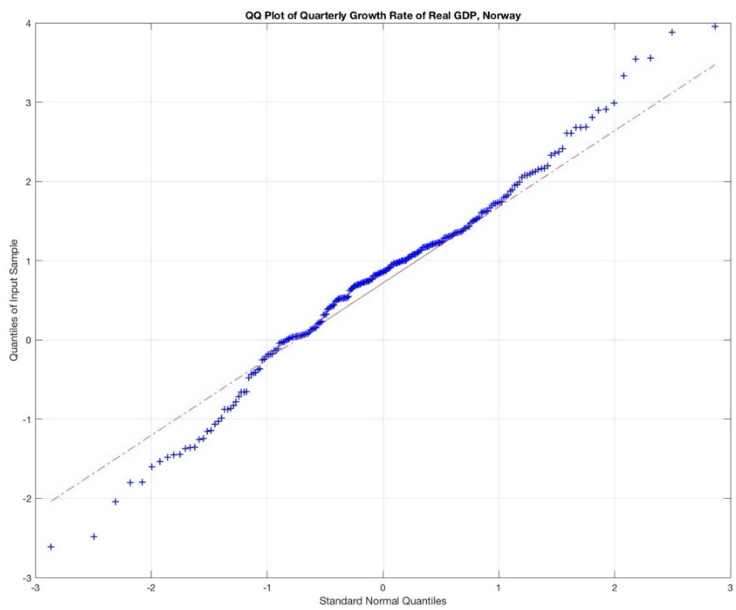


Figure 19. QQ -plot of NOR_growth_rate

To validate the results from graphical investigation of the time series, summary statistics and statistical tests are performed. Summary statistic are presented in table 17. Mean of the time series is 0.7688 while median and standard deviation being 0.8575 and 1.1262, respectively. Maximum value of the time series is 3.9520 and minimum value is -2.6074. Value of -0.1813 for skewness validates the insight from graphical evaluation that the data of the distribution is skewed slightly more to the left of the mean and value of 3.5321 for kurtosis the result that the distribution has some slight excess kurtosis. These results support the findings that the data is not normally distributed since normally distributed data should have value of zero for skewness and value smaller than three for kurtosis.

Table 17. Summary statistic for NOR_growth_rate

Variable	**Mean**	**Median**	**Standard deviation**	**Max**	**Min**	**Skewness**	**Kurtosis**
NOR_growth_rate	0.7688	0.8575	1.1262	3.9520	-2.6074	-0.1813	3.5321

Table 20: Summary statistic for NOR_growth_rate.

For verifying the results, Jarque-Bera test for normality is performed for the time series. Results of the test presented in table 18. Null hypothesis that NOR_growth_rate time series is normally distributed fails to be rejected with test statistic of 4.1286 and p-value of 0.099 with significance level of 0.050. Critical value for the test statistic is 5.7238. The fact, that the null hypothesis fails to be rejected indicates that the time series data for NOR_growth_rate is actually normally distributed against the results of graphical evaluation and summary statistics.

Table 18. Result of Jarque-Bera test for NOR_growth_rate

Jarque-Bera Test for Normality				
H_0: NOR_growth_rate is normally distributed				
Null Rejected	P-Value	Test Statistic	Critical Value	Significance Level
FALSE	0.099	4.1286	5.7238	0.050

To use ARMA -models for modelling conditional mean of time series data, the data has to be stationary. To find out if the data is stationary, Augmented Dickey-Fuller (ADF) test is performed for NOR_growth_rate time series. Null hypothesis of the test is that NOR_growth_rate contains a unit root. General assumption for ADF test using quarterly data is that 4 lags should be enough for the test. The null hypothesis is rejected with test statistic of -3.0005 and p-value of 0.004 and significance level of 0.050. Table 19 below present the statistics of ADF tests.

Table 19. Result of ADF test for NOR_growth_rate

Augumented Dickey-Fuller Test					
H_0l: NOR_growth_rate contains a unit root					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Significance Level
TRUE	0.004	-3.0005	-1.9421	4	0.050

To complement ADF test results, KPSS test for stationarity with 4 lags is performed for NOR_growth_rate time series. Null hypothesis of the test is that NOR_growth_rate is trend stationary which fails to be rejected with test statistic of 0.0418 and p-value >0.100 with significance level of 0.050. The critical value for the test is 0.1460. The result of the test complements the result of ADF test that the time series is stationary. Table 20 presents the results of KPSS test.

Table 20. Result of KPSS test for NOR_growth_rate

KPSS Test for Stationarity					
H_0: NOR_growth_rate is trend stationary					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Signifigance Level
FALSE	>0.100	0.0418	0.1460	4	0.050

Next, Variance ratio test for random walk with 4 periods is performed for the quarterly time series NOR_growth_rate. The null hypothesis of the test is that NOR_growth_rate is a random walk. This null hypothesis is rejected with test statistic of -4.2422 and p-value >0.001 with significance level of 0.050. The critical value for the test is 1.9600. The result of the test is that NOR_growth_rate follows non-random walk. The results are presented in table 21.

Table 21. Result of Variance ratio test for NOR_growth_rate.

Variance Ratio Test for Random Walk					
H_0: NOR_growth_rate is a random walk					
Null Rejected	P-Value	Test Statistic	Critical Value	Period	Signifigance Level
TRUE	>0.001	-4.2422	1.9600	4	0.050

4.5 Descriptive statistics of Quarterly Growth Rate of real GDP, Sweden

Growth rate of real GDP in Sweden is presented as variable named “SWE_growth_rate” and it is seasonally adjusted growth rate of real gross domestic product compared to previous quarter. Sample size consists of 239 quarterly observations of growth rate percentage for time period from 1.4.1960 to 31.12.2019 and it is divided into two sub-samples; in-the-sample period consisting of 219 quarterly observations from 1.4.1960 to 31.12.2014 and out-of-sample period consisting of 20 quarterly observations from 1.1.2015 to 31.12.2019.

Graphical evaluation starts with investigating the time series of SWE_growth_rate -variable which is presented in figure 20 showing quarterly observations plotted as a time series for the whole sample period. The quarterly real GDP growth rate seems to fluctuate around its mean which is assumed to be between zero and two percent according to the plotted time series. Unlike other Nordic countries, Sweden’s economic growth rate seems to fluctuate more at the beginning of the sample period and seem to become steadier towards the end of the sample period. Global financial crisis in 2008 followed by European debt crisis in 2011 has also had negative impact on economic growth in Sweden.

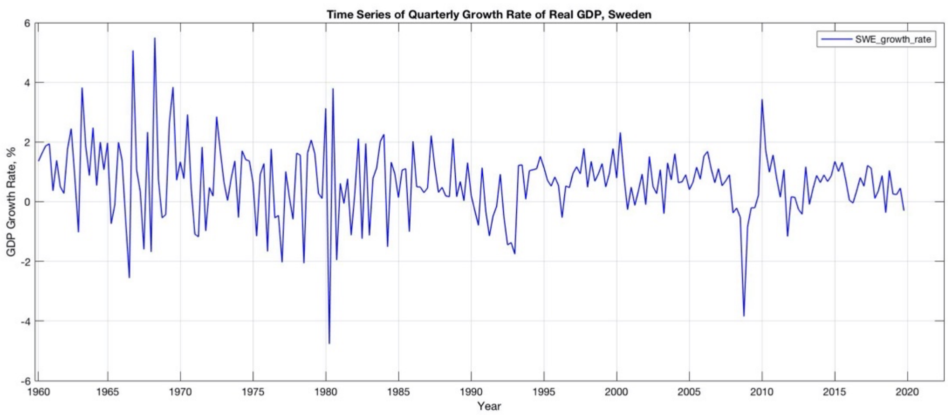


Figure 20. Time series of SWE_growth_rate

There are clear alternating periods of high and low volatility in figure 20 which imply volatility clustering indicating that the data has autocorrelation among observations since large deviations from the mean seem to be followed by large deviations of other sign from the mean. To investigate further possible autocorrelation in the data it is useful to check plotted autocorrelation function (ACF) which is presented in figure 21.

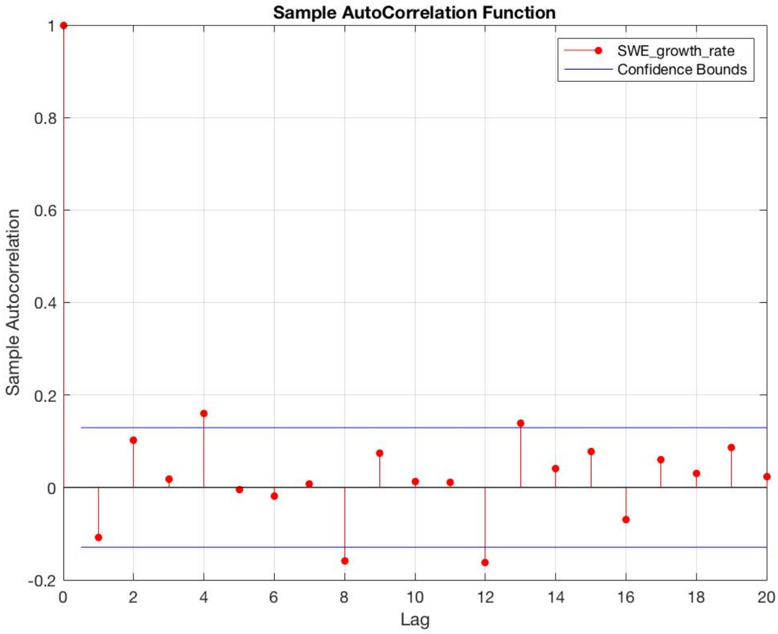


Figure 21. ACF -plot of SWE_growth_rate

Figure 21 above indicates that there are positive and negative autocorrelations in the data until up to lag 13. Lags 4,8,12 and 13 seem to have autocorrelations above the 95% confidence bounds, which implies statistically significant autocorrelation differing from zero.

In figure 22 histogram of time series with normal distribution is plotted and figure 23 shows QQ-plot of the time series to evaluate normality of the time series. From the histogram of SWE_growth_rate time series it can be seen that the observations are not normally distributed since the histogram has fatter tails and high excess kurtosis around the mean compared the normal distribution. It can be also seen that the observations are skewed more to the left.

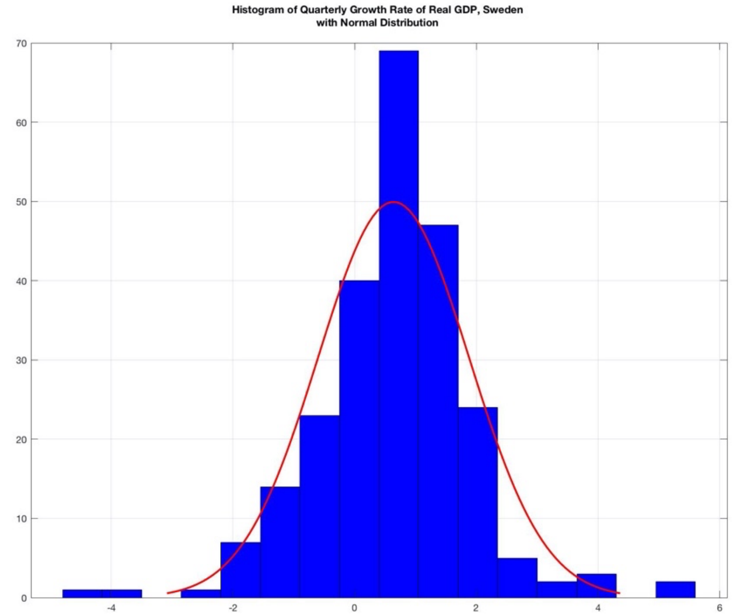


Figure 22. Histogram of SWE_growth_rate

The QQ-plot below shows clearly leptokurtic characteristics in the SWE_growth_rate time series especially for the tails of the distribution. The tails of the distribution are larger and doesn’t follow linearity in the plot indicating not normally distributed time series.

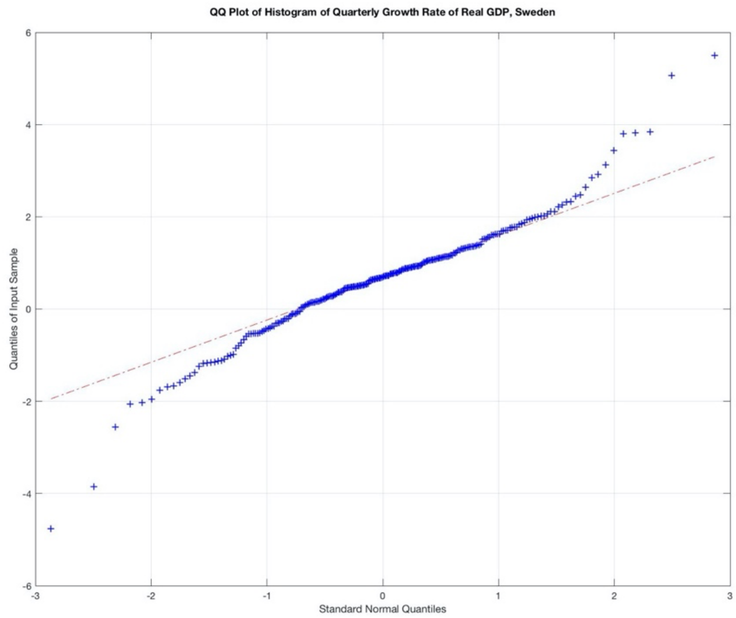


Figure 23. QQ -plot of SWE_growth_rate

To validate the results from graphical investigation of the time series, summary statistics and statistical tests are performed. Summary statistic are presented in table 22. Mean of the time series is 0.6389 while median and standard deviation being 0.6981 and 1.2411, respectively. Maximum value of the time series is 5.4955 and minimum value is -4.7673. Value of -0.1350 for skewness validates the insight from graphical evaluation that the data of the distribution is skewed a bit more to the left of the mean and value of 5.9931 for kurtosis the result that the distribution has excess kurtosis. These results support the findings that the data is not normally distributed since normally distributed data should have value of zero for skewness and value smaller than three for kurtosis.

Table 22. Summary statistics for SWE_growth_rate

Variable	**Mean**	**Median**	**Standard deviation**	**Max**	**Min**	**Skewness**	**Kurtosis**
SWE_growth_rate	0.6389	0.6981	1.2411	5.4955	-4.7673	-0.1350	5.9931

For verifying the results, Jarque-Bera test for normality is performed for the time series. Results of the test presented in table 23. Null hypothesis that SWE_growth_rate time series is normally distributed is rejected with test statistic of 89.9366 and p-value <0.001 with significance level of 0.050. Critical value for the test statistic is 5.7238. Rejection of the null hypothesis verifies that the time series data for SWE_growth_rate is not normally distributed and that the result is statistically significant.

Table 23. Result of Jarque-Bera test for SWE_growth_rate

Jarque-Bera Test for Normality				
H_0: SWE_growth_rate is normally distributed				
Null Rejected	P-Value	Test Statistic	Critical Value	Significance Level
TRUE	<0.001	89.9366	5.7238	0.050

For using ARMA models for modelling conditional mean of time series data, the data has to be stationary. To find out if the data is stationary, Augmented Dickey-Fuller (ADF) test with 4 lags is used for SWE_growth_rate, because of quarterly data. Null hypothesis of the test is that SWE_growth_rate contains a unit root. This null hypothesis is rejected with test statistic of -3.7240 and p-value of 0.001 with significance level of 0.050. Critical value for the test statistic is -1.9421. Rejection of the null hypothesis means that the data is stationary. Table 24 below present the statistics of ADF test.

Table 24. Result of ADF test for SWE_growth_rate

Augumented Dickey-Fuller Test					
H_0: SWE_growth_rate contains a unit root					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Significance Level
TRUE	0.001	-3.7240	-1.9421	4	0.050

To verify ADF test results, KPSS test for stationarity with 4 lags is performed for SWE_growth_rate time series. Null hypothesis of the test is that SWE_growth_rate is trend stationary which fails to be rejected with test statistic of 0.1293 and p-value of 0.081 with significance level of 0.050. The critical value for the test is 0.1460. The result of the test complements the result of ADF test that the time series is stationary. Table 25 presents the results of KPSS test.

Table 25. Result of KPSS test for SWE_growth_rate

KPSS Test for Stationarity					
H_0: SWE_growth_rate is trend stationary					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Signifigance Level
FALSE	0.081	0.1293	0.1460	4	0.050

Next, SWE_growth_rate time series is tested for being random walk with Variance ratio test for random walk. In the test the number of periods used is 4 due to quarterly data. The null hypothesis of the test is that SWE_growth_rate is a random walk. This null hypothesis is rejected with test statistic of -3.4740 and p-value of <0.001 with significance level of 0.050. The critical value for the test is 1.9600. The result of the test is that SWE_growth_rate follows non-random walk. The results are presented in table 26.

