Forecasting NZ GDP with LSTM modelling

Author:

Scott Graham

Supervisor:

Dr Jaqueson K. Galimberti

A thesis submitted to

Auckland University of Technology

in partial fulfilment of the requirements for the degree

of Master of Business (MBus)

2023

School of Economics

Abstract

This thesis investigates traditional macroeconomic modelling technique's forecasting ability of GDP growth rates against a Long Short Term Memory Neural Networks forecasting ability of GDP growth rates for the New Zealand economy. The main forecast time frame is from March 2012 to December 2019 just before Covid-19. The models use an 8-step quarterly time horizon which is two years ahead. The models use pseudo-real time data to train the models in this thesis. I use a Root Mean Squared Error (RMSE) to measure the performance of the models and compare the forecasts with a Diebold-Mariano test (DM test). After the pre-Covid-19 period I use the models to forecast the Covid-19 period from March 2020 to September 2022. Finally, I compare the LSTM models to ANZ forecasts. The research in this thesis shows that the traditional model Vector Autoregression (VAR) has the best performance across all 8-time horizons and the LSTM out performances the VARMAX model for most of the 8-time horizons for the pre-Covid-19 period. The LSTM model performed best over the Covid-19 period but the RMSE results were so high that no meaningful insights could be used from the models. Finally, the LSTM outperforms ANZ forecasting over the Covid-19 period and slightly under performs before Covid-19. This thesis contributes to the research that is just beginning in this area of testing traditional models to newly developed models and helps Economists inspire more confidence in their forecasting abilities.

Acknowledgements

I would like to thank AUT University School of Economics for allowing me to complete my thesis on a topic I am deeply interested and passionate about. Economic research in this area is increasing and my hope is that we will see more work on Machine Learning come from the Economics department at AUT.

I am also very grateful to my supervisor, Dr. Jaqueson Kingeski Galimberti, for his guidance and continual feedback to improve my thesis. Without his expertise I would not have finished this research.

Last but not least, I would like to thank the online community that contribute to open source projects. This research project was largely code which was completed in Jupyter Notebook in Python and using a range of open-source libraries like Keras.

Thank you.

Scott Graham

Contents

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Chapter 1: Introduction

Forecasting GDP and other economic indicators is a fundamental task of any economics department in commercial banks and monetary government institutions. Economists are always looking for ways to improve forecasting accuracy of GDP so they can increase the confidence their stakeholders have in their forecasts. This increases business confidence which is a key driver for investment into an economy. Economic forecasts are very important to economic theory as well, especially with the development of expectations theory on inflation, if people expect inflation to go up, they will spend more now while their money is more valuable driving inflation up (Abaidoo, 2016). The more accurate economic forecasts are the more faith people will have in the forecasts which will impact their spending behaviour.

There are a wide range of approaches economists can take when forecasting GDP: at one end you have solely judgemental forecasts which are based on an expert person or persons stating what they think the future GDP figures are going to be and at the other end you have purely statistical models that forecast GDP. This thesis is based on statistical modelling approach. The poor forecasting performance of many macro models in the 1970s led to the application of vector autoregressions (VAR) models for forecasting macroeconomic indicators which are still largely used today (Stock, Watson, 2001). With the emergence of big data and Machine Learning (ML) it is time to see how the field of Machine Learning can contribute to the field of economics. The highest performing machine learning models for sequence data are the Long Short Term Memory (LSTM) models and attention models like Transformers. Attention models need a lot of data to converge to their optima which makes them very useful for Natural Language Processing (NLP) tasks but given economic timeseries is usually in quarters and data only goes so far back LSTM models are the most appropriate to apply to economic forecasting.

There are many different ways to apply ML models to economic forecasting of GDP. You could apply a univariate or multivariate model; you could predict one time step ahead or multiple timesteps ahead or you could split your data into training and test datasets to fit your model on the training data and test its predictive accuracy on the test data. Due to the success of ML models in other fields it is very natural for economists to try to apply them to economic problems and compare them against their traditional models to see if they can improve their forecasts. Reading through the literature on this topic I could not see that there has been an effective application of ML models to economic forecasting of real GDP that would be useful for institutions looking to improve their forecasts i.e., forecasting past the time period for which data is available.