Table 26. Result of Variance ratio test for SWE_growth_rate

Variance Ratio Test for Random Walk				
H_0: SWE_growth_rate is a random walk				
Null Rejected	P-Value	Test Statistic	Critical Value	Signifigance Level
TRUE	<0.001	-3.4740	1.9600	0.050

4.6 Forecasting procedure

The aim of this thesis is to examine forecasting performance between conditional mean and volatility forecasting ARMA-GARCH -models and nonlinear autoregressive neural network

NAR models. For this purpose, the sample data is divided into in-the-sample and out-of-sample sub-samples. Forecasting models are fitted with in-the-sample data and forecasted values are compared with out-of-sample data to obtain the loss function values.

Forecasting procedure and evaluation of using ARMA-GARCH -models is done as follows:

1. Defined time series data is imported into MATLAB.
2. Data is divided into two sub-samples: in-the-sample sub-sample consisting of first 219 observations and out-of-sample sub-sample of consisting last 20 observations.
3. AIC matrix is constructed to obtain the values of the information criteria AIC for any combination of order 1 to 9 of AR(p) and MA(q) models of the defined time series' in-the-sample data and the ARMA -model with lowest AIC value is chosen to be fitted for that time series' in-the-sample data.
4. Performance of the model is graphically evaluated by plotting the ACF and PACF of the model's residuals and to detect possible autocorrelation among the residuals.
5. Residuals are tested with Ljung-Box test for autocorrelation to verify the result from graphical evaluation.
6. Engle's ARCH test is used for residuals to see if the residual exhibit autoregressive conditional heteroscedastic effects and if GARCH -models can be implemented to ARMA -model.
7. Normality of the residuals is tested with Jarque-Bera test to see if it would be beneficial to use also Student's t-distribution in GARCH -models.
8. Hybrid ARMA-GARCH -models are constructed and fitted to the in-the-sample data of the defined time series.
9. Values for next 20 time-steps are predicted from in-the-sample data.
10. MAE and MSE loss functions are calculated on the basis of 20 out-of-sample real values and the next 20 predicted values with ARMA-GARCH -models from in-the-sample data
11. The procedure is repeated for all of the time series.

Forecasting procedure and evaluation of using NAR -models is done as follows:

1. Defined time series data is imported into MATLAB.
2. Data is divided into two sub-samples: in-the-sample sub-sample consisting of first 219 observations and out-of-sample sub-sample of consisting last 20 observations.

3. In-the-sample sub-sample is divided into training set, validation set and test set so that 75% of the observations are included in the training set, 15% of the observations are included in the validation set and 15% of the observations are included in the test set.
4. Number of hidden delay neurons is considered to be 2 according to results of research by Zhang et.al. (1998) who show with simulation that best network structure for NAR corresponds to a hidden layer with maximum of two neurons.
5. To construct the rest of the appropriate architecture for NAR with different training algorithms, the value for delay parameter is decided on the basis of calculating average MSE for training set from 10 training times with delays $d = \{1; 2; 3,4; 8; 12; 16; 20; 24\}$ to correspond for observations on timespan of 1 quarter, 2 quarters, 3 quarters, 4 quarters, 8 quarters, 12 quarters, 16 quarters, 20 quarters and 24 quarters.
6. The result of delay parameter estimation for each of the training algorithms is used in final architecture of NAR using said training algorithm.
7. NAR with unique architecture for each training algorithm is trained using in-the-sample data in open loop. As a result, 3 different NARs are obtained
8. NARs are transformed to closed loop and used for predicting the values for next 20 time-steps on the basis of in-the-sample data.
9. MAE and MSE loss functions are calculated for NARs with different training algorithms on the basis of 20 out-of-sample real values and the next 20 predicted values with NARs.
10. The procedure is repeated for all of the time series

It is noteworthy to mention, that separate best-fitting models need to be constructed for each of the time series separately, since one model to be fitted for all of the time series would not be able to capture the characteristics of the different time series and could lead to badly performing models which are not suitable for a certain time series. Therefore, it is crucial to find the best performing structure for the models to represent the certain time series separately.

5 EMPIRICAL RESULTS AND ANALYSIS

In this chapter the results of empirical analysis of the research are presented separately for each of the time series. The results are reported separately to make clear the different statistical properties of the time series and to avoid misinterpretation of the results.

5.1 Results for Quarterly Growth Rate of real GDP, Denmark

5.1.1 Results of conditional mean and variance model estimation

The best conditional mean model for modelling quarterly growth rate of real GDP in Denmark turn out to be ARMA(8,8) with AIC of 626.314. The complete constructed matrix of the values of information criteria AIC for any combination of order 1 to 9 of AR(p) and MA(q) models can be seen in appendix 1. To evaluate the performance of the model, it is fitted to the in-the-sample data. To check how the model performs graphically ACF and PACF of the model is evaluated in figure 24. It can be seen that the residuals of the model exhibit no serial autocorrelation.

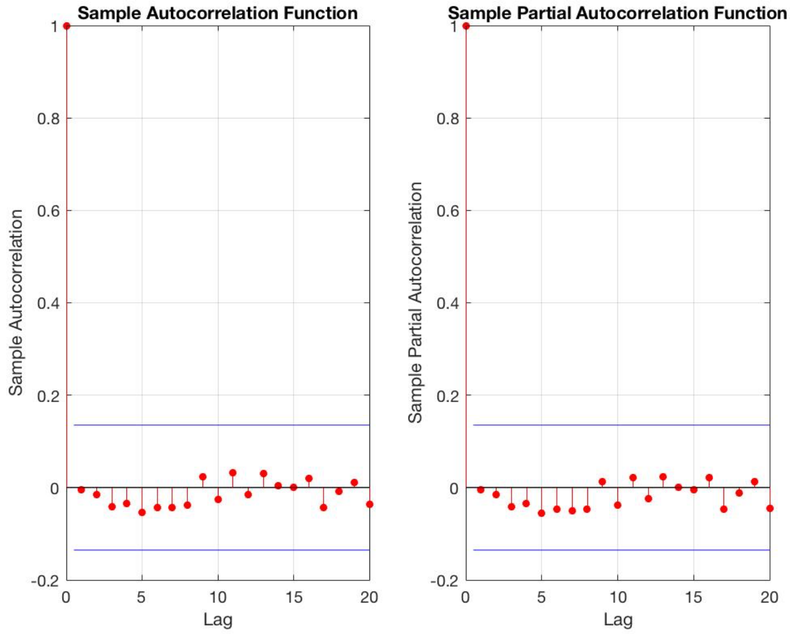


Figure 24. ACF and PACF -plots of fitted ARMA(8,8) -model for DEN_growth_rate

To verify this Ljung-Box test for autocorrelation is used. The result of the Ljung-Box test is presented in Table 27 below. The null hypothesis of Ljung-Box test that residuals exhibit no serial autocorrelation fails to be rejected with test statistic of 4.2285 and p-value of 0.999 with significance level of 0.050. The critical value of the test is 31.4140.

Table 27. Result of Ljung-Box test for ARMA(8,8) -model residuals

Ljung-Box Test for Autocorrelation				
H_0: Residuals exhibit no serial autocorrelation				
Null Rejected	P-Value	Test Statistic	Critical Value	Signifigance Level
FALSE	0.999	4.2285	31.4140	0.050

To detected whether the residuals exhibit autoregressive conditional heteroscedastic effects, Engle's ARCH test is used. The null hypothesis of the test is that residuals exhibit no ARCH effects. The number of lags used in the test is 4 because of quarterly data. The null hypothesis is rejected with test statistic of 23.3810 and p-value of <0.001 with significance level of 0.050. The critical value of the test is 9.4877. The result of the test is presented in table 28 below. The test result indicates that there is autoregressive conditional heteroscedasticity among the residuals and therefore GARCH -models can be applied for the time series models.

Table 28. Result of Engle's ARCH test for ARMA(8,8) -model residuals

Engle's ARCH Test for Heteroscedasticity					
H_0: Residuals exhibits no ARCH effects					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Signifigance Level
TRUE	<0.001	23.3810	9.4877	4	0.050

To test if the residuals of the model are normally distributed Jarque-Bera test for normality is used. The null hypothesis that residuals are normally distributed is rejected with test statistic of 9.4365 and p-value of 0.017 with significance level of 0.050. The critical value of the test is 5.7022. The test result indicates that it would be beneficial to also use alternatively t-distribution in conditional volatility modelling with GARCH -models. The result of the test is presented in table 29 below.

Table 29. Result of Jarque-Bera test for ARMA(8,8) -model residuals

Jarque-Bera Test for Normality				
H_0: Residuals are normally distributed				
Null Rejected	P-Value	Test Statistic	Critical Value	Signifigance Level
TRUE	0.017	9.4365	5.7022	0.050

5.1.2 Result of NAR architecture estimation

From the Levenberg-Marquardt performance matrix in appendix 2 it can be seen that the best NAR test performance with LM training algorithm is achieved with delay parameter value of 1 when the average MSE of NAR with LM training algorithm for the in-the-sample test sample is 1.1541. From the performance matrix in appendix 3 it is seen that best test performance with SCG training algorithm is achieved with delay parameter value of 3 when average MSE of the in-the-sample test sample is 1.1339. For BR training algorithm the best test performance is achieved when the delay parameter value is set to 16 according to the performance matrix in appendix 4. With delay parameter value of 16 for BR algorithm, when the average MSE of NAR for the in-the-sample test sample is 1.0222. The complete performance results for delay -parameter estimation of LM, SCG and BR training algorithms with defined delays is presented in the appendices 2, 3 and 4, respectively. Figure 25 presents the different values of test performance for NAR with different training algorithms on defined delays measured with MSE of the in-the-sample test sample. Value of hidden layer neurons is set to 2 as mentioned previously in the chapter of regarding model building and forecasting procedure.

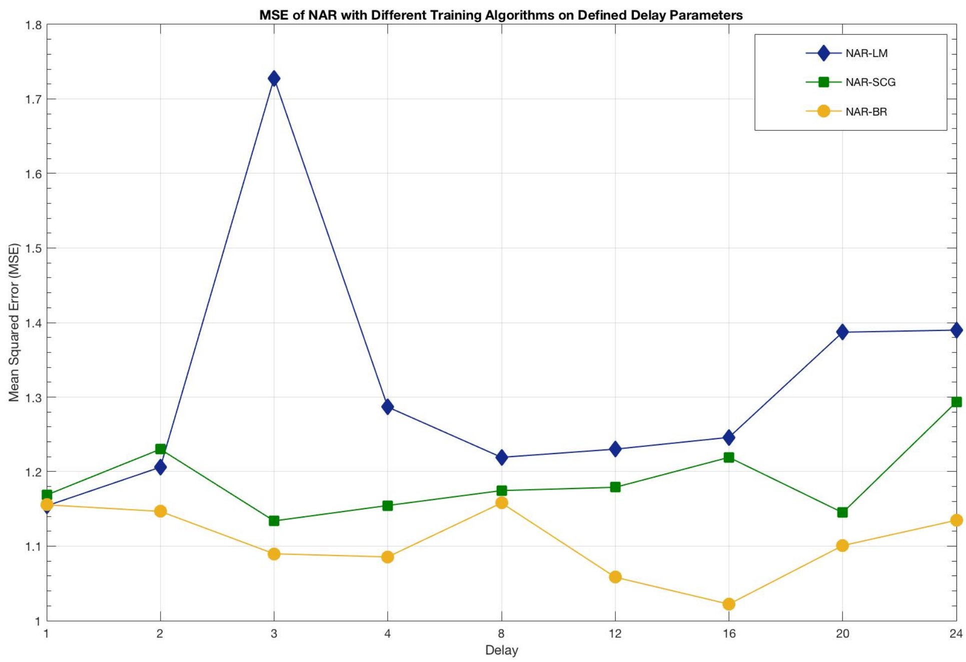


Figure 25. MSE of defined delay parameters for NARs; DEN_growth_rate

As the result of performance evaluation, the right architecture for NAR with LM training algorithm is decided to be with 2 hidden layer neurons and with 1 delay because this architecture produces the best performance for the network with lowest test sample MSE. The architecture of open loop NAR with LM training algorithm and closed loop NAR with LM training algorithm with these parameters are presented in appendix 4.

The result of performance evaluation for SCG training algorithm indicates that the best NAR model performance with SCG training algorithm is achieved when the delay parameter is set to 3 with 2 hidden layer neurons. This is decided to be the NAR architecture for SCG training algorithm since it produces the lowest test sample MSE. The architecture of open loop NAR with SCG training algorithm and closed loop NAR with SCG training algorithm with these parameters are presented in appendix 4.

For BR training algorithm, the best NAR performance is achieved with delay parameter value of 16 with 2 hidden layer neurons, which produces the lowest test sample MSE. For modelling and forecasting, this is decided to be the NAR architecture with BR training algorithm. The architecture of open loop NAR with BR training algorithm and closed loop NAR with BR training algorithm with these parameters are presented in appendix 4.

5.1.3 Results of model performance and forecasting

The results of forecasting performance and ranking of the forecasting models for quarterly growth rate of real GDP in Denmark are presented in Table 30. The results implicates that the most accurate forecast is provided by NAR-LM among the models when loss function MAE is used for evaluating the forecasts. MAE for this model is 0.2930. The second-best model is ARMA(8,8)-EGARCH(1,1) with normal distribution with MAE of 0.3150. Third and fourth best performing models are NAR-BR and NAR-SCG with MAEs of 0.3275 and 0.3339, respectively. As the results show, these four models outperform the rest of the models which are ARMA-GARCH models with normal and Student' t-distribution. This result is inconclusive since the errors of the residuals from conditional mean modelling had leptokurtic effects and introducing alternative t-distribution to GARCH models seem not to produce more accurate forecast than with normal distribution. It is notable that the NAR models outperform ARMA-GARCH models when interpreting performance results as a whole according to MAE.

When MSE is used as a loss function the result switches the other way around making ARMA(8,8)-EGARCH(1,1) the best performing model with MSE of 0.1496 and putting NAR-LM in second place with MSE of 0.1551. Also, NAR-SCG and NAR-BR switch positions in the ranking with MSEs of 0.1662 and 0.1668, respectively. The differences between these four models according to MSE are more distinct than with MAE and the differences are distinct especially between the first best four models and the rest of them. Most of the hybrid ARMA-GARCH conditional mean and volatility models seem to have again worse performance than NAR models when measured with loss function MSE.

Table 30. Model performance ranking for DEN_growth_rate

Model	MAE	Ranking	Model	MSE	Ranking
NAR-LM	0.2930	1	ARMA(8,8)-EGARCH(1,1) normal	0.1496	1
ARMA(8,8)-EGARCH(1,1) normal	0.3150	2	NAR-LM	0.1551	2
NAR-BR	0.3275	3	NAR-SCG	0.1662	3
NAR-SCG	0.3339	4	NAR-BR	0.1668	4
ARMA(8,8)-GJR-GARCH(1,1) Student-t	0.3965	5	ARMA(8,8)-GJR-GARCH(1,1) Student-t	0.2335	5
ARMA(8,8)-GJR-GARCH(1,1) normal	0.3987	6	ARMA(8,8)-GJR-GARCH(1,1) normal	0.2359	6
ARMA(8,8)-GARCH(1,1) normal	0.4346	7	ARMA(8,8)-GARCH(1,1) Student-t	0.2631	7
ARMA(8,8)-GARCH(1,1) Student-t	0,4347	8	ARMA(8,8)-GARCH(1,1) normal	0,2637	8
ARMA(8,8)-EGARCH(1,1) Student-t	0,5644	9	ARMA(8,8)-EGARCH(1,1) Student-t	0,4406	9

5.2 Results for Quarterly Growth Rate of real GDP, Finland

5.2.1 Results of conditional mean and variance model estimation

The best conditional mean model for modelling quarterly growth rate of real GDP in Finland found out to be ARMA(5,8) with AIC of 790.651. The complete constructed matrix of the values of information criteria AIC for any combination of order 1 to 9 of AR(p) and MA(q) models can be seen in appendix 1. To evaluate the performance of the model, it is fitted to the in-the-sample data. To check how the model performs graphically ACF and PACF of the model is evaluated in figure26. It can be seen that the residuals of the model exhibit no serial autocorrelation.

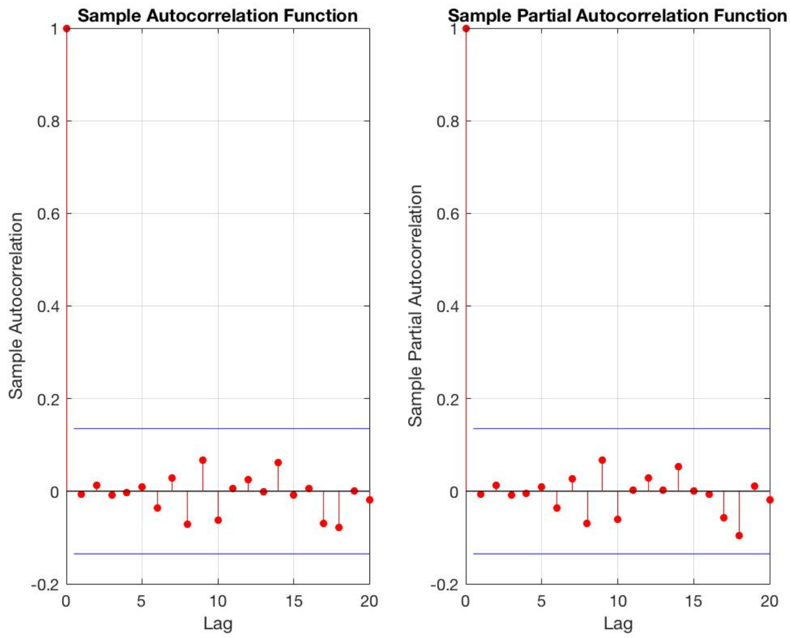


Figure 26. ACF and PACF -plots of fitted ARMA(5,8) -model for FIN_growth_rate

To verify this Ljung-Box test for autocorrelation is used. The result of the Ljung-Box test is presented in table below. The null hypothesis of Ljung-Box test that residuals exhibit no serial autocorrelation fails to be rejected with test statistic of 7.4591 and p-value of 0.995 with significance level of 0.050. The critical value of the test is 31.4140.

Table 31. Result of Ljung-Box test for ARMA(5,8) -model residuals

Ljung-Box Test for Autocorrelation				
H_0: Residuals exhibit no serial autocorrelation				
Null Rejected	P-Value	Test Statistic	Critical Value	Signifigance Level
FALSE	0.995	7.4591	31.4140	0.050

To detected whether the residuals exhibit autoregressive conditional heteroscedastic effects, Engle's ARCH test is used. The null hypothesis of the test is that residuals exhibit no ARCH effects. The number of lags used in the test is 4 because of quarterly data. The null hypothesis fails to be rejected with test statistic of 5.4465 and p-value of 0.245 with significance level of 0.050. The critical value of the test is 9.4877. The result of the test is presented in table 32 below. The test result indicates that there is no autoregressive conditional heteroscedasticity among the residuals and therefore GARCH -models cannot be applied for the time series models. The result is surprising and lead to the fact that only conditional mean model should be used for time series modelling and comparing with NAR models.

Table 32. Result of Engle's ARCH test for ARMA(5,8) -model residuals

Engle's ARCH Test for Heteroscedasticity					
H_0: Residuals exhibits no ARCH effects					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Signifigance Level
FALSE	0.245	5.4465	9.4877	4	0.050

Even though GARCH -models can be implemented to ARMA model, because residuals don't have autoregressive conditional heteroscedastic effects, it is interesting to see how the residuals are distributed. To do this, Jarque-Bera test for normality is used. The null hypothesis that residuals are normally distributed is rejected with test statistic of 76.0570 and p-value of <0.001 with significance level of 0.050. The critical value of the test is 5.7022. The result of the test is presented in table 33 below.

Table 33. Result of Jarque-Bera test for ARMA(5,8) -model residuals

Jarque-Bera Test for Normality				
H_0: Residuals are normally distributed				
Null Rejected	P-Value	Test Statistic	Critical Value	Signifigance Level
TRUE	<0.001	76.0570	5.7022	0.050

5.2.2 Result of NAR architecture estimation

From the Levenberg-Marquardt performance matrix in appendix 2 it can be seen that the best NAR test performance with LM training algorithm is achieved with delay parameter value of 8 when the average MSE of NAR with LM training algorithm for the in-the-sample test sample is 3.3003. From the performance matrix in appendix 3 it is seen that best test performance with SCG training algorithm is also achieved with delay parameter value of 8 when average MSE of the in-the-sample test sample is 3.2070. For BR training algorithm the best test performance is achieved when the delay parameter value is set to 3 according to the performance matrix in appendix 4, when the average MSE of NAR for the in-the-sample test sample is 3.1189. The complete performance results for delay -parameter estimation of LM, SCG and BR training algorithms with defined delays is presented in the appendices 2, 3 and 4, respectively. Figure 27 presents the different values of test performance for NAR with different training algorithms on defined delays measured with

MSE of the in-the-sample test sample. Value of hidden layer neurons is set to 2 as mentioned previously in the chapter of regarding model building and forecasting procedure.

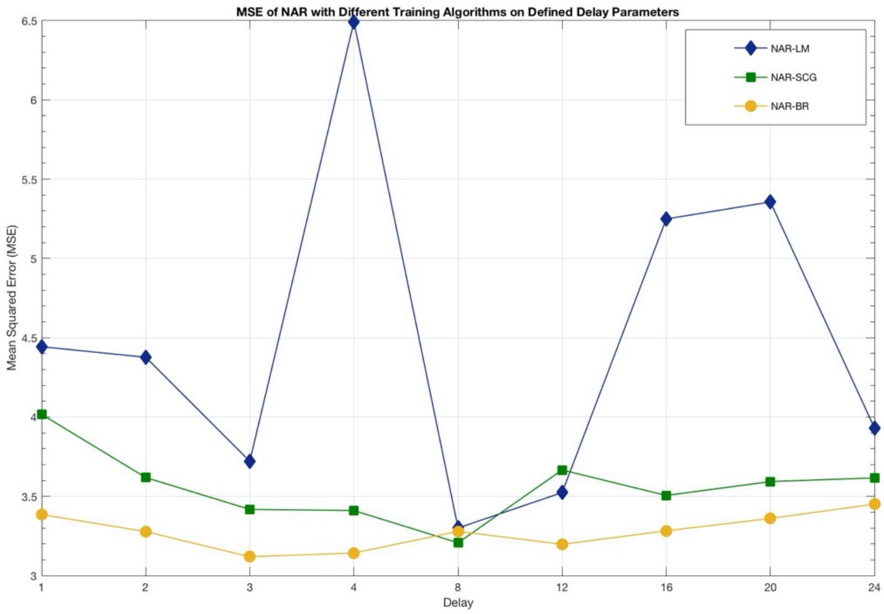


Figure 27. MSE of defined delay parameters for NARs; FIN_growth_rate

As the result of performance evaluation, the right architecture for NAR with LM training algorithm is decided to be with 2 hidden layer neurons and with 8 delays because this architecture produces best performance for the network with lowest test sample MSE. The architecture of open loop NAR with LM training algorithm and closed loop NAR with LM training algorithm with these parameters are presented in appendix 5.

The result of performance evaluation for SCG training algorithm indicates that the best NAR model performance with SCG training algorithm also is achieved when the delay parameter is set to 8 with 2 hidden layer neurons. This is decided to be the NAR architecture for SCG training algorithm since it produces the lowest test sample MSE. The architecture of open loop NAR with SCG training algorithm and closed loop NAR with SCG training algorithm with these parameters are presented in appendix 5.

For BR training algorithm, the best NAR performance is achieved with delay parameter value of 3 with 2 hidden layer neurons, which produces the lowest test sample MSE. For modelling and forecasting this is decided to be the NAR architecture with BR training algorithm. The architecture of open loop NAR with BR training algorithm and closed loop NAR with BR training algorithm with these parameters are presented in appendix 5.

5.2.3 Results of model performance and forecasting

Table 34 shows the results of forecasting performance and ranking of the forecasting models for quarterly growth rate of real GDP in Finland. It is noteworthy to mention, that GARCH -models couldn't be used to produce forecast and model time series since the result of Engle's ARCH test after fitting the conditional mean model was that the residuals exhibit no autoregressive conditional heteroscedastic effects. Therefore, result for hybrid ARMA-GARCH -models are not existing. The results shows that the most accurate forecast is produced by NAR-SCG among the models when loss function MAE is used for evaluating the forecasts. MAE for this model is 0.5506. The second-best performing model is ARMA(5,8) with MAE of 0.5507. The difference between performance of the models is minimal. Third and fourth best performing models are NAR-BR and NAR-SCG with MAEs of 0.5548 and 0.7045, respectively. The results show that NAR-LM and ARMA(8,5) models perform almost equally well. The worst performing model is NAR-LM, and the other models outperform it.

When viewing the results according to MSE, ARMA(5,8) -model outperform all of the NAR models with MSE value of 0,4298. The second-best performing model is NAR-BR with MSE of 0.4493. NAR-SCG and NAR-LM performs the worst with MSE values of 0.4610 and 0.6843, respectively

Table 34. Model performance ranking for FIN_growth_rate

Model	MAE	Ranking	Model	MSE	Ranking
NAR-SCG	0.5506	1	ARMA(5,8)	0.4298	1
ARMA(5,8)	0.5507	2	NAR-BR	0.4493	2
NAR-BR	0.5548	3	NAR-SCG	0.4610	3
NAR-LM	0.7045	4	NAR-LM	0.6843	4
ARMA(5,8)-GARCH(1,1) normal	N/A	N/A	ARMA(5,8)-GARCH(1,1) normal	N/A	N/A
ARMA(5,8)-GARCH(1,1) Student-t	N/A	N/A	ARMA(5,8)-GARCH(1,1) Student-t	N/A	N/A
ARMA(5,8)-EGARCH(1,1) normal	N/A	N/A	ARMA(5,8)-EGARCH(1,1) normal	N/A	N/A
ARMA(5,8)-EGARCH(1,1) Student-t	N/A	N/A	ARMA(5,8)-EGARCH(1,1) Student-t	N/A	N/A
ARMA(5,8)-GJR-GARCH(1,1) normal	N/A	N/A	ARMA(5,8)-GJR-GARCH(1,1) normal	N/A	N/A
ARMA(5,8)-GJR-GARCH(1,1) Student-t	N/A	N/A	ARMA(5,8)-GJR-GARCH(1,1) Student-t	N/A	N/A

5.3 Results for Quarterly Growth Rate of real GDP, Iceland

5.3.1 Results of conditional mean and variance model estimation

The best conditional mean model for modelling quarterly growth rate of real GDP in Iceland seem to be ARMA(7,7) with AIC of 945.956. The complete constructed matrix of the values of information criteria AIC for any combination of order 1 to 9 of AR(p) and MA(q) models can be seen in appendix 1. To evaluate the performance of the model, it is fitted to the in-the-sample data. To check how the model performs graphically ACF and PACF of the model is evaluated in figure 28. It can be seen that the residuals of the model might exhibit some serial autocorrelation.

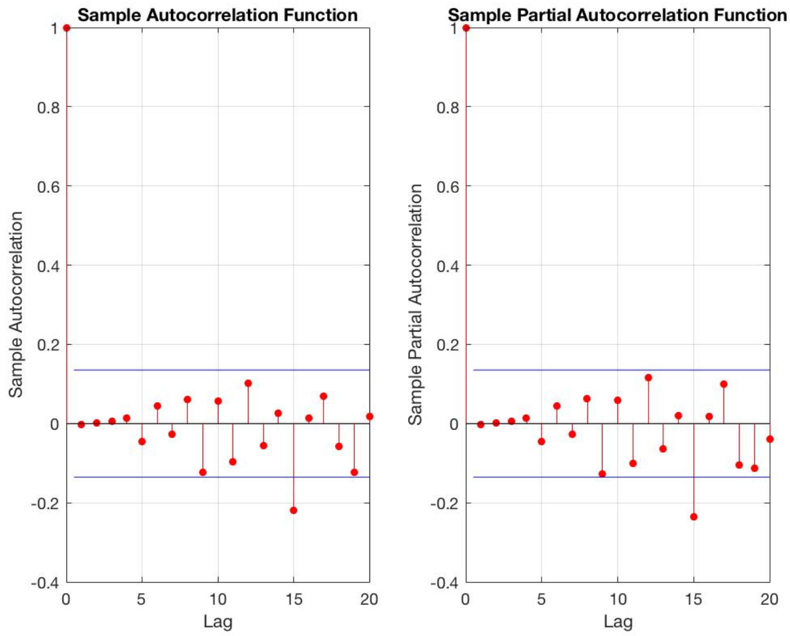


Figure 28. ACF and PACF -plots of fitted ARMA(7,7) -model for ISL_growth_rate

To further investigate this Ljung-Box test for autocorrelation is used. The result of the Ljung-Box test is presented in table 35 below. The null hypothesis of Ljung-Box test that residuals exhibit no serial autocorrelation fails to be rejected with test statistic of 28.7808 and p-value of 0.092 with significance level of 0.050. The critical value of the test is 31.4140. The result indicates that the residuals of the fitted model are not autocorrelated even though the result comes to close rejection of the null hypothesis.

Table 35. Result of Ljung-Box test for ARMA(7,7) -model residuals

Ljung-Box Test for Autocorrelation				
H_0: Residuals exhibit no serial autocorrelation				
Null Rejected	P-Value	Test Statistic	Critical Value	Signifigance Level
FALSE	0.092	28.7808	31.4140	0.050

To detected whether the residuals exhibit autoregressive conditional heteroscedastic effects, Engle's ARCH test is used. The null hypothesis of the test is that residuals exhibit no ARCH effects. The number of lags used in the test is 4 because of quarterly data. The null hypothesis is rejected with test statistic of 14.0006 and p-value of 0.007 with significance level of 0.050. The critical value of the test is 9.4877. The result of the test is presented in table 36 below. The test result indicates that there is autoregressive conditional heteroscedasticity among the residuals and that it would be useful to include GARCH model in time series modelling and forecasting.

Table 36. Result of Engle's ARCH test for ARMA(7,7) -model residuals

Engle's ARCH Test for Heteroscedasticity					
H_0: Residuals exhibits no ARCH effects					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Signifigance Level
TRUE	0.007	14.0006	9.4877	4	0.050

Jarque-Bera test for normality is used to test if the residuals of the model are normally distributed. The null hypothesis that residuals are normally distributed is rejected with test statistic of 81.8234 and p-value of <0.001 with significance level of 0.050. The critical value of the test is 5.7022. The test result indicates that it would be beneficial to also use alternatively t-distribution in conditional volatility modelling with GARCH -models. The result of the test is presented in Table 37 below.

Table 37. Result of Jarque-Bera test for ARMA(7,7) -model residuals

Jarque-Bera Test for Normality				
H_0: Residuals are normally distributed				
Null Rejected	P-Value	Test Statistic	Critical Value	Signifigance Level
TRUE	<0.001	81.8234	5.7022	0.050

5.3.2 Result of NAR architecture estimation

From the Levenberg-Marquardt performance matrix in appendix 2 it can be seen that the best NAR test performance with LM training algorithm is achieved with delay parameter value of 3 when the average MSE of NAR with LM training algorithm for the in-the-sample test sample is 14.4949. From the performance matrix in appendix 3 it is seen that best test performance with SCG training algorithm is also achieved with delay parameter value of 3 when average MSE of the in-the-sample test sample is 14.8640. For BR training algorithm the best test performance is achieved when the delay parameter value is set to 4 according to the performance matrix in appendix 4, when the average MSE of NAR for the in-the-sample test sample is 13.1522. The complete performance results for delay -parameter estimation of LM, SCG and BR training algorithms with defined delays is presented in the appendices 2, 3 and 4, respectively. Figure 29 presents the different values of test performance for NAR with different training algorithms on defined delays measured with MSE of the in-the-sample test sample. Value of hidden layer neurons is set to 2 as mentioned previously in the chapter of regarding model building and forecasting procedure.

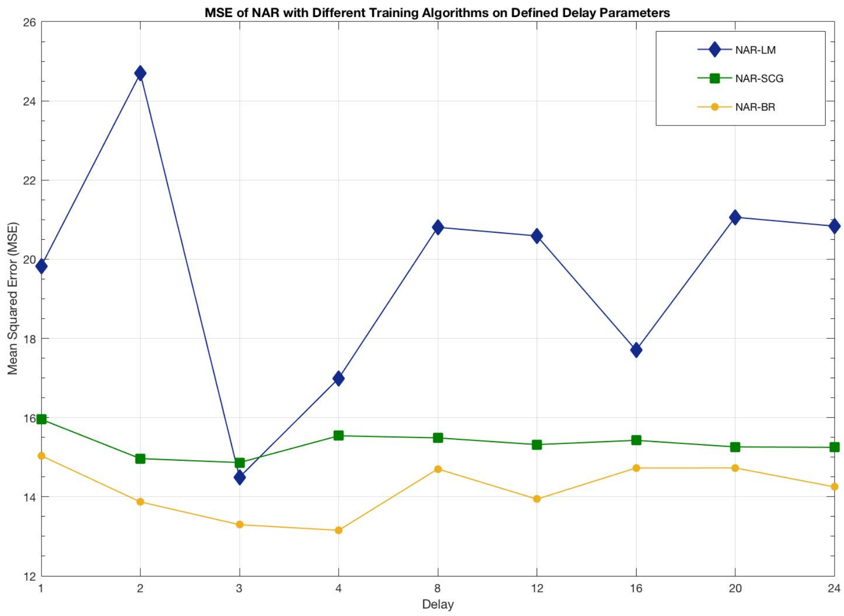


Figure 29. MSE of defined delay parameters for NARs, ISL_growth_rate

As the result of performance evaluation, the right architecture for NAR with LM training algorithm is decided to be with 2 hidden layer neurons and with 3 delay because this architecture produces best performance for the network with lowest test sample MSE. The architecture of open loop NAR with LM training algorithm and closed loop NAR with LM training algorithm with these parameters are presented in appendix 6.