To that end I have looked to focus on New Zealand's economy because it is the economy, I'm most familiar with and whose financial institutions I understand the best. Statistics New Zealand provides good reliable economic data that can be used for to support the modelling on real GDP. Because the data is available, I will build a multivariate model. ANZ bank, New Zealand's largest commercial bank, produces a quarterly economic outlook report which forecasts real GDP along with other macroeconomic indicators two years into the future. Traditionally when training a ML model, you split the data 80:20 into training and test data where you fit the model on the training data and test the model by using the test data's explanatory variables to enable to model to make predictions and then test them against the test data's response variable, in this case GDP growth rates. This approach is not suitable for forecasting economic data because the explanatory variables used for forecasting GDP are often not available until sometime after the time period has passed (it takes time to accurately collate the data). A univariate model is also not desirable for multistep forecasting into the future because the later predictions become more and more reliant on predicted values; a model that looks one time step backwards that is predicting the fourth quarter into the future is based on a predicted value that is based on the predicted value before which is based on the predicted value before it which is based on a real data value. The best approach for an LSTM model is to use a model that takes an input shape that predicts multiple timesteps in the future through the final output step.

Standard economic modelling software is limited to models that are inbuilt. As I am looking at applying a new type of model to an econometric problem, I had to use software outside the standard economic packages. I chose to use Python because of its Keras library which makes building LSTM Neural Networks intuitive. Because of this the data had to be imported into python and pre-processed to be in the right format to feed into the input layer of the LSTM. As well, functions had to be built to capture the data and format it correctly when it came out of the output layer of the LSTM model.

The objective of this thesis is to contribute to the field of macroeconomic forecasting by comparing an LSTM model to traditional statistical models for predicting GDP growth rates in New Zealand to see if the field of Machine Learning can be of benefit for economists. I have done this by building an LSTM model and trained it on data from Statistics New Zealand and forecasted GDP out eight quarters. I have done this from January 2013 to December 2019. I have then compared the results to traditional multivariate models Vector Autoregression (VAR) and Vector Autoregressive Moving Average with exogenous variables (VARMAX) models which were built on the same data. I have focused on forecasting GDP growth rates, although there are a range of economic indicators, I could predict, such as interest rates, exchanged rates, etc. However, GDP growth rates is generally considered the most important economic indicator when forecasting macroeconomic indicators. Most of the time was spent building the models rather than gathering data since the main objective was to test the LSTM models performance against the VAR and VARMAX models. The performance indicator Root Mean Squared Error (RMSE) was chosen to measure the model's performance against each other across the four quarters of the year.

Previewing the results, it shows that LSTM models, based on this data, cannot outperform traditional VAR models but economists should still be aware of these models to try improve their forecasting accuracy. The LSTM model was compared to a VARMAX and a VAR model and the results show that during an expansion period the LSTM model can outperform VARMAX models over certain forecast horizons but the VAR model had strong results over all time horizons. The VARMAX model had high variance with predictions often spiking forecasts in particular periods of the forecast window. The LSTM model's predictions had a good level of variance and the RMSE for all eight quarters were low. My hope is that research will start to accelerate machine learning modelling in this area and we'll see more applications of these models as economists incorporate them into their toolkit. It will be interesting to see how the LSTM models perform on more sophisticated datasets.

The rest of the thesis is organised as follows: the second chapter is a literary review of macroeconomic modelling that narrows down to forecasting in New Zealand. It then follows onto looking at LSTM modelling and Neural Networks application in forecasting and then further narrows down to ML economic forecasting done in New Zealand. In the third chapter I look at the methodology of each of the modelling approaches and lay out the general mathematical framework for each model. The fourth chapter details the data and insights into how the data was collected. I also discuss the models used and justify the hyperparameters chosen when building the models. In the fifth chapter I look into the forecasting results of the models pre covid 19. In the sixth chapter I use the same models and compare forecasts across and post covid 19. In chapter seven I compare

LSTM models to ANZ forecasts. In chapter eight I present the conclusions and how my research can be expanded upon.

Chapter 2: Literature Review

The first part of this literature review is a revision of the literature on multivariate forecasting of real GDP. The chapter then explores some of the data variables used in the models for forecasting. It also looks at how econometric modelling is applied within New Zealand's (NZ) main financial institutions. Following the econometric modelling I then review the evolution of NN to LSTM models and then looks at the application of NN on forecasting exercises mainly within the field of economic and financial modelling. Again, the final part of this chapter reviews research and applications of Machine Learning (ML) applied to economic modelling within NZ.

2.1 Macroeconomic modelling

Economists have a range of multivariate and univariate models they use for forecasting real GDP. Chauvet and Potter (CP 2013) review four multivariate models forecasting US GDP growth compared to the Blue-Chip indicators. The models reviewed are a structural DSGE model, reduced-form linear VAR, BVAR, and the non-linear Dynamic Factor Model with Regime Switching model. All these models are current options for economists who want to forecast economic output and their selection depends on a range of factors. All models when applied to forecasting GDP have the same goal, to forecast economic output with the greatest level of accuracy. Models can perform better in different stages of the business cycle so there is no silver bullet when forecasting GDP. All of the above models perform better during expansions than recessions. During expansions there is not much difference between the performance of each model. In-fact the benchmark univariate model performs best. During a recession however the multivariate models perform better than the univariate models which the best performing model is the Dynamic Factor Model with Markov Switching (AR-DFMS) (Chauvet & Potter, 2013). Economists need to evaluate several factors such as data availability and where in the business cycle they are when choosing which model they wish to employ to forecast GDP.