The result of performance evaluation for SCG training algorithm indicates that the best NAR model performance with SCG training algorithm also is achieved when the delay parameter is set to 3 with 2 hidden layer neurons. This is decided to be the NAR architecture for SCG training algorithm since it produces the lowest test sample MSE. The architecture of open loop NAR with SCG training algorithm and closed loop NAR with SCG training algorithm with these parameters are presented in appendix 6.

For BR training algorithm, the best NAR performance is achieved with delay parameter value of 4 with 2 hidden layer neurons, which produces the lowest test sample MSE. For modelling and forecasting this is decided to be the NAR architecture with BR training algorithm. The architecture of open loop NAR with BR training algorithm and closed loop NAR with BR training algorithm with these parameters are presented in appendix 6.

5.3.3 Results of model performance and forecasting

The results of forecasting performance and ranking of the forecasting models for quarterly growth rate of real GDP in Iceland are presented in table 38. Ranking of the models seem to be same for loss functions MAE and MSE. The best performing model is NAR-LM with MAE of 2.5684 and MSE of 9.7806. The second-best performing model is NAR-BR with MAE of 2.5846 and MSE of 9.8216. NAR-SCG became third in ranking with MAE of 2.5945 and MSE of 9.8663. All NAR models outperform ARMA-GARCH models. Among ARMA-GARCH -models the best performing model was ARMA(7,7)-EGARCH(1,1) with Normal distribution. Loss function values for this model were MAE of 3.2994 and MSE of 10.0872. The worst performing model was ARMA(7,7)-EGARCH(1,1) with Student’s t-distributions with MAE of 3.3758 and MSE of 20.7289.

Table 38. Model performance ranking for ISL_growth_rate

Model	MAE	Ranking	Model	MSE	Ranking
NAR-LM	2.5684	1	NAR-LM	9.7806	1
NAR-BR	2.5846	2	NAR-BR	9.8216	2
NAR-SCG	2.5945	3	NAR-SCG	9.8663	3
ARMA(7,7)-EGARCH(1,1) normal	3.2994	4	ARMA(7,7)-EGARCH(1,1) normal	9.9854	4
ARMA(7,7)-GARCH(1,1) normal	2,6525	5	ARMA(7,7)-GARCH(1,1) normal	10.0872	5
ARMA(7,7)-GJR-GARCH(1,1) normal	2.6602	6	ARMA(7,7)-GJR-GARCH(1,1) normal	10.1358	6
ARMA(7,7)-GJR-GARCH(1,1) Student-t	2,6602	7	ARMA(7,7)-GJR-GARCH(1,1) Student-t	17.3852	7
ARMA(7,7)-EGARCH(1,1) Student-t	3.2994	8	ARMA(7,7)-EGARCH(1,1) Student-t	18.9049	8
ARMA(7,7)-GARCH(1,1) Student-t	3.3758	9	ARMA(7,7)-GARCH(1,1) Student-t	20.7289	9

5.4 Results for Quarterly Growth Rate of real GDP, Norway

5.4.1 Results of conditional mean and variance model estimation

The best conditional mean model for modelling quarterly growth rate of real GDP in Norway is also ARMA(7,7) with AIC of 662.099. The complete constructed matrix of the values of information criteria AIC for any combination of order 1 to 9 of AR(p) and MA(q) models can be seen in appendix 1. To evaluate the performance of the model, it is fitted to the in-the-sample data. To check how the model performs graphically ACF and PACF of the model is evaluated in figure 30. ACF and PACF of the fitted model implicates no serial autocorrelation among the residuals.

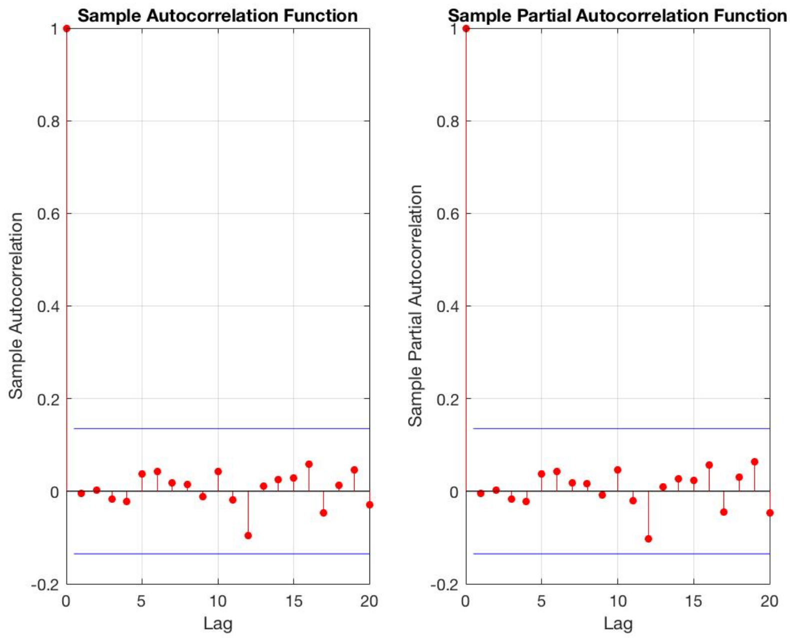


Figure 30. ACF and PACF -plots of fitted ARMA(7,7) -model for NOR_growth_rate

Ljung-Box test for autocorrelation is used to verify graphical evaluation. The result of the Ljung-Box test is presented in table 39 below. The null hypothesis of Ljung-Box test that residuals exhibit no serial autocorrelation fails to be rejected with test statistic of 6.1356 and p-value of 0.998 with significance level of 0.050. The critical value of the test is 31.4140. The result verifies that the residuals of the fitted model are not autocorrelated.

Table 39. Result of Ljung-Box test for ARMA(7,7) -model residuals

Ljung-Box Test for Autocorrelation				
H_0: Residuals exhibit no serial autocorrelation				
Null Rejected	P-Value	Test Statistic	Critical Value	Signifigance Level
FALSE	0.998	6.1356	31.4140	0.050

To detected whether the residuals exhibit autoregressive conditional heteroscedastic effects, Engle's ARCH test is used. The null hypothesis of the test is that residuals exhibit no ARCH effects. The number of lags used in the test is 4 because of quarterly data. The null hypothesis is rejected with test statistic of 16.9361 and p-value of 0.002 with significance level of 0.050. The critical value of the test is 9.4877. The result of the test is presented in table 40 below. The test result indicates that there is autoregressive conditional heteroscedasticity among the residuals and that GARCH -model could be used with ARMA -model in time series modelling and forecasting.

Table 40. Result of Engle's ARCH test for ARMA(7,7) -model residuals

Engle's ARCH Test for Heteroscedasticity					
H_0: Residuals exhibits no ARCH effects					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Signifigance Level
TRUE	0.002	16.9361	9.4877	4	0.050

Jarque-Bera test for normality is used to test if the residuals of the model are normally distributed. The null hypothesis that residuals are normally distributed is rejected with test statistic of 12.4951 and p-value of 0.009 with significance level of 0.050. The critical value of the test is 5.7022. The test result indicates that it would be beneficial to also use alternatively Student’s t-distribution in conditional volatility modelling with GARCH -models. The result of the test is presented in table 41below.

Table 41. Result of Jarque-Bera test for ARMA(7,7) -model residuals

Jarque-Bera Test for Normality				
H_0: Residuals are normally distributed				
Null Rejected	P-Value	Test Statistic	Critical Value	Signifigance Level
TRUE	0.009	12.4951	5.7022	0.050

5.4.2 Result of NAR architecture estimation

From the Levenberg-Marquardt performance matrix in appendix 2 it can be seen that the best NAR test performance with LM training algorithm is achieved with delay parameter value of 3 when the average MSE of NAR with LM training algorithm for the in-the-sample test sample is 1.4471. From the performance matrix in appendix 3 it is seen that best test performance with SCG training algorithm is achieved with delay parameter value of 1 when average MSE of the in-the-sample test sample is 1.5802. For BR training algorithm the best test performance is achieved when the delay parameter value is set to 24 according to the performance matrix in appendix 4, when the average MSE of NAR for the in-the-sample test sample is 1.6093. The complete performance results for delay -parameter estimation of LM, SCG and BR training algorithms with defined delays is presented in the appendices 2, 3 and 4, respectively. Figure 31 presents the different values of test performance for NAR with different training algorithms on defined delays measured with MSE of the in-the-sample test sample. Value of hidden layer neurons is set to 2 as mentioned previously in the chapter of regarding model building and forecasting procedure.

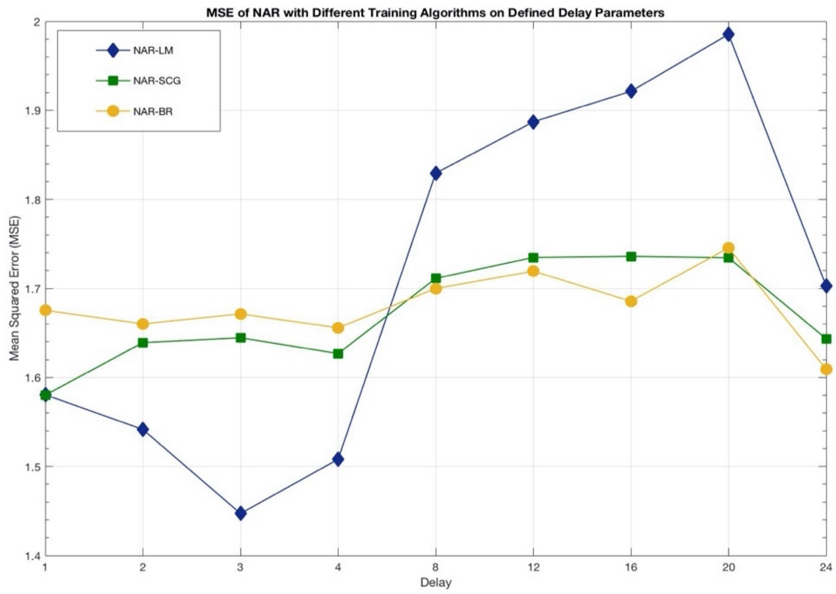


Figure 31. MSE of defined delay parameters for NARs, NOR_growth_rate

As the result of performance evaluation, the right architecture for NAR with LM training algorithm is decided to be with 2 hidden layer neurons and with 3 delays because this architecture produces best performance for the network with lowest test sample MSE. The architecture of open loop NAR with LM training algorithm and closed loop NAR with LM training algorithm with these parameters are presented in appendix 7.

The result of performance evaluation for SCG training algorithm indicates that the best NAR model performance with SCG training algorithm is achieved when the delay parameter is set to 1 with 2 hidden layer neurons. This is decided to be the NAR architecture for SCG training algorithm since it produces the lowest test sample MSE. The architecture of open loop NAR with SCG training algorithm and closed loop NAR with SCG training algorithm with these parameters are presented in appendix 7.

For BR training algorithm, the best NAR performance is achieved with delay parameter value of 24 with 2 hidden layer neurons, which produces the lowest test sample MSE. For modelling and forecasting this is decided to be the NAR architecture with BR training algorithm. The architecture of open loop NAR with BR training algorithm and closed loop NAR with BR training algorithm with these parameters are presented in appendix 7.

5.4.3 Results of model performance and forecasting

The results of forecasting performance and ranking of the forecasting models for quarterly growth rate of real GDP in Norway are presented in table 42. From the table it can be seen that the ranking is same for loss functions MAE and MSE. The best performing model is ARMA(7,7)-GARCH(1,1) with normal distribution, when the value of MAE is 0.6137 and MSE is 0.6089. The second-best performing model is ARMA(7,7)-GARCH(1,1) with Student's t-distribution when MAE is 0.6141 and MSE is 0.6096. All NAR models were outperformed by major of the ARMA-GARCH models. The difference in performance between best performing ARMA-GARCH -model and NAR -models is quite large. The worst performing model was ARMA(7,7)-EGARCH(1,1) with normal distribution with MAE of 0.8642 and MSE of 0.9695.

Table 42. Model performance ranking for NOR_growth_rate

Model	MAE	Ranking	Model	MSE	Ranking
ARMA(7,7)-GARCH(1,1) normal	0.6137	1	ARMA(7,7)-GARCH(1,1) normal	0.6089	1
ARMA(7,7)-GARCH(1,1) Student-t	0.6141	2	ARMA(7,7)-GARCH(1,1) Student-t	0.6096	2
ARMA(7,7)-GJR-GARCH(1,1) Student-t	0.6230	3	ARMA(7,7)-GJR-GARCH(1,1) Student-t	0.6139	3
ARMA(7,7)-GJR-GARCH(1,1) normal	0.6243	4	ARMA(7,7)-GJR-GARCH(1,1) normal	0.6264	4
ARMA(7,7)-EGARCH(1,1) Student-t	0.6339	5	ARMA(7,7)-EGARCH(1,1) Student-t	0.6275	5
NAR-LM	0.6858	6	NAR-LM	0.7632	6
NAR-BR	0.7018	7	NAR-BR	0.7744	7
NAR-SCG	0.7708	8	NAR-SCG	0.8880	8
ARMA(7,7)-EGARCH(1,1) normal	0.8642	9	ARMA(7,7)-EGARCH(1,1) normal	0.9695	9

5.5 Results for Quarterly Growth Rate of real GDP, Sweden

5.5.1 Results of conditional mean and variance model estimation

The best conditional mean model for modelling quarterly growth rate of real GDP in Sweden is ARMA(6,2) with AIC of 721.526. The complete constructed matrix of the values of information criteria AIC for any combination of 1 to 9 of AR(p) and MA(q) models can be seen in appendix 1. To evaluate the performance of the model, it is fitted to the in-the-sample data. To check how the model performs graphically ACF and PACF of the model is evaluated in figure 31. ACF and PACF of the fitted model shows that there might be some serial autocorrelation among the residuals of fitted ARMA(6,2) model.

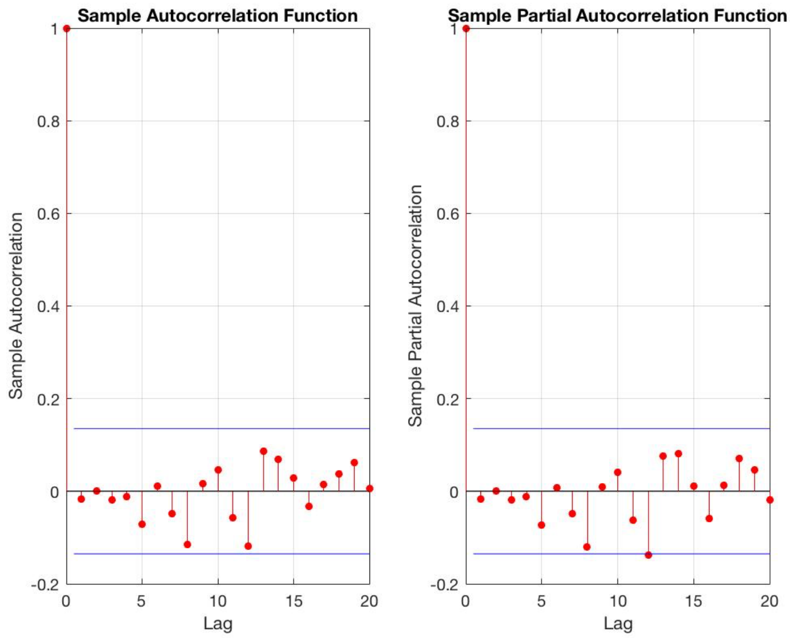


Figure 32. ACF and PACF -plots of fitted ARMA(6,2) -model for SWE_growth_rate

Ljung-Box test for autocorrelation is used to investigate further result of graphical evaluation. The result of the Ljung-Box test is presented in table 43 below. The null hypothesis of Ljung-Box test that residuals exhibit no serial autocorrelation fails to be rejected with test statistic of 14.1641 and p-value of 0.822 with significance level of 0.050. The critical value of the test is 31.4140. The result of the test is that the residuals of the fitted model are not autocorrelated.

Table 43. Result of Ljung-Box test for ARMA(6,2) -model residuals

Ljung-Box Test for Autocorrelation				
H_0: Residuals exhibit no serial autocorrelation				
Null Rejected	P-Value	Test Statistic	Critical Value	Signifigance Level
FALSE	0.822	14.1641	31.4140	0.050

To detected whether the residuals exhibit autoregressive conditional heteroscedastic effects, Engle's ARCH test is used. The null hypothesis of the test is that residuals exhibit no ARCH effects. The number of lags used in the test is 4 because of quarterly data. The null hypothesis is rejected with test statistic of 11.6952 and p-value of 0.020 with significance level of 0.050. The critical value of the test is 9.4877. The result of the test is presented in table 44 below. The test result indicates that there is autoregressive conditional heteroscedasticity among the residuals and that it is possible to use hybrid ARMA-GARCH -models in time series modelling and forecasting.

Table 44. Result of Engle's ARCH test for ARMA(6,2) -model residuals

Engle's ARCH Test for Heteroscedasticity					
H_0: Residuals exhibits no ARCH effects					
Null Rejected	P-Value	Test Statistic	Critical Value	Lags	Signifigance Level
TRUE	0.020	11.6952	9.4877	4	0.050

Jarque-Bera test for normality is used to test if the residuals of the model are normally distributed. The null hypothesis that residuals are normally distributed is rejected with test statistic of 64.2018 and p-value of <0.001 with significance level of 0.050. The critical value of the test is 5.7022. The test result indicates that it would be beneficial to also use alternatively t-distribution in conditional volatility modelling with GARCH -models. The result of the test is presented in table 45 below.

Table 45. Result of Jarque-Bera test for ARMA(6,2) -model residuals

Jarque-Bera Test for Normality				
H_0: Residuals are normally distributed				
Null Rejected	P-Value	Test Statistic	Critical Value	Signifigance Level
TRUE	<0.001	64.2018	5.7022	0.050

5.5.2 Result of NAR architecture estimation

From the Levenberg-Marquardt performance matrix in appendix 2 it can be seen that the best NAR test performance with LM training algorithm is achieved with delay parameter value of 2 when the average MSE of NAR with LM training algorithm for the in-the-sample test sample is 1.6693. From the performance matrix in appendix 3 it is seen that best test performance with SCG training algorithm is achieved also with delay parameter value of 2 when average MSE of the in-the-sample test sample is 1.4894. For BR training algorithm the best test performance is achieved yet again when the delay parameter value is set to 2 according to the performance matrix in appendix 4, when the average MSE of NAR for the in-the-sample test sample is 1.6215. The complete performance results for delay - parameter estimation of LM, SCG and BR training algorithms with defined delays is presented in the appendices 2, 3 and 4, respectively. Figure 33 presents the different values of test performance for NAR with different training algorithms on defined delays measured with MSE of the in-the-sample test sample. Value of hidden layer neurons is set to 2 as mentioned previously in the chapter of regarding model building and forecasting procedure.

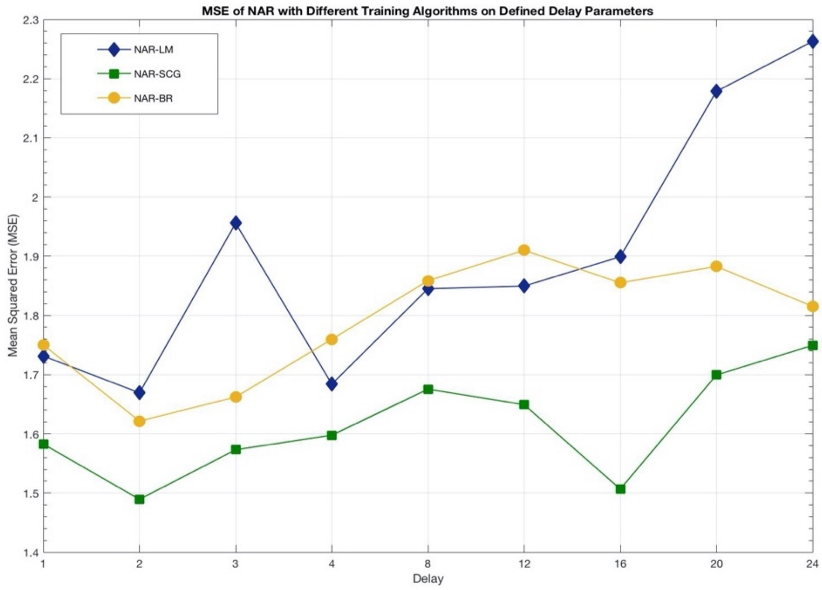


Figure 33. MSE of defined delay parameters for NARs, SWE_growth_rate

As the result of performance evaluation, the right architecture for NAR with LM training algorithm, SCG training algorithm and BR training algorithm is decided to be with 2 hidden layer neurons and with 2 delays because this architecture produces best performance for the network with lowest test sample MSE. The architecture of open loop and closed loop NAR with LM training algorithm, SCG training algorithm and BR training algorithm with these parameters are presented in appendix 8, respectively.

5.5.3 Results of model performance and forecasting

The results of forecasting performance and ranking of the forecasting models for quarterly growth rate of real GDP in Sweden are presented in table 46. The results implicates that the most accurate forecast is provided by ARMA(6,2,)-GJR-GARCH(1,1) with Student's t-distribution among the models when loss function MAE is used for evaluating the forecasts. MAE for this model is 0.4192. The second-best model is ARMA(6,2)-GARCH(1,1) with normal distribution when MAE is 0.4303. The difference between performance of the models notable. Third and fourth best performing models are ARMA(6,2)-GARCH(1,1) with Student's t-distribution and NAR-SCG with MAEs of 0.4412 and 0.4585, respectively. It is notable that the NAR models perform very differently among each other in forecasting this time series. The worst performing model is NAR-LM with MAE of 0.4608

When MSE is used as a loss function the results changes. ARMA(6,2)-GJR-GARCH(1,1) still remains as the best performing model with MSE of 0.2644. NAR-SCG becomes second-best performing model in this case with MSE of 0.2774. Again, the performance among NAR model varies a lot. The worst performing model according to MSE is ARMA(6,2,)-GARCH(1,1) with normal distribution. MSE of this model is 0.3082. The differences of MSEs between the models are not very large, except between the best performing model and rest of the models. The best performing ARMA(6,2)-GJR-GARCH(1,1) -model can be said to clearly outperform the other models

Table 46. Model performance ranking for SWE_growth_rate

Model	**MAE**	**Ranking**	**Model**	**MSE**	**Ranking**
ARMA(6,2)-GJR-GARCH(1,1) Student-t	0.4192	1	ARMA(6,2)-GJR-GARCH(1,1) Student-t	0.2644	1
ARMA(6,2)-EGARCH(1,1) normal	0.4303	2	NAR-SCG	0.2774	2
ARMA(6,2)-GARCH(1,1) Student-t	0.4412	3	ARMA(6,2)-GARCH(1,1) Student-t	0.2786	3
NAR-SCG	0.4585	4	ARMA(6,2)-EGARCH(1,1) Student-t	0.2807	4
ARMA(6,2)-GJR-GARCH(1,1) normal	0.4598	5	NAR-BR	0.2837	5
ARMA(6,2)-GARCH(1,1) normal	0.4605	6	ARMA(6,2)-GJR-GARCH normal	0.2857	6
NAR-BR	0.4608	7	ARMA(6,2)-EGARCH(1,1) normal	0.2911	7
ARMA(6,2)-EGARCH(1,1) Student-t	0.4633	8	NAR-LM	0.3021	8
NAR-LM	0.4685	9	ARMA(6,2)-GARCH(1,1) normal	0.3082	9

5.6 Overall ranking results of the models

Tables 47 and 48 presents the final rankings of the different types of models according to loss functions MAE and MSE. The final ranking is based on how the model has positioned on time series specific rankings so that the final ranking corresponds to how the model has been performed when all the time series specific ranking positions are considered and put in order. Points are given according to the time series specific ranking from 1 to 9 and the final ranking is done on basis of these points so that the best performing model has least points and worst performing model has the most points. Finland is excluded from these rankings since the results couldn’t be produced for that time series using ARMA-GARCH -models.

Table 47. Final ranking of the models according to MAE

Final rankings based on loss function MAE									
	NAR-LM	NAR-SCG	NAR-BR	ARMA-GARCH normal	ARMA-GARCH Student-t	ARMA-EGARCH normal	ARMA-EGARCH Student-t	ARMA-GJR-GARCH normal	ARMA-GJR-GARCH Student-t
Denmark	1	4	3	7	8	2	9	6	5
Finland	-	-	-	-	-	-	-	-	-
Iceland	1	3	2	5	9	4	8	6	7
Norway	6	8	7	1	2	9	5	4	3
Sweden	9	4	7	6	3	2	8	5	1
Points total	17	19	19	19	22	17	30	21	16
Overall ranking	**2**	**3**	**3**	**3**	**5**	**2**	**6**	**4**	**1**

Table 48. Final ranking of the models according to MSE

Final rankings based on loss function MSE									
	NAR-LM	NAR-SCG	NAR-BR	ARMA-GARCH normal	ARMA-GARCH Student-t	ARMA-EGARCH normal	ARMA-EGARCH Student-t	ARMA-GJR-GARCH normal	ARMA-GJR-GARCH Student-t
Denmark	2	3	4	8	7	1	9	6	5
Finland	-	-	-	-	-	-	-	-	-
Iceland	1	3	2	5	9	4	8	6	7
Norway	6	8	7	1	2	9	5	4	3
Sweden	8	2	5	9	3	7	4	6	1
Points total	17	16	18	23	21	21	26	22	16
Overall ranking	**2**	**1**	**3**	**6**	**4**	**4**	**7**	**5**	**1**

The best performing model overall according to MSE is ARMA-GJR-GARCH with Student’s t-distribution. Second best performing models are NAR-LM and ARMA-EGARCH with normal distribution. For third best performing model there are multiple ones performing equally well. These models are NAR-SCG, NAR-BR and ARMA-GARCH with normal distribution. For fourth and fifth becomes ARMA-GJR-GARCH with normal distribution and ARMA-GARCH with Student’s t-distribution. Worst performing model ARMA-EGARCH with Student’s t-distribution.

According to loss function MSE there are two equally well performing models. These models are NAR-SCG and ARMA-GJR-GARCH with Student’s t-distribution. Second-best performing model was NAR-LM and third became NAR-BR. ARMA-GARCH with Student’s t-distribution and ARMA-EGARCH with normal distribution were fourth best performing models. For fifth became ARMA-GJR-GARCH with normal distribution and for sixth ARMA-GARCH with normal distribution. The worst performing model was again ARMA-EGARCH with Student’s t-distribution

When examining the results as a whole it seems that NAR -models didn’t clearly outperform ARMA-GARCH models but did perform better that most of the ARMA-GARCH models.

6 CONCLUSIONS

6.1 Discussion

The aim of this thesis was to study differences in predicting capability of quarterly real growth rate of GDP in Nordic countries between of hybrid ARMA-GARCH -models and NAR -models with different training algorithms. Moreover, the focus was also on finding out are these models capable to predict GDP growth from time series data and how the performance of the models varies within each other according to predicting capabilities. With these objectives in mind, four research questions were formed to study in this thesis. The answers for these research questions are being discussed in this chapter.

Can ARMA-GARCH and NAR -models be used to predict quarterly growth rate of real GDP?

Hybrid ARMA-GARCH -models are result of combined conditional mean models and conditional volatility models. The advantage of these models is that they also consider nonlinear dependency of the data and are especially used to model such a time series data which have rather nonlinear characteristics than linear characteristics. Economic data, such as quarterly real growth rate of GDP is rarely linear in terms of variance, so therefore ARMA-GARCH models are useful when such an economic time series data is being modelled. However, ARMA-GARCH models do require a lot of observations in the data, since their modelling and predicting capabilities are based on previous values of the time series data. These models can also be utilized only when the data has autoregressive conditional heteroscedastic effects especially among the residuals of the conditional mean model. Therefore, there are limitations for the data being used and not all economic data has such a character which could be modelled by implementing hybrid ARMA-GARCH model to the time series. This was also found out in this study. The time series data of quarterly real growth rate of GDP in Finland had such a statistical feature (no autoregressive conditional heteroscedasticity among the residuals) that it couldn’t be modelled using hybrid ARMA-GARCH models. This leads to a conclusion that in some cases ARMA-GARCH models can be used to model and predict economic data such as quarterly real growth rate of GDP, but this is highly dependent on the time series data itself.

NAR -models doesn't assume linear dependency of the time series and are very flexible in terms of the characteristics of the time series data. These models can be utilized to model and predict noisy and volatile time series data and there are no underlying constraints for the data being used. However, familiarly with statistical requirements, if the data is not following certain distributions and has volatile characteristics, the results in modelling and predicting with NAR -models may be misleading and lead to misinterpretations. One advantage of NAR -models compared to ARMA-GARCH -models is the fact of no underlying constraints for the data, and therefore any kind of time series data can be modelled and predicted using NARs, but the result interpretation requires create judgement from the user whether the created and trained NAR -model is adequate enough to be used for modelling and predicting purposes.

Are NARs able to generate more accurate forecasts for predicting quarterly growth rate of real GDP in Nordic countries than hybrid ARMA-GARCH models?

NAR -models do not beat hybrid ARMA-GARCH models in predictive accuracy of quarterly real growth rate of GDP in Nordic countries, as the overall ranking showed. The overall best performing model was ARMA-GJR-GARCH with Student's t-distribution. The result is somewhat inconclusive, however, NAR -models seem to perform quite well among the models positioning to among better performing and more accurate end in the rankings beating most of the ARMA-GARCH models. As the result it can be said that even though NAR -models can't beat the best performing ARMA-GARCH -models, they do offer an equally well performing alternative to ARMA-GARCH -models in time series modelling and forecasting, especially when economic data, such as growth rate of GDP, is used. The results of this research somewhat are contradictory to those of Samir (2016) and Zukime (2004) but are not quite equally comparable, since these studies also utilized external regressor in ANN architecture.

The performance in predictive accuracy of NAR -models versus ARMA-GARCH -models is highly dependent on the time series data being modelled. Therefore, it can be said that for some time series NAR -models could be better option for time series modelling and predicting than ARMA-GARCH -models but again, this matter also requires great judgement from the user whether NAR -models should be implemented or not instead of ARMA-GARCH -models. In case of predicting economic growth for Denmark, NAR-SCG outperform the other models according to loss function MAE. NAR with BR training algorithm became third and NAR with SCG training algorithm became fourth among the models according to MAE. Best performing ARMA-GARCH -model according to MAE was ARMA-EGARCH with normal distribution. When investigating loss function MSE ARMA-EGARCH with normal distribution outperform NAR-LM -model, but al the NAR -models did, however, end up in the top four performing models according to MSE. For Finland the results were not interpretable since the ARMA-GARCH models could not be implemented for time series because of the characteristics of the data. However, NAR-SCG did outperform simple ARMA -model according to MAE, but not when MSE is used as loss function. Interestingly in case of Iceland, NARs were able to predict quarterly real growth of GDP more accurately that ARMA-GARCH -models and the result was same for both loss functions MAE and MSE. In case of Sweden, NAR models did perform worse than most of the implemented ARMA-GARCH -models in predicting economic growth based on quarterly data. However, when measured with loss function MSE NAR-SCG managed to become second-best performing model in prediction. The result of performance ranking implicates that the performance of NAR training algorithms varies depending on the used time series data.

Is there difference in predicting capability among ARMA-GARCH -models?

In overall MAE ranking ARMA-EGARCH with Student's t-distribution was performing the worst. In same overall ranking ARMA-GJR-GARCH with Student's t-distribution performed the best. There are clearly varying in predicting capabilities of ARMA-GARCH -models. The capability in predicting economic growth is highly dependent on the data, but still when investigating the overall ranking it can be clearly seen that positions in the ranking varies a lot among ARMA-GARCH -models. According to MAE, there seem to be high performing ARMA-GARCH models and models that do not perform so well.

In overall MSE ranking the result is not so clear. When investigating the overall performance of ARMA-GARCH -models according to MSE, most of the models seem to perform equally. In the basis of MSE performance it can be said that there is no difference in predicting capability among ARMA-GARCH -models.

Is there difference in predicting capability among NAR-models?