In the last 45 years there has been significant development in the macroeconomic models for forecasting the business cycle and although economists have a range of models they can use, Vector Autoregressions (VAR) have become the most widely adopted. Watson (Watson, 2000) states that VARs have become the workhorse models for answering questions about the business cycle variability. Both univariate autoregressions and VARs are now standard benchmarks used to evaluate economic forecasts. Although VARs are widely used there is still a lot of room for development. VARs often miss nonlinearities, conditional heteroskedasticity and drifts or breaks in parameters (Stock & Watson, 2001). VARs can perform well on forecasting some macroeconomic indicators and poorly on others even if they are over the same time period and within the same economy (Watson, 2000). Like most models VAR perform better during times of expansion and worse during recessions (Chauvet & Potter, 2013). These performance issues of VARs are general macroeconomic modelling issues so macroeconomists are always researching new ways to more accurately forecast economic output. This means the field is always developing.

Another multivariate macroeconomic technique is the Vector Autoregressive Moving Average (VARMA). VARMA models are closely related to VAR models, theoretically, we know that finite-order VAR models are a subset of finite-order VARMA models. VARMA models have shown mixed results for forecasting macroeconomic timeseries. Athanasopoulo and Vahid (2006) conclude in their paper, VARMA versus VAR for Macroeconomic Forecasting, that there is no reason for restricting your class of models to VAR given the advancements in VARMA models and they forecast macroeconomic variables with more accuracy than VAR. On the other hand, Kascha and Mertens (2009) have found few gains from using VARMA models to predict impulse responses of hours worked to a technology shock simulated data and compare true with estimated. Like any modelling method, results will vary depending on the data used and the modelling objective.

2.1.1 Variables used in macroeconomic modelling

When forecasting output with multivariate models economists have a wide range of variables to choose from. Although some research shows that adding variables in simple autoregressive time series models generally improves the performance of models in forecasting output growth (Chauvet & Potter, 2013) adding too many variables will no longer improve forecasting performance and leads to complications. In contrast to this however Stock and Watson (1996) found that small VARs of two or three variables are often unstable and therefore make poor predictors of the future states. State-of-the-art VAR forecasting systems use more than three variables but adding variables to the VAR creates complications, because the number of VAR parameters increases as the square of the number of variables: a nine-variable, four-lag VAR has 333 unknown coefficients. Time series models cannot provide estimates for all the coefficients unless further restrictions are applied (Stock & Watson, 2001). Other macroeconomic research has found that by using a large set of predictors instead of a moderate set does not improve forecasts (Carriero et al., 2019). Building forecasting models is often an art as much as a science and Economists need to balance their models correctly by not just selecting appropriate variables but getting the dataset the right size. This is often an iterative process that the researcher must conduct.

Financial variables can be added to multivariate models to help improve forecasting accuracy. Research has found that a few key financial variables have frequently provided useful additional information and sound predictions of economic activity. However, the predictive association between financial variables and real economic activity has been found to be unstable across developed economies and different time periods. The importance of the link between financial markets and economic growth is stronger in developed economies than for emerging markets. Reviewing the G-7 economies it has been found real financial predictors perform better than nominal predictors for forecasting GDP growth in G-7 countries during a financial crisis and the recovery from it (Kuosmanen & Vataja, 2017). Edirisuriya (2015) has found financial variables, such as the 90-day Treasury bill rate, 10-year Treasury bond rate, interest rate spread, Australian stock index data and housing prices play a key role in Australia's future economic activities. He goes on to conclude that the results reinforce previous research that macro-economic variables are a useful tool when forecasting Australia's economic activity (Edirisuriya, 2015). The Reserve Bank of New Zealand have a list of indicators they believe to be useful for forecasting GDP for the New Zealand economy; consumption, GDP, Business investment, residential investment, government expenditure, exports of goods and services, import goods and services, output gap and CPI inflation (RBNZ, n.d.). Because financial data can provide additional support to model performance when forecasting, it is important to benchmark multivariate models against other multivariate models rather than univariate ones when trying to find the best modelling approach to GDP forecasting.