When considering overall model performance of NAR -models according to MAE there seem to difference in predicting capability between NAR-LM and the rest of the NAR -models. In overall ranking according to MAE NAR-LM became third, while the rest of the models became bot sixth. This led to a fact that there is no difference in predicting capability between NAR-SCG and NAR-BR, but there is difference between NAR-LM and the rest of the NAR-models, which NAR-LM outperformed.

When examining overall performance based on loss function MSE NAR-SCG performed the best among NAR -models. NAR-LM became second and NAR-BR third among NAR -models. However, since the positions of NAR -models in overall ranking were subsequent, it can't be said that there is difference in predicting capability among NAR -models according to loss function MSE

The objectives of this thesis were to compare predicting capabilities between ARMA-GARCH -models and NAR -models with different training algorithms. It can be concluded that there is some difference in predicting capabilities among these two types of model families, but the differences are highly dependent on the time series data being used. The results of this research are interesting and leads to a conclusion that both, ARMA-GARCH and NAR -type of models are useful in modelling and predicting economic time series but they should be used with caution and consideration need to be made about the used time series data so that the most accurate model is implemented in forecasting procedure. The models do produce usually equally good predictions for quarterly real growth rate of GDP in Nordic countries, so therefore there is no one model above all which would be advisable to use for all of the Nordic countries for prediction. There are few shortcomings in the analysis, for example using unique specifications for the different models in time series modelling and forecasting with ARMA-GARCH -models. However, the analysis of this thesis gives some insight into performance of NAR -models compared to ARMA-GARCH -models in modelling and forecasting univariate economic growth time series.

6.2 The limitations of the study

Limitations of this study is related to the time series data being used, which was quarterly real growth rate of GDP in five Nordic countries; Denmark, Finland, Iceland, Norway and Sweden. It is notable that the data had only 239 observations per time series and therefore too major conclusions of the model performance couldn't be made. This research succeeds giving insights into predicting economic growth with different time series modelling techniques but is not absolute guide for model selection. Model selection is always dependable of the data being used and therefore too broad conclusion of the model performance in general needs to be avoided. The other main limitation is related to the characteristics of economic data being used. Since the characteristics for one of the time series (Finland) did not fulfil the requirements of implementing ARMA-GARCH -models, the conclusions of their predictive capabilities for quarterly real growth rate of GDP in Finland could not be made leading to overall ranking made on basis of four Nordic countries only. Characteristics of the time series residual data for Finland did not have autoregressive conditional heteroscedastic effects, which lead to conclusion that GARCH -type of model cannot be implemented for such a time series that has no autoregressive conditional heteroscedastic effects. This might have not been the case, if the number of lags used for Engle's ARCH test had been increased, but since the lag parameter of the test is supposed to be set to 4 according to quarterly data, that would have led misleading test results. The different lag lengths for each of the time series may be problematic in comparing the models, but the choice of using specific lag lengths for each of the time series was made by the fact that every time series has unique characteristics, which cannot be captured by implementing one "universal" model for all of the time series and therefore each of the time series should be modelled and forecasted by best fitting ARMA-GARCH -model. This was also the case in defining the parameters of ANNs, where the delay parameter for each of the time series for different training algorithms was made by the decision criteria of smallest MSE for training set. Models of the different time series are therefore comparable only by their type, not their specifications and parameters. Limitation to the study was also using NAR -models which cannot be used with, and therefore, do not require use of external macroeconomic variables.

6.3 Suggestions for further research

For further research it would be advisable to use some other economic time series data than GDP growth, since the number of observations is limited. Further research could be done for example on investigating the predicting capabilities of NAR -model in unemployment rates or financial indexes. It would be also good to further investigate the predictive capabilities of NAR -models by introducing external factors for these models. The results might give more insight what economic factors are associated with economic growth and is it possible to predict economic growth with artificial neural networks based on external factors and how this improves the predictive accuracy.

REFERENCES

Abdullah, S. & Tayfur, A. 2004. An experimental study on forecasting using TES processes. Proceedings of the 2004 Winter Simulation Conference. editors: R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters

Babani, L., Jadhav, S. & Chaudhari, B. 2016. Scaled Conjugate Gradient Based Adaptive ANN Control for SVM-DTC Induction Motor Drive. *12th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI)*, pp.384- 395.

Baghirli, O. 2015. Comparison of Levenberg-Marquardt, Scaled Conjugate Gradient and Bayesian Regularization Backpropagation Algorithms for Multistep Ahead Wind Speed Forecasting Using Multilayer Perceptron Feedforward Neural Network. Uppsala University. Campus Gotland. Department of Earth Sciences. Dissertation. pp. 12-13

Batra, D. 2014. Comparison Between Levenberg-Marquardt And Scaled Conjugate Gradient Training Algorithms For Image Compression Using MLP. *International Journal of Image Processing*, vol. 8, no. 6.

Benmouiza, K. & Cheknane, A. 2013. Forecasting hourly global solar radiation using hybrid k-means and nonlinear autoregressive neural network models. *Energy Conversion and Management*, vol 75, pp. 561-569.

Benrhmach, G., Namir, A., Namir, K. & Bouyaghroumni, J. 2020. Nonlinear Autoregressive Neural Network and Extended Kalman Filters for Prediction of Financial Time Series. *Journal of Applied Mathematics,* vol. 20. [Online document]. [Accessed 28 April 2012]. Available at https://www.hindawi.com/journals/jam/2020/5057801/

Bollerslev, T. 1986. Generalised Autoregressive Conditionally Heteroscedasticity. *Journal of Econometrics*, vol. 31, pp. 307–327.

Brooks, C. 2008. Introductory Econometrics for Finance. Cambridge. Cambridge University Press.

Elsayir, H. 2018. An Econometric Time Series GDP Model Analysis: Statistical Evidences and Investigations. *Journal of Applied Mathematics and Physics*, vol. 6, no. 12, pp. 2635-2649.

Engle, R. 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of UK inflation. *Econometrica*, no. 50, pp. 987–1008.

Floros, C. 2005. Forcasting the UK Unemployment Rate: Model Comparisons. *International Journal of Applied Econometrics and Quantitative Studies*. no. 4, pp. 55-72.

Gavin, H. 2020. The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems. Duke University. Department of Civil and Environmental Engineering. [online document]. [Accessed 3 May 2021]. Available at http://people.duke.edu/~hpgavin/ce281/lm.pdf

Glosten, L., Jagannathan, R. & Runkle, D. 1993. On the Relation Between the Expected Value and the Volatility of the Nominal Excess Return on Stocks. *The Journal of Finance*, vol. 48, no. 5 pp. 1779–1801.

Hall, M., Muljawan, D., Moreena, S., & Moreena, M. 2008. Using The Artificial Neural Network (ANN) to Assess Bank Credit Risk: A Case Study of Indonesia. Loughborough: Loughborough University. Discussion paper series. WP2008-06. [online document]. [Accessed 27 April 2021]. Available at https://www.lboro.ac.uk/departments/sbe/RePEc/lbo/lbowps/CreditRisk-Using-ANN.pdf

Hennani, N., 2013. Application of ARMA-GARCH model and Support Vector Regression in Financial Time Series Forecasting. Master's Thesis. Universite Paris X Nanterre. UFR de sciences Economiques, de Gestion, Mathématique et d'Informatique.

Jahn, M. 2018. Artificial neural network regression models: Predicting GDP growth. HWWI Research papers 185. Hamburg Institute of International Economics.

Kamil, A. & Noor, A. 2006. Time series modeling of malaysian raw palm oil price: Autoregressive conditional heteroskedasticity (ARCH) model approach. *Discov. Math.* no. 28, pp. 19-32.

Kwiatkowski, D., Phillips, P., Schmidt, P. & Shin, Y. 1992. Testing the Null Hypothesis of Stationarity against the Alternative of a Unit Root. *Journal of Econometrics*, vol. 54, pp. 159-178.

Li, G. & Shi, J. 2012. Applications of Bayesian methods in wind energy conversion systems, *Renewable Energy*, vol. 43, pp. 1-8

Ljung, G. and Box, G. 1978. On a Measure of a Lack of Fit in Time Series Models. *Biometrika*. vol. 65, no. 2, pp. 297–303.

MathWorks. 2021a. adftest. [online document]. [Accessed 6 May 2021]. Available at https://se.mathworks.com/help/econ/adftest.html

MathWorks. 2021b. kpsstest. [online document]. [Accessed 6 May 2021]. Available at https://se.mathworks.com/help/econ/kpsstest.html

MathWorks. 2021c. vratiotest. [online document]. [Accessed 6 May 2021]. Available at https://se.mathworks.com/help/econ/vratiotest.html

Medeiros, M., Teräsvirta. T. & Rech, G. 2006. Building Neural Network Models for Time Series: A Statistical Approach. *Journal of Forecasting.* vol. 25, pp. 49-75.

Moller, M. 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, vol. 6, no. 4, pp. 525-533.

Nelson, D. B. 1991. Conditional Heteroskedasticity in Asset Returns: A New Approach. *Econometrica*, vol. 59, no. 2, pp. 347–370.

Nkwatoh, L. 2016. Can Cameroon become an emerging economy by the year 2035? Projections from univariate time series analysis. *Journal of Economics and International Finance*. vol. 8, pp. 155-167.

OECD. 2021. Quarterly National Accounts: Quarterly Growth Rates of real GDP, change over previous quarter. [online database]. [Accessed 13 January 2021]. Available at https://stats.oecd.org/index.aspx?queryid=60702

Pahlavni, M., & Roshan, R. 2015. The Comparison among ARIMA and hybrid ARIMA-GARCH Models in Forecasting the Exchange Rate of Iran. International Journal of Business and Development Studies, vol. 7, no. 1, pp. 31-50.

Samir, K. 2016. A Comparison of Artificial Neural Network and Time Series Models for Forecasting GDP in Palestine. *American Journal of Theoretical and Applied Statistics*, vol. 5, no. 2, pp.58-63.
Stephanie. 2016. What is the Jarque-Bera Test?. Statistics How To. [online document]. [Accessed 3 June 2021]. Available at https://www.statisticshowto.com/jarque-bera-test/

Sutelainen, E., 2019. A forecast comparison of volatility models: Evidence from Nordic equity markets. Master's Thesis. Lappeenranta University of Technology. School of Business and Management.

Taylor, S. J. 1986. Forecasting the Volatility of Currency Exchange Rates. *International Journal of Forecasting*, vol. 3, pp. 159–170.

Tealab, A. 2018. Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal,* vol. 3, Issue 2, pp. 334-340.
Terence, C., Mills, R. & Markellos, N. 2008. The Econometric Modelling of Financial Time Series. Cambridge University Press.

Tkacz, G. 2001. Neural network forecasting of Canadian GDP growth. *International Journal of Forecasting*, vol. 17, no. 1, pp. 57-69.

Wang, W., Van Gelder, P., Vrijling, J. & MA, J. 2005. Testing and modelling autoregressive conditional heteroskedasticity of streamflow processes. *Nonlinear Processes in Geophysics*, no. 12, pp. 55-56.

Whelan, K. 2016. 1. Introduction: Time Series and Macroeconomics. Lecture material. University College Dublin. School of Economics.

Würtz, D., Chalabi, Y. & Luksan, L. 2006. Parameter estimation of ARMA Models with GARCH/APARCH Errors An R and SPlus Software Implementation. *Journal of Statistical Software.* [online document]. [Accessed 1 June 2021]. Available at https://www.math.pku.edu.cn/teachers/heyb/TimeSeries/lectures/garch.pdf

Yue, Z., Songzheng, Z & Liu Tianshi, L. 2011, Bayesian regularization BP Neural Network model for predicting oil-gas drilling cost, *International conference on Business Management and Electronic Information (BMEI),* vol. 2, pp. 483,487

Zhang, G., Patuwo, B. & hu, M. 1999. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, vol. 14, pp. 35-62.

Zhang, X., Xue, T., &Stanley, H. 2018. Comparison of Econometric Models and Artificial Neural Networks Algorithms for the Prediction of Baltic Dry Index. *IEEE Access*, vol. 7, pp. 1647-1657

Zukime, M. 2004. Predicting GDP growth in Malaysia using knowledge-based economy indicators: A comparison between neural network and econometric approaches. *Sunway College Journal,* vol. 1, pp. 39-50

APPENDICES

Appendix 1 AIC matrices

AIC matrix, Denmark

Index of the AR(p) order

Index of the MA(q) order

	1	2	3	4	5	6	7	8	9
1	659.090	657.663	658.430	654.625	655.125	650.517	651.939	653.673	637.338
2	657.064	656.076	657.286	656.264	652.851	652.245	653.921	653.010	637.259
3	657.073	657.606	655.930	656.595	640.219	638.859	638.407	636.567	635.083
4	655.353	657.353	646.133	639.598	640.000	635.520	632.042	637.801	634.166
5	657,352	657.600	641.663	641.491	641.568	633.363	633.707	635.568	636.066
6	651.141	651.951	643.308	644.622	646.039	628.597	631.856	634.017	638.064
7	651.031	653.028	646.374	642.057	643.960	636.900	627.675	629.328	636.804
8	653.018	648.410	642.645	640.791	633.253	633.902	629.035	626.314	627.343
9	647.773	642.762	642.183	635.903	628.205	629.368	630.921	633.349	629.093

AIC matrix, Finland

Index of the AR(p) order

Index of the MA(q) order

	1	2	3	4	5	6	7	8	9
1	814.141	816.106	812.595	801.013	799.935	802.471	820.283	828.267	850.028
2	816.096	817.744	801.682	800.404	800.822	801.509	825.125	831.320	849.366
3	814.243	805.905	801.182	800.004	796.487	801.946	823.630	835.508	832.923
4	798.571	800.014	801.607	794.772	797.879	796.495	825.591	821.885	834.156
5	799.948	801.867	797.133	800.121	827.935	801.516	823.630	815.517	831.923
6	801.402	796.518	798.471	796.674	826.988	798.228	824.246	820.348	830.373
7	817.959	820.919	826.273	825.330	823.658	822.805	828.973	831.770	825.684
8	820.738	826.263	831.300	834.739	790.651	825.233	830.623	826.997	822.669
9	823.727	827.799	822.850	829.690	790.987	829.058	830.564	838.035	829.704

AIC matrix, Iceland

Index of the AR(p) order (columns); *Index of the MA(q) order* (rows)

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
1	990.580	984.285	969.768	968.740	962.825	964.626	962.983	964.308	984.752
2	979.860	968.640	963.349	965.259	962.266	957.933	959.209	961.112	978.333
3	970.191	964.983	965.257	954.035	954.603	956.567	958.280	961.121	980.241
4	969.377	966.753	955.634	954.508	956.506	957.027	946.593	948.054	970.618
5	959.810	957.987	956.708	956.467	957.239	958.891	949.781	951.644	971.692
6	960.885	955.835	957.816	956.453	958.397	946.590	947.447	953.417	972.800
7	962.782	957.834	957.295	958.085	960.063	946.148	945.956	951.914	972.278
8	962.300	962.599	958.254	954.947	954.345	951.597	954.576	954.466	973.237
9	973.254	970.630	959.511	958.385	960.383	960.904	950.470	951.931	974.495

AIC matrix, Norway

Index of the AR(p) order (columns); *Index of the MA(q) order* (rows)

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
1	683.707	680.993	680.669	682.603	683.645	681.925	683.213	693.851	693.527
2	680.119	681.423	677.448	679.427	679.290	679.897	681.897	694.281	690.306
3	680.896	676.177	676.713	674.335	667.720	674.689	676.538	689.035	689.571
4	682.257	678.176	667.311	670.383	669.685	676.686	678.536	691.033	680.168
5	683.995	680.511	671.071	671.200	663.391	665.390	666.595	693.369	683.928
6	680.481	679.858	678.790	677.518	665.387	667.908	671.590	692.716	691.648
7	682.215	681.810	680.790	674.533	666.588	665.244	662.099	694.667	693.648
8	683.983	683.578	682.559	676.301	668.356	667.013	663.868	696.436	695.416
9	685.752	685.347	684.327	678.070	670.125	668.781	665.636	698.205	697.185

AIC matrix, Sweden

Index of the AR(p) order

Index of the MA(q) order

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
1	735,069	734,768	734,683	735,881	728,161	730,071	732,939	735,807	738,675
2	734,297	734,572	732,604	733,892	730,074	721,526	724,394	727,262	730,130
3	734,486	725,361	725,631	736,673	732,052	726,789	729,657	732,525	735,393
4	736,140	726,493	727,611	728,390	734,047	724,907	727,775	730,643	733,511
5	731,525	733,315	732,707	733,213	732,993	734,945	737,813	740,681	743,549
6	733,316	727,216	732,821	729,080	728,664	725,027	727,895	730,763	733,631
7	740,017	730,370	731,488	732,267	737,924	728,784	731,652	734,520	737,388
8	735,402	737,192	736,583	737,090	736,870	738,822	741,690	744,558	747,426
9	737,193	731,093	736,698	732,957	732,541	728,904	731,772	734,640	737,507