2.1.2 Forecasting GDP within NZ

Forecasting GDP in the private sector is mainly done by the large banks (such as ANZ in their Economic outlook report) and in the public sector by The Treasury and the Reserve Bank of New Zealand (RBNZ). All institutions prefer to apply judgment-adjusted forecasts. ANZ apply judgmental overlays over model forecasts before publishing their forecasts in their Economic Outlook report (ANZ, 2020). The Treasury's Budget Economic and Fiscal Update 2021 also states on their website that the forecasts are based on assumptions and judgements from the time they're prepared. The Treasury does comparisons of their forecasts and has found that their GDP forecast performance was ranked at seventh place out of sixteen on an average relative rank basis. They found that their forecasts misjudged the impact of the Asian financial crisis, and droughts on economic activity in 1998 which resulted in large forecast errors. This had a material impact on Treasury's overall forecast performance (Treasury, 2021). The RBNZ also applies a judgement-adjusted forecast even though they have VAR models that outperform the RBNZ forecasts (Lees et al., 2011). Statistical models are overlooked by New Zealand's institutions when it comes to forecasting GDP which means that the new models available can add significant value.

2.2 Theoretical background on LSTM

Long Short-Term Memory (LSTM) model is a type of Neural Network (NN) that was developed as an extension of Recurrent Neural Networks (RNN). The first NN was developed at Cornell University by Frank Rosenblatt where he built the Perceptron (Lefkowitz, 2019). NNs were an area of research in both neuroscience and computer science in the 1960s and then their popularity faded. There was a resurgence in the 1980's when the back propagation algorithm was applied in the paper "Learning representations by back-propagating error" by Rumelhart, Hinton and Williams. The algorithm is used to train a NN through a chain rule. After the data moves forward through a network, backpropagation performs a retrospective pass back while adjusting the model's weights (Rumelhart et al., 1986). Research went quiet again by the 1990's as computers were not powerful enough to process the models. The field began to advance only in the 2000s with the advent of computers that were orders of magnitude more powerful and social media sites offered a tsunami of images, sounds, and other training data. The field started to officially pick up in 2009 when Hinton built NN models that could recognize speech better than any other known method (Waldrop, 2009). Since then, NN models have taken off and are used for a wide range of modelling applications such as speech recognition, pattern recognition, image classification, forecasting and many other applications. Many variants of the NN architecture have been developed to improve performance in particular areas of application, RNN were developed for handling sequence data.

RNN's provide an elegant way of dealing with sequential data that has correlations between data points that are close in the sequence. They were first proposed by Ruineihart, Hint, and Williams (1985) for learning internal representations and then they applied them to basic learning representation tasks with few errors. The key difference in an RNN to an NN is that an RNN updates its weights by measuring the error associated with a particular unit change, the units within the RNN layers have the capability to pass the error back to itself. After each iteration, as the error is being passed back through the network, the change in weight for that iteration is added to the weight changes specified by the preceding iterations and the sum stored. This process of passing the error through the network can continue for a number of iterations equal to the number of iterations through which the activation was originally passed. At this point, all of the weights are updated (Rumelhart et al., 1985).

The big break in forecasting with RNNs came when the architecture of the RNN was structured to process bidirectional inputs, that is past sequences and future sequences. This model is called a Bidirectional Recurrent Neural Network (BRNN). Future input information coming up later than the predicted period is usually also useful for prediction. This is done by splitting the state neurons of a regular RNN in a part that is responsible for the positive time direction (forward states) and a part for the negative time direction (backward states). Outputs from forward states are not connected to inputs of backward states, and vice versa. The BRNN can be trained without the limitation of using input information just up to a present future frame. This means that their future input information is reachable from the current state (Schuster & Paliwal, 1997).

However, there was still two major limitations of RNN models even with the bidirectional architecture; the vanishing and the exploding gradient problems. The exploding gradients problem is when there is a large increase in the norm of the gradient during training. This is due to the explosion of the long-term components, which can grow exponentially more than short-term ones. The vanishing gradients problem is the opposite issue, when long term components go exponentially fast to norm 0, making it impossible for the model to learn the correlation between temporally distant events (Bengio et al., 1994). LSTMs were introduced by Hochreiter & Schmidhuber (1997) and are capable of learning long-term dependencies. The key to LSTMs is the cell state, the horizontal line running through the top of the cell state. LSTMs, like RNN, have a chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, which interact different to an RNN. There is a cell state which runs straight down the entire chain with only some minor linear interactions. Important information can flow along it unchanged. In this way, the important information weight on predictions is unchanged (Hochreiter & Schmidhuber, 1997). There is a lot more to the structure of LSTMs (and the other NNs) and I will expand further on this in later sections. LSTMs are state of the art Neural Networks that perform highly in speech recognition, language modelling, translation, image captioning and forecasting.

2.2.1 Neural Network applications in forecasting

Using NN for forecasting was first done in 1964 by Hu and Root when trying to forecast weather patterns (Hu & Root, 1964). However, the research was limited at that time and the forecasting performance of the models has followed a similar timeline to the models mentioned in section 2.0 above. NN are unique from traditional econometric modelling like VARs and ARIMA modelling. Firstly, NNs learn from examples and capture relationships among the data even if the underlying relationships are unknown or hard to describe. This makes it unbiased to theoretical assumptions. However, there is a drawback to this approach, as the relationships can be masked by noise. NN’s generalise well as they learn from patterns in the data. This makes them useful when past data explains future forecasts. NN’s are also universal functional approximators which means they have more general and flexible functional forms than the traditional statistical methods can effectively deal with. Finally, NNs are nonlinear so they can learn patterns that ARIMA or VAR models cannot pick up. Because of all these functionalities NN’s are used for a range of forecasting applications in business and finance such as forecasting stock prices, bankruptcy and business failure and foreign exchange rates (Zhang et al., 1998).

There have been many applications of linear models to forecasting output based on financial variables in research and business. However due to the non-linear nature of financial indicators and economic output NN can provide better insights. Pradhan and Kumar (2008) compared a NN against a Linear Regression Model (LRM) on forecasting India’s economic growth driven by Foreign Direct Investment (FDI). Their findings conclude that NN can forecast economic growth much better than LRM and that linear unpredictability and non-linearity can be captured by NN. This provides India’s policymakers with insights on the effects of policies with respect to FDI on economic output in their country (Pradhan & Kumar, 2008). However, one needs to be careful when comparing NN models to traditional statistical approaches. NN are black boxes so it is very hard to understand why they are predicating the values they are outputting. NN are also built and have many hyperparameters compared to traditional models. There are a different number of layers, output size, input shape, activation functions, loss function, epochs and batch sizes all of which need to be considered when building an NN which makes each NN different.

ML models are also very dynamic and you can merge multiple models for the same forecasting tasks, these are called ensemble models. A classic ensemble model is the XGBoost decision tree model. Longo, Riccaboni and Rungi (2021) used an ensemble model to predict GDP one quarter ahead. They found when combining a ML model with a metric-based model that the model was able to build on advantages of both models that take part in different phases of the business cycle. This is deductive reasoning and there is no current metric to test this hypothesis. This is one of the draw backs of ML models that they are a black box and it is very difficult to say what is driving the predictions the model is outputting. Nevertheless, their results are encouraging and concluded that for forecasting the quarterly US GDP growth rates one quarter ahead in the period 2005Q2-2020Q1 that their ensemble model of a Dynamic Factor Model with a Generalized Autoregressive Score (DFM-GAS) and an RNN were able to outperform a range of other models. They focused on two errors of crisis the GFC in 2008–2009, and the most recent Covid-19 crisis. During the Covid -19 crisis their model produces good predictions during the economic rebound which they conclude is thanks to the neural network component (Longo et al., 2022). Forecasting GDP during downturns has long been an intractable problem for economists, current research is showing the NN can help them with this task.

LSTM models are the state-of-the-art NN models for using on sequential data. They are now starting to be applied to economic forecasting problems with encouraging results. There have been some investigations into applying LSTM models to economic and financial data. Namin and Namin (2018) compare ARIMA models to LSTM models for forecasting finance and economic data. They applied univariate models and found the LSTM had an average error reduction of 84 - 87 percent (Namin & Namin, 2018). This research shows the promise LSTM models hold however it was split into train and test data sets using a univariate model. Expanding on this area Hopp (2022) published a paper which looked at comparing 12 different modelling methodologies against each other for nowcasting US GDP. He compared these models across multiple economic events and is findings show that when you aggregate the results the LSTM model performed the best out of all 12 models (Hopp, 2022). Hopp's paper was published while writing this thesis and more literature will become available on this topic in the immediate future.

2.2.2 ML research on economic forecasting in NZ

NZ institutions do not have as much appetite for statistical models as overseas institutions. There has been very little research into the application of Machine Learning (ML) models to help improve forecasting performance in NZ. The only research I could find on ML with respect to the NZ economy was by the RBNZ where they investigated the performance of different ML algorithms in obtaining accurate nowcasts of real GDP growth. They conclude that the top-performing models-boosted trees, support vector machine regression and neural networks - can reduce average nowcast errors by around 20-23 per cent relative to the AR benchmark (Richardson et al., 2019). The only NN they applied was a Feed Forward Neural Network (FNN or ANN, refer to as an NN in this paper), the standard NN which are not the best NN architecture for time series data or for forecasting. There is significant value to be added by applying more up to date NN architectures such as LSTMs, on forecasting NZ economic indicators.