Appendix 2 Levenberg-Marquardt performance matrices

Performance matrix, Levenberg-Marquardt algorithm, Denmark

Delay parameter

No. of training

	1	*2*	*3*	*4*	*8*	*12*	*16*	*20*	*24*
1	1.1260	1.5898	0.9308	0.9651	1.4626	0.7832	0.9776	1.2065	1.4081
2	1.1194	1.1448	0.9597	1.0817	1.0816	1.0072	1.0157	3.2721	1.7235
3	1.1849	1.1897	1.1759	0.9958	1.5282	1.0319	1.1326	1.1053	2.3318
4	1.0443	1.2201	1.3657	0.9473	1.5881	1.3855	1.1144	1.0470	1.1094
5	1.1301	1.2567	7.6237	1.0910	1.0441	1.1101	1.4129	1.2170	1.2521
6	1.1055	1.1210	1.0855	2.9230	1.1277	0.9184	1.1490	1.2759	1.3385
7	1.0869	1.1899	1.1968	1.2745	0.9960	1.5786	1.4925	1.3746	1.2858
8	1.1591	1.1593	1.1394	1.1226	1.1435	1.9294	1.2436	1.6323	1.1242
9	1.5222	1.1279	0.7582	1.1012	0.8928	1.3186	1.2036	0.7505	1.1938
10	1.0622	1.0607	1.0438	1.3648	1.3275	1.2386	1.7184	0.9896	1.1322
Avg.	**1.1541**	**1.2060**	**1.7279**	**1.2867**	**1.2192**	**1.2302**	**1.2460**	**1.3871**	**1.3899**

Performance matrix, Levenberg-Marquardt algorithm, Finland

Delay parameter

No. of training

	1	*2*	*3*	*4*	*8*	*12*	*16*	*20*	*24*
1	4.9771	3.3897	3.4541	3.2702	4.4138	9.8580	3.6838	4.4796	3.8755
2	6.7290	2.8658	3.0787	19.2997	3.5131	4.2231	6.0884	3.3443	3.6219
3	4.3199	5.2434	3.0968	6.7031	3.3269	3.5679	3.5128	10.5875	3.2126
4	3.4732	3.4973	2.9565	3.0207	3.2714	3.6493	6.0471	3.9508	3.9021
5	4.6498	3.4024	3.4144	12.3091	3.7689	2.9152	14.2021	3.0796	4.3741
6	3.3799	6.2913	3.0957	4.1457	2.9101	3.6647	3.1107	3.5276	4.3355
7	3.3272	4.9840	3.4837	3.0347	3.4633	2.9216	4.0969	4.1027	4.4992
8	5.1398	6.2965	8.0972	3.2219	3.1944	3.2938	3.2936	3.3145	4.1562
9	5.1150	3.3356	3.0268	3.5076	2.8973	4.1011	2.7299	4.4161	3.9865
10	3.3240	3.4760	3.2411	3.1888	3.3574	3.3758	4.1612	11.8814	3.2828
Avg.	**4.3842**	**4.3769**	**3.7212**	**6.4924**	**3.3003**	**3.5236**	**5.2492**	**5.3561**	**3.9301**

Performance matrix, Levenberg-Marquardt algorithm, Iceland

Delay parameter

No. of training	1	2	3	4	8	12	16	20	24
1	20.0223	15.5685	13.6750	15.2225	28.8842	15.0768	22.0477	13.3110	17.6923
2	16.6573	17.0360	14.2868	16.8631	18.3476	18.4862	17.7630	17.4310	17.3440
3	16.3609	34.0937	16.9456	17.3522	16.6848	16.0847	14.2463	15.9290	31.2079
4	16.5179	15.1308	15.4541	14.3904	15.7056	15.5171	38.7682	19.1121	21.7053
5	16.3807	13.7199	14.2485	18.4887	20.2870	15.3835	14.2964	24.5674	15.5576
6	12.1788	14.7751	17.4374	15.4655	20.5655	53.9922	12.4961	15.9052	14.8118
7	18.8842	22.0092	14.2430	15.2260	17.0233	17.6521	14.9111	22.2319	38.3188
8	15.4387	16.2490	12.2201	28.0826	16.2740	20.3685	12.6274	23.5989	14.1197
9	31.6781	82.2276	11.1470	13.6137	20.4342	17.4003	12.8450	40.1878	20.7215
10	34.1452	16.1286	15.2911	15.1885	33.8594	15.9275	17.0143	18.3394	16.8824
Avg.	**19.8264**	**24.6938**	**14.4949**	**16.9893**	**20.8066**	**20.5889**	**17.7016**	**21.0614**	**20.8361**

Performance matrix, Levenberg-Marquardt algorithm, Norway

Delay parameter

No. of training	1	2	3	4	8	12	16	20	24
1	1.6822	1.5812	1.2828	1.3882	2.0889	2.1955	1.6408	2.1763	1.5508
2	1.5827	1.5598	1.3921	1.4138	1.7513	2.0045	1.4558	2.3030	1.5583
3	1.4746	1.2677	1.1609	1.2156	1.9121	2.3509	2.4011	1.4655	1.4312
4	1.7536	1.2545	1.3735	1.4153	1.9846	1.5581	1.8559	1.4636	1.8202
5	1.4231	1.2526	1.7059	1.5508	1.5446	1.6409	2.1051	1.8651	1.7546
6	1.5700	1.8484	1.3041	1.4512	2.0359	2.8023	2.2736	2.1751	1.5072
7	1.7443	1.7164	1.4509	1.6795	1.9373	1.2673	2.1624	2.2693	1.4797
8	1.4011	1.6802	1.8091	1.5451	1.2667	1.4566	1.4209	1.9235	1.9569
9	1.5277	1.6834	1.3771	1.7412	1.9083	1.6890	1.9266	1.4858	2.0253
10	1.6458	1.5713	1.6153	1.6803	1.8646	1.9062	1.9715	2.7259	1.9420
Avg.	**1.5805**	**1.5416**	**1.4471**	**1.5081**	**1.8294**	**1.8871**	**1.9214**	**1.9853**	**1.7026**

Performance matrix, Levenberg-Marquardt algorithm, Sweden

Delay parameter

No. of training	*1*	*2*	*3*	*4*	*8*	*12*	*16*	*20*	*24*
1	1.9082	1.4264	1.7338	1.6913	1.6870	1.9102	1.6228	2.7643	1.9771
2	1.4780	1.7109	1.7284	1.8537	1.7318	1.8081	1.8191	3.0938	1.7929
3	2.4746	1.9129	1.9749	1.6248	2.3113	2.5202	1.6438	3.0744	3.5497
4	2.2321	1.5991	1.5425	1.6015	1.7415	1.8194	1.7481	1.9383	2.8300
5	1.6449	1.6021	2.1687	1.6706	1.6724	1.5667	1.9838	1.5670	1.7908
6	1.5086	1.6599	2.1495	1.9290	1.7461	1.8097	1.9757	2..1549	1.7759
7	1.4896	1.5635	2.6572	1.4936	1.4300	1.6694	2.5228	1.9187	3.0039
8	1.6492	1.5497	1.4876	1.7993	1.7272	1.7775	1.9021	1.6788	1.8525
9	1.4715	1.5974	2.0469	1.5432	2.9822	1.9507	1.5467	1.7648	1.7049
10	1.4575	2.0709	2.0691	1.6328	1.4225	1.6641	2.2268	1.8310	2.3511
Avg.	**1.7314**	**1.6693**	**1.9558**	**1.6840**	**1.8452**	**1.8496**	**1.8992**	**2.1786**	**2.2629**

Appendix 3 Scaled conjugate gradient performance matrices

Performance matrix, Scaled Conjugate Gradient algorithm, Denmark

Delay parameter

No. of training	1	2	3	4	8	12	16	20	24
1	1.1182	1.1410	1.0992	1.1719	0.8547	1.1655	1.1267	1.1675	1.1764
2	1.1556	1.7397	1.2067	1.0628	1.1068	1.2196	1.2108	1.3670	1.6112
3	1.1427	1.0753	1.2019	1.2468	1.3903	1.0217	1.4614	1.2037	1.2313
4	1.1396	1.1511	1.1358	1.1766	1.1891	1.4449	1.3012	1.1525	1.2317
5	1.1486	1.1931	1.2597	1.1742	1.2414	1.1783	1.1388	1.1556	1.2519
6	1.3544	1.1960	1.1759	1.1161	1.1328	1.1663	1.7260	0.9001	1.5311
7	1.1845	1.2264	1.0523	1.1069	1.5084	1.1852	1.0898	1.1707	1.3082
8	1.1526	1.1959	1.0525	1.1523	1.0124	1.1649	0.9791	1.1334	0.9943
9	1.1409	1.1949	1.1187	1.2771	1.1506	1.1082	1.1727	1.0765	1.4858
10	1.1501	1.1876	1.0366	1.0613	1.1595	1.1358	0.9855	1.1230	1.1121
Avg.	**1.1687**	**1.2301**	**1.1339**	**1.1546**	**1.1746**	**1.1790**	**1.2192**	**1.1450**	**1.2934**

Performance matrix, Scaled Conjugate Gradient algorithm, Finland

Delay parameter

No. of training	1	2	3	4	8	12	16	20	24
1	4.3675	3.3755	3.3451	3.3961	3.4791	5.1123	3.5926	3.5161	3.6198
2	3.5163	3.2397	3.0329	3.1512	3.2502	3.2586	4.0426	3.4345	3.9398
3	4.0550	3.9303	3.3365	3.6363	3.2273	3.5475	3.6042	3.5791	3.6111
4	3.3135	4.5260	3.3832	3.3015	3.0928	3.7253	3.5867	4.0523	3.6683
5	6.0395	4.0404	4.0779	3.8403	3.2130	3.4902	3.5307	3.3763	3.4018
6	3.3300	3.8683	3.3471	3.6276	3.2965	3.4410	3.4448	3.1852	3.4844
7	4.2059	3.2142	3.2180	3.3697	3.1877	3.3161	3.4664	4.3372	3.6331
8	3.3891	3.3013	3.0424	3.0885	3.2405	3.9696	3.2791	3.4699	3.6781
9	3.8799	3.3464	3.3649	3.5980	3.3542	3.1890	3.2639	3.4006	3.3126
10	4.0716	3.3453	4.0267	3.0924	2.7282	3.6137	3.2420	3.5801	3.8127
Avg.	**4.0168**	**3.6187**	**3.4175**	**3.4102**	**3.2070**	**3.6663**	**3.5053**	**3.5931**	**3.6162**

Performance matrix, Scaled Conjugate Gradient algorithm, Iceland

Delay parameter

No. of training	1	2	3	4	8	12	16	20	24
1	15.5939	16.5322	13.5177	12.0301	14.8650	12.2706	17.1264	14.3721	15.4203
2	13.7915	14.8000	13.5861	16.8809	13.5183	18.0994	15.3945	16.5270	14.1187
3	14.4722	11.8669	15.9733	14.0606	17.0065	13.2692	14.5719	16.4851	15.3724
4	19.0300	14.6277	15.5638	17.6976	16.4875	15.8375	15.0069	15.1658	13.8347
5	14.7907	14.2524	15.9014	15.4009	14.8216	15.3060	15.5812	12.7837	14.2627
6	15.0221	14.8896	14.6911	18.3353	18.3801	16.6048	15.6037	13.3118	14.8862
7	22.1879	15.5284	17.5059	15.4215	11.0522	12.4244	15.7660	14.5120	15.3694
8	15.4576	15.2433	14.4827	15.2431	16.8980	15.8740	15.0373	15.5769	19.3439
9	13.7116	17.2610	13.0380	15.2377	16.2740	17.6936	14.5119	17.6532	15.4295
10	15.4957	14.6382	14.3799	15.0989	15.5694	15.8030	15.6571	16.2066	14.4448
Avg.	**15.9553**	**14.9640**	**14.8640**	**15.5406**	**15.4873**	**15.3182**	**15.4257**	**15.2594**	**15.2483**

Performance matrix, Scaled Conjugate Gradient algorithm, Norway

Delay parameter

No. of training	1	2	3	4	8	12	16	20	24
1	1.5783	1.5039	1.4054	1.6846	1.7733	1.7352	1.6906	1.5997	1.8415
2	1.6041	1.6415	1.6026	1.5767	1.2254	1.5906	1.7029	1.9681	1.7146
3	1.6585	1.8183	1.7960	1.7342	1.6607	1.7183	1.7565	2.0737	1.3459
4	1.5717	1.5984	1.6634	1.7102	2.1098	1.7525	1.5780	1.6892	1.4352
5	1.4899	1.6896	1.6731	1.7615	1.8787	1.9203	2.0263	1.3333	2.1412
6	1.5576	1.6316	1.6916	1.6112	1.5631	1.5041	1.7703	1.7176	1.3758
7	1.5325	1.6532	1.6877	1.4791	1.8610	1.9744	1.7236	1.8176	1.7138
8	1.6230	1.5849	1.5888	1.4925	1.5822	1.6051	1.5839	1.7688	2.1108
9	1.6124	1.6570	1.5658	1.6359	1.4227	1.8040	1.8468	1.7661	1.4839
10	1.5738	1.6126	1.7710	1.5818	2.0360	1.7425	1.6803	1.6097	1.2715
Avg.	**1.5802**	**1.6391**	**1.6445**	**1.6268**	**1.7113**	**1.7347**	**1.7359**	**1.7344**	**1.6434**

Performance matrix, Scaled Conjugate Gradient algorithm, Sweden

Delay parameter

No. of training	*1*	*2*	*3*	*4*	*8*	*12*	*16*	*20*	*24*
1	1.5217	1.3459	1.7141	1.7960	1.5695	1.6794	1.4998	1.5513	2.3452
2	1.4863	1.4921	1.6240	1.8306	1.4761	1.5674	1.4795	1.6020	1.8183
3	1.4799	1.6523	1.6278	1.5474	1.4627	1.6773	1.6223	1.5209	1.6322
4	1.5472	1.5324	1.4288	1.4771	1.5469	1.6459	1.4386	1.8706	1.6603
5	1.7578	1.6065	1.6473	1.5943	1.5453	1.8500	1.4355	1.8873	1.6234
6	1.5698	1.3728	1.5833	1.7013	2.2274	1.5077	1.5060	1.5838	1.5343
7	1.4772	1.4201	1.6912	1.3053	1.8653	1.6249	1.4561	1.7882	1.8791
8	1.6832	1.4874	1.3741	1.4973	1.7245	1.9249	1.4144	1.8186	1.6858
9	1.7614	1.5994	1.4976	1.5487	1.6757	1.4543	1.6306	1.7587	1.6590
10	1.5424	1.3852	1.5433	1.6788	1.6618	1.5614	1.5840	1.6128	1.6562
Avg.	**1.5827**	**1.4894**	**1.5732**	**1.5977**	**1.6755**	**1.6493**	**1.5067**	**1.6994**	**1.7494**

Appendix 3 Bayesian regularization performance matrix

Performance matrix, Bayesian Regularization algorithm, Denmark

Delay parameter

No. of training	*1*	*2*	*3*	*4*	*8*	*12*	*16*	*20*	*24*
1	1.1731	1.1770	1.0874	1.0506	1.1678	1.0558	1.0222	1.1014	1.0954
2	1.0189	1.1601	1.1142	1.1093	1.1644	1.0837	1.0222	1.0867	1.1462
3	1.1700	1.0962	1.1119	1.1155	1.1483	1.0555	1.0222	1.1039	1.1523
4	1.1519	1.1620	1.0938	1.1444	1.1497	1.0556	1.0222	1.1038	1.1462
5	1.1871	1.1733	1.0940	1.0546	1.1470	1.0555	1.0222	1.1039	1.0811
6	1.1957	1.1595	1.0300	1.0617	1.1795	1.0556	1.0222	1.1039	1.1462
7	1.1837	1.1859	1.0853	1.0995	1.1496	1.0555	1.0222	1.1039	1.1419
8	1.2112	1.0291	1.0966	1.1641	1.1471	1.0554	1.0222	1.0554	1.1465
9	1.1693	1.1635	1.0139	1.0068	1.1470	1.0560	1.0222	1.1439	1.1462
10	1.0942	1.1601	1.1705	1.0492	1.1795	1.0556	1.0222	1.1014	1.1461
Avg.	**1.1555**	**1.1467**	**1.0898**	**1.0856**	**1.1580**	**1.0584**	**1.0222**	**1.1008**	**1.1348**