2.3 NN possible contributions to economics

NN are continually evolving and as their performance improves and they become more accessible through software developments and teaching, they will be applied to more fields. They are already being tested by researchers on economic applications and it is only a matter of time before economic and financial institutions start applying them. They will start to be applied in economics in time-series modelling and forecasting, nonparametric estimation, and learning by economic agent. In 2007 Kuan and White wrote:

> … not only do artificial neural networks have much to offer economics and econometrics, but there is also considerable potential for economics and econometrics to benefit the neural network field, arising to a considerable degree from economic and econometric experience in modelling and estimating dynamic systems. Thus, a larger goal of this article is to provide an entry point and appropriate background for those wishing to engage in the

fascinating intellectual arbitrage required to fully realize the potential gains from trade between economics, econometrics and artificial neural networks. (pp. 3)

Chapter 3: Methodology

3.1 Modelling objective

The aim of this thesis is to compare newly developed LSTM models forecasting ability of macroeconomic indicators against traditional models forecasting ability. The models will forecast over an eight timestep horizon which equates to two years starting from 2012 up until the end of 2019. Each of the models will loop through the data again after each forecast one timestep ahead and then forecast the next eight steps ahead. The output of the models will be a data frame from March 2012 to December 2019 with eight forecasts for each model for each quarter from March 2012 to December 2019. The forecasts will then be measured for each model and each quarter by RMSE to measure the accuracy of each forecast period and model and then compare them. To compare the forecasts of each model further and to see if there is a significant difference in forecasting accuracy at a 5% significance level, I will use a DM test.

3.1 VAR

Vector autoregression (VAR) models are models that capture linear interdependencies between multiple time series. VAR are a commonly used multivariate model for forecasting in economics. That means, the basic requirements in order to use VAR, like many timeseries model are:

1. You need at least two variables, an independent variable and a dependant variable.
2. There should be relationship between the so that they influence each other.

Each time series is a function of its past values which makes it an autoregressive model. What separates VAR models from other autoregressive models like ARIMA, ARMA or AR models is that VAR is a bi-directional model which means, as stated above, that the variables influence each other. In a VAR model is a linear model where each variable is modelled of past values of itself and the past values of other variables that are in the model. In a multivariate model there are multiple time series that influence each other, it is modelled as a set of equations with one equation per variable. If you have two variables that influence each other, then you have a set of two equations. These equations are illustrated below:

$$Y_{1,t} = \alpha_1 + \beta_{11} Y_{1,t-1} + \beta_{12} Y_{2,t-1} + \epsilon_{1,t}$$

$$Y_{2,t} = \alpha_2 + \beta_{21} Y_{1,t-1} + \beta_{22} Y_{2,t-1} + \epsilon_{2,t}$$

There are two time series variables Y_1 and Y_2, and these variables are forecasted values at time (t). To calculate Y_1 (t), VAR will use the past values of both Y_1 and of Y_2. To compute Y_2 (t), the past values of both Y_1 and Y_2 will be used. These equations are better represented in vector form below:

$$\begin{pmatrix} Y_{1,t} \\ Y_{2,t} \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} + \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix} \begin{pmatrix} Y_{1,t-1} \\ Y_{2,t-1} \end{pmatrix} + \begin{pmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \end{pmatrix}$$

$Y_{1,t-1}$ and $Y_{2,t-1}$ are the variables Y_1 and Y_2 with the first lag. This is a VAR(1) model, because each equation has a lag order of 1, that is, it contains up to one lag of each of the predictors Y_1 and Y_2. The Y terms in the equations are interrelated. Y_1 and Y_2 are endogenous variables not exogenous predictors because they are interrelated.

Before you can input data into a VAR model there needs to be a number of tests run and data manipulation applied. These steps are fundamental to the modelling process so it is important to think about them as part of the modelling methodology. Firstly, you can test for a causality relationship between the variables with a Granger's Causality Test. To test the null hypothesis, which is, the coefficients of past values in the regression equation do not influence Y. To test the null we use the Granger's causality test. This tests that the past values of time series X do not cause the series Y. You can reject the null hypothesis if the p-value obtained from the test is less than the significance level, usually of 0.05, and can state that X does not cause the other series Y. This step is a good starting point but you should not take the output as definitive, iteratively test the model to see if the exogenous predicator variables are reducing your RMSE (if you are using that metric of measure for model performance).

VAR models require data to be stationary, stationary time series is where the variables mean and variance do not change over time. A common test for stationarity is the Augmented Dickey Fuller Test (ADF). If the data is non-stationary, you difference the data to make it stationary. Differencing reduces the length of the data by how many times you difference it because you minus the observation from the next one. Because of this all the variables have to be differenced to the same degree. Once the data is stationary you then need to choose the optimal level of lags you want the VAR model to look back. Common practise here is to iteratively test the number of lags and choose the number that gives the lowest Akaike Information Criterion (AIC) measure.

3.2 VARMAX

Vector Auto-Regressive (VAR) model is a generalization of the auto-regressive model for multivariate time series where the time series is stationary and we consider only the lag order 'p' in the model. The Vector Moving Average (VMA) model is a generalization of the Moving Average Model for multivariate time series where the time series is stationary and we consider only the order of moving average 'q' in the model. The Vector Autoregressive Moving Average (VARMA) model is a combination of VAR and VMA models that helps in multivariate time series modelling by considering both lag order and order of moving average (p and q) in the model.

VARMA models require data to be stationary like VAR models so you can run a lot of the same data preprocessing steps as you do with VAR, such as cointegration tests, ADF Test and data differencing.

A VARMAX is an extension of a VARMA model but it has exogenous variables and is represented by the equation below.

$$y_t = v + \mathrm{A}_1 y_{t-1} + \cdots + \mathrm{A}_p y_{t-p} + \mathrm{B}\chi_t + \epsilon_t + \mathrm{M}_1 \epsilon_{t-1} + \cdots + \mathrm{M}_q \epsilon_{t-q}$$

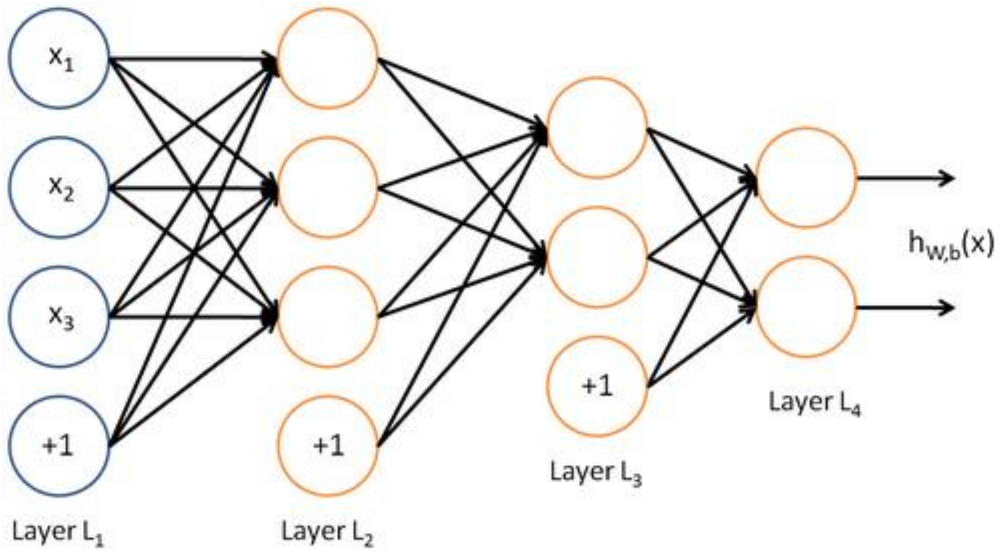
3.3 LSTM

To better understand the structure of an LSTM model it is best to start with the NN and then the RNN which LSTM models are an extension of.

3.3.1 NN

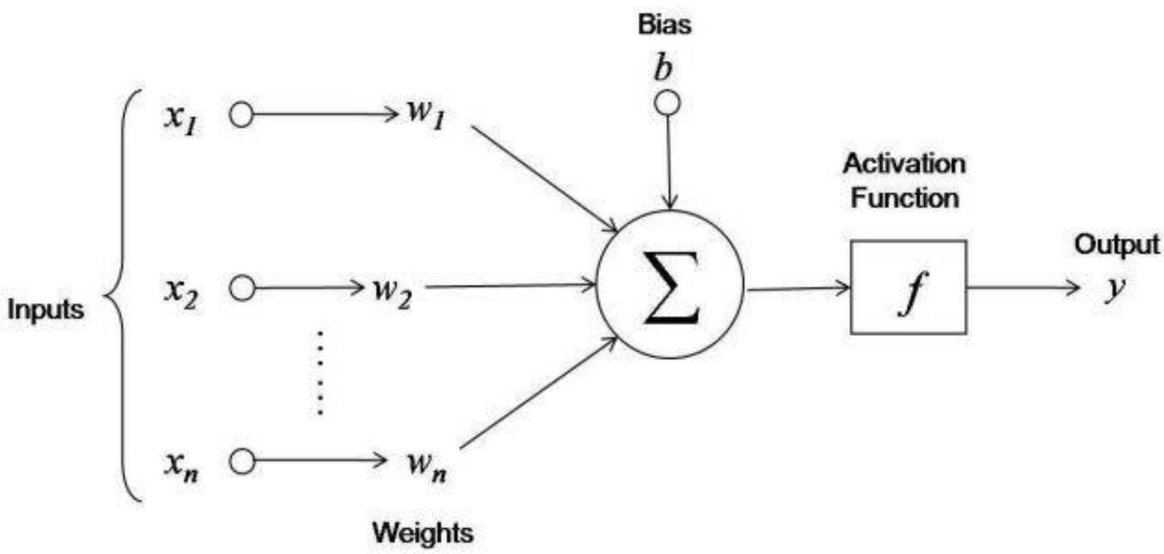
The original idea behind neural networks is that they are a modelled representation of our brains by having interconnected neurons. The operations are straight forward, there is input data which the NN applies mathematical operations to, to get an output which is the model's predictions. The figures are used from the articles First neural network for beginners explained (Arnx, 2019) and How Does Back-Propagation Work in Neural Networks? (Koech, 2022).