Performance matrix, Bayesian Regularization algorithm, Finland

Delay parameter

No. of training	*1*	*2*	*3*	*4*	*8*	*12*	*16*	*20*	*24*
1	3.4273	3.4080	3.1245	3.1424	3.2542	3.1930	3.2877	3.3563	3.4524
2	3.3107	3.2274	3.1242	3.1428	3.2542	3.2022	3.2819	3.3615	3.4523
3	3.2977	3.3103	3.1161	3.1424	3.2193	3.2015	3.2847	3.3609	3.4504
4	3.2152	3.2009	3.1244	3.1428	3.2542	3.1983	3.2756	3.3582	3.4513
5	3.2331	3.3103	3.1240	3.1428	3.2542	3.1937	3.2840	3.3617	3.4524
6	3.2925	3.1912	3.1242	3.1428	3.2542	3.1952	3.2801	3.3620	3.4523
7	3.4894	3.3103	3.1245	3.1423	3.2542	3.1920	3.2882	3.3619	3.4516
8	3.4273	3.2012	3.0778	3.1428	3.2542	3.2015	3.2818	3.3620	3.4524
9	3.6167	3.3092	3.1245	3.1428	3.2478	3.2023	3.2809	3.3591	3.4474
10	3.5426	3.3075	3.1245	3.1428	3.5433	3.1950	3.2795	3.3618	3.4476
Avg.	**3.3853**	**3.2776**	**3.1189**	**3.1426**	**3.2790**	**3.1975**	**3.2824**	**3.3605**	**3.4510**

Performance matrix, Bayesian Regularization algorithm, Iceland

Delay parameter

No. of training

	1	2	3	4	8	12	16	20	24
1	15.3512	13.6987	13.2950	12.9606	14.6800	13.1761	14.7048	14.9247	14.2491
2	15.4213	13.6981	13.2950	12.9580	14.9450	15.0005	14.7023	14.6776	14.2487
3	15.1005	13.6986	13.2951	12.9622	14.6391	13.1719	14.7023	14.6737	14.2540
4	15.4031	13.6983	13.2951	12.9580	14.6801	13.1744	14.7023	14.6732	14.2499
5	14.5412	13.6980	13.2951	14.9555	14.6801	15.0012	14.7040	14.6737	14.2535
6	14.7522	13.6981	13.2950	12.9580	14.6801	13.5778	14.9459	14.6736	14.2492
7	15.0995	13.6981	13.2950	12.9579	14.6801	15.0005	14.7023	14.6732	14.2487
8	15.3137	13.6981	13.2950	12.8907	14.6803	13.1718	14.7023	14.9589	14.2487
9	14.8896	15.4247	13.2951	12.9627	14.6803	13.1732	14.7032	14.6747	14.2539
10	14.5029	13.6981	13.2950	12.9581	14.6802	14.9522	14.7044	14.6773	14.2490
Avg.	**15.0375**	**13.8709**	**13.2951**	**13.1522**	**14.7025**	**13.9399**	**14.7274**	**14.7280**	**14.2505**

Performance matrix, Bayesian Regularization algorithm, Norway

Delay parameter

No. of training

	1	2	3	4	8	12	16	20	24
1	1.6751	1.6545	1.6756	1.7341	1.7405	1.7441	1.6270	1.6528	1.4998
2	1.6751	1.7071	1.5620	1.6583	1.5368	1.7723	1.6071	1.8134	1.5038
3	1.6750	1.6545	1.6718	1.5612	1.7406	1.7442	1.7383	1.8134	1.6401
4	1.6751	1.7540	1.6721	1.6583	1.5375	1.6177	1.7382	1.8134	1.6072
5	1.6830	1.6545	1.6721	1.6583	1.7405	1.7441	1.7388	1.6385	1.6420
6	1.6750	1.6545	1.8514	1.6583	1.7405	1.7441	1.7383	1.8134	1.6401
7	1.6751	1.6545	1.6721	1.6572	1.7406	1.7441	1.7386	1.6590	1.6401
8	1.6750	1.6545	1.5922	1.6583	1.7408	1.7442	1.7383	1.8134	1.6401
9	1.6751	1.5594	1.6721	1.6583	1.7408	1.7441	1.5873	1.6234	1.6395
10	1.6692	1.6545	1.6721	1.6540	1.7408	1.5940	1.6048	1.8134	1.6401
Avg.	**1.6753**	**1.6602**	**1.6713**	**1.6556**	**1.6999**	**1.7193**	**1.6857**	**1.7454**	**1.6093**

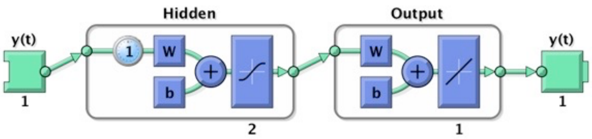
Performance matrix, Bayesian Regularization algorithm, Sweden

Delay parameter

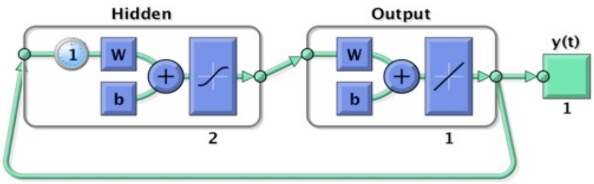
No. of training	*1*	*2*	*3*	*4*	*8*	*12*	*16*	*20*	*24*
1	1.7162	1.6367	1.6770	1.7597	1.5012	1.5645	1.8940	1.9469	1.9072
2	1.7258	1.6362	1.6770	1.7597	1.8975	1.5814	1.8939	1.9448	1.8561
3	1.7397	1.4855	1.6770	1.7597	1.8989	2.0307	1.8949	1.5831	1.9072
4	1.7152	1.6366	1.6770	1.7597	1.8989	2.0307	1.8940	1.9566	1.8084
5	1.7151	1.6367	1.6770	1.7597	1.8988	2.0307	1.8940	1.9302	1.8771
6	1.7152	1.6367	1.4964	1.7597	1.8986	2.0304	1.7632	1.9460	1.8754
7	1.7151	1.6367	1.6770	1.7597	1.8989	2.2399	1.8939	1.9473	1.9072
8	1.9478	1.6367	1.6770	1.7597	1.8959	1.5312	1.8939	1.9591	1.8040
9	1.7150	1.6366	1.6770	1.7597	1.8990	2.0307	1.7623	1.9467	1.6334
10	1.7972	1.6367	1.7091	1.7597	1.8988	2.0307	1.7664	1.6677	1.5774
Avg.	**1.7502**	**1.6215**	**1.6621**	**1.7597**	**1.8587**	**1.9101**	**1.8551**	**1.8828**	**1.8153**

Appendix 4 Architecture or open and closed loop NARs, Denmark

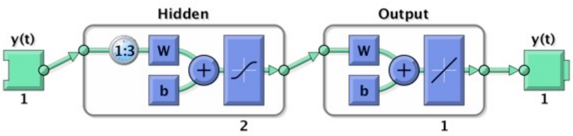
Architecture of open loop NAR-LM



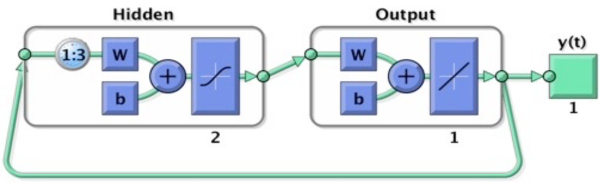
Architecture of closed loop NAR-LM



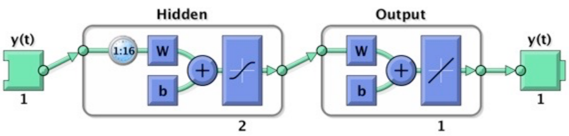
Architecture of open loop NAR-SCG



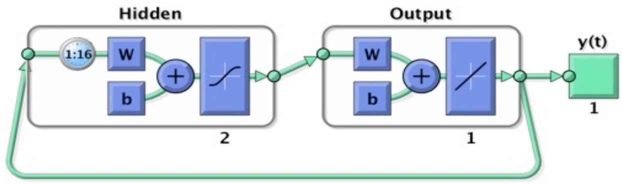
Architecture of closed loop NAR-SCG



Architecture of open loop NAR-BR

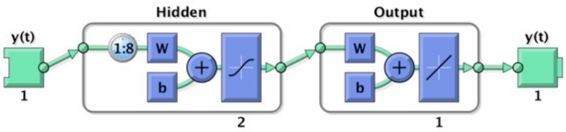


Architecture of closed loop NAR-BR

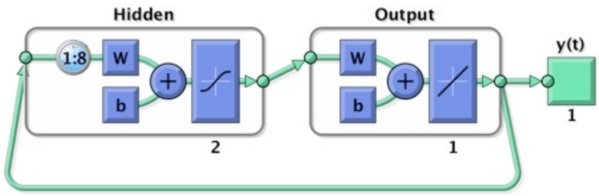


Appendix 5 Architecture of open and closed loop NARs, Finland

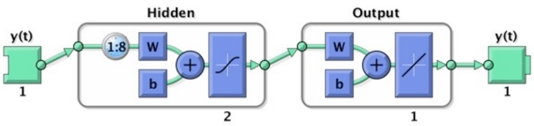
Architecture of open loop NAR-LM



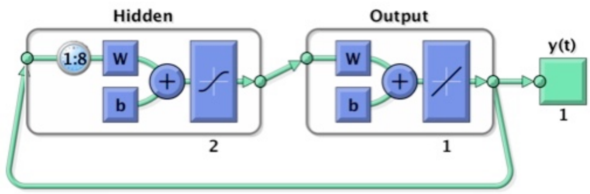
Architecture of closed loop NAR-LM



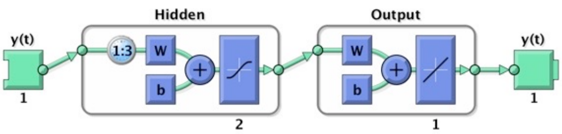
Architecture of open loop NAR-SCG



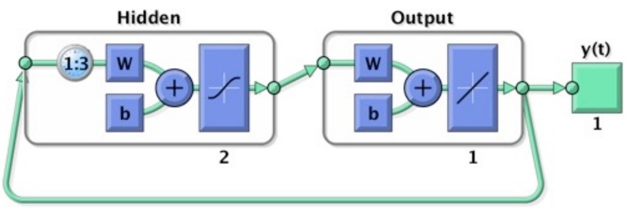
Architecture of closed loop NAR-SCG



Architecture of open loop NAR-BR

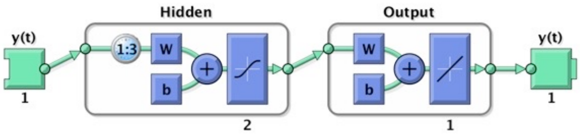


Architecture of closed loop NAR-BR

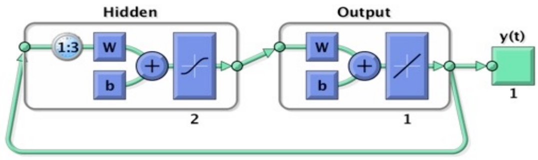


Appendix 6 Architecture of open and closed loop NARs, Iceland

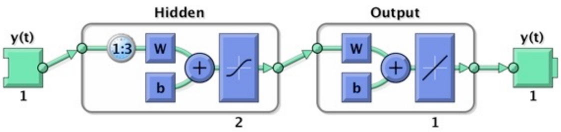
Architecture of open loop NAR-LM



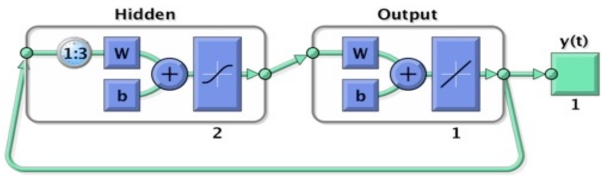
Architecture of closed loop NAR-LM



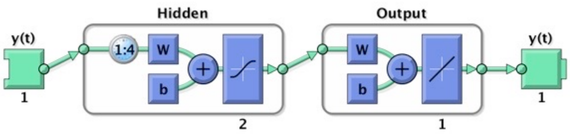
Architecture of open loop NAR-SCG



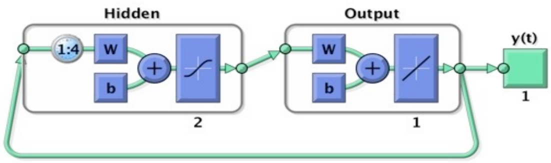
Architecture of closed loop NAR-SCG



Architecture of open loop NAR-BR

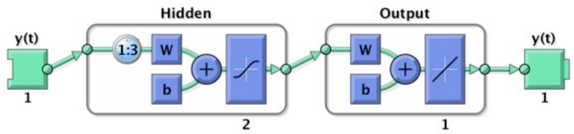


Architecture of closed loop NAR-BR

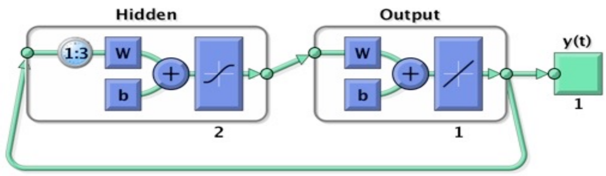


Appendix 7 Architecture of open and closed loop NARs, Norway

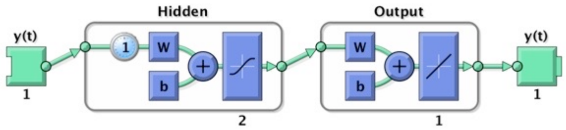
Architecture of open loop NAR-LM



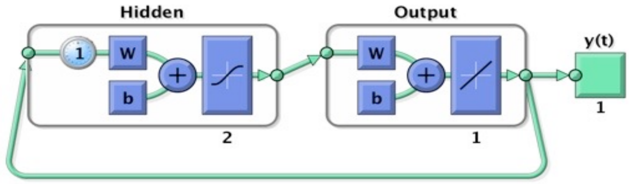
Architecture of closed loop NAR-LM



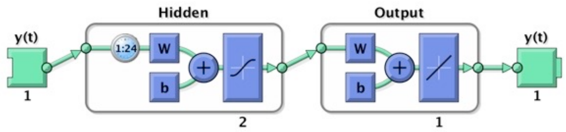
Architecture of open loop NAR-SCG



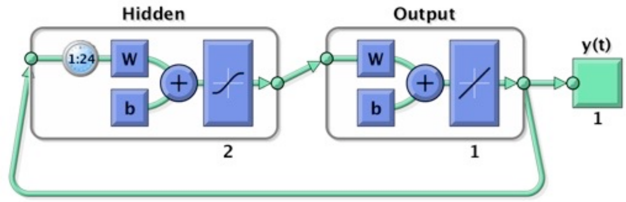
Architecture of closed loop NAR-SCG



Architecture of open loop NAR-BR

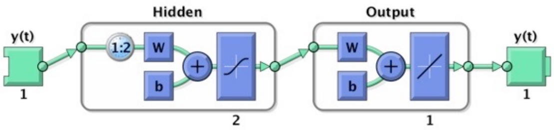


Architecture of closed loop NAR-BR

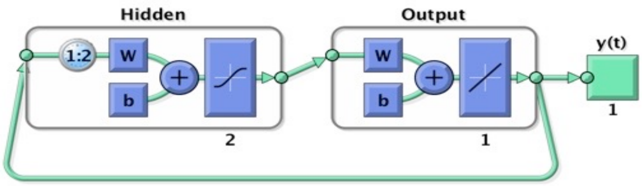


Appendix 8 Architecture of open and closed loop NARs, Sweden

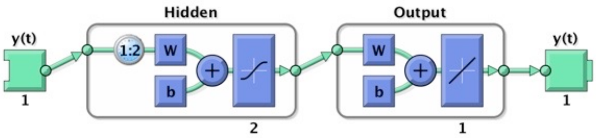
Architecture of open loop NAR-LM



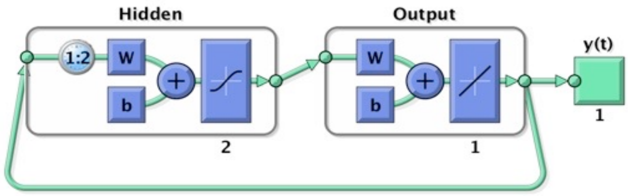
Architecture of closed loop NAR-LM



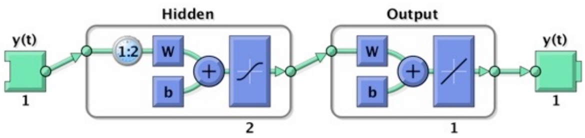
Architecture of open loop NAR-SCG



Architecture of closed loop NAR-SCG



Architecture of open loop NAR-BR



Architecture of closed loop NAR-BR