Figure 3.3.1 Recurrent Neural Network structure



NN can commonly be read from left to right when presented in diagrams, where X is the input data in the first layer and the hidden layers are where the weights are applied to the input data and then passed to the output layer. The +1 are the bias which are added to the total calculation of the layers. Figure 3.3.2 shows the operation done by the neurons.

Figure 3.3.2 NN neurons

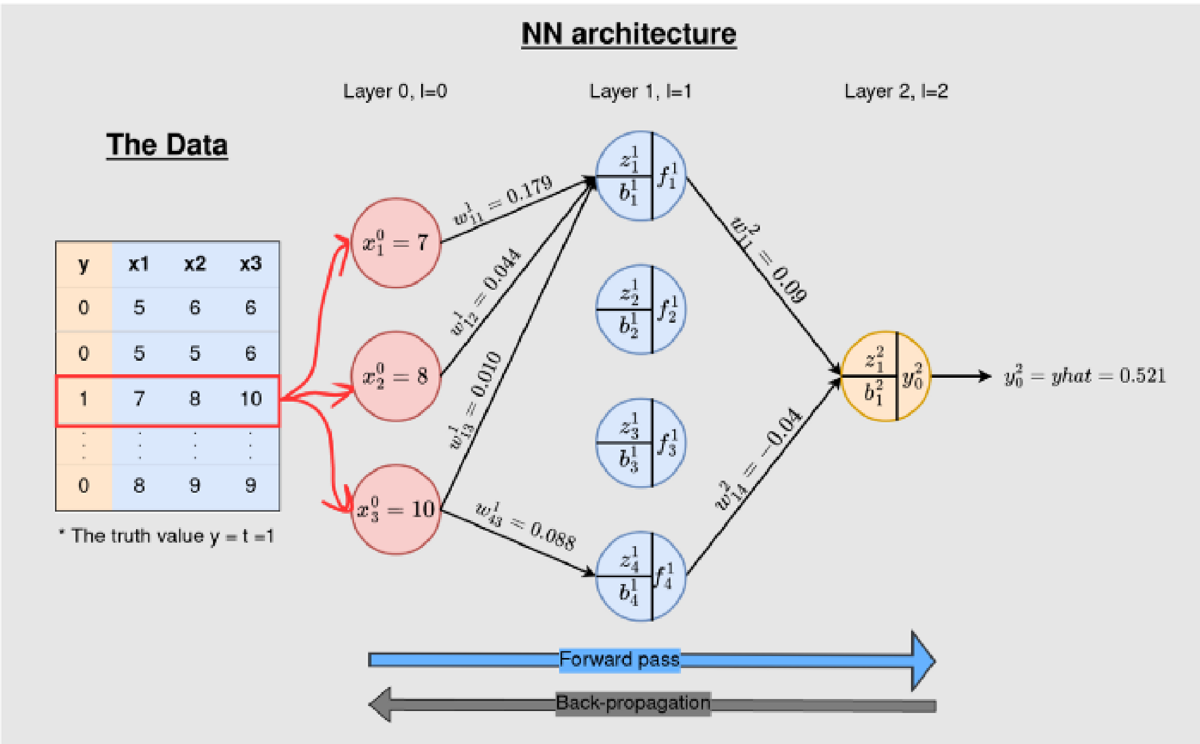


First, it sums up the value of every neuron from the previous layer it is connected to. These are the x inputs coming to the neuron, so 3 neurons of the previous layer that are connected to the neuron in figure 3.3.2. These values are multiplied and then summed by the weights which are the w in the image which determines the connection between the two neurons. There is a weight for every connection of neurons has its own value, and those are the values that will be updated during the learning process. A bias value may be added to the total value calculated. It is not a value specified within the neurons but rather is a hyperparameter set before the model is compiled when running the code, it can be useful for model predictions.

The weights are updated by an optimization function that uses back-propagation. Back-propagation is a method for supervised learning used by NN to update the weights to make the network's predictions more accurate. The parameter optimization process is achieved using an optimization algorithm called gradient descent. Unlike the forward movement for the model predictions, back-propagation works backward from the output layer to layer one. This is done by calculating the derivatives/gradients with respect to parameters for all layers using the chain rule of differentiation. My LSTM model uses Root Mean Square Propagation (RMSP) activation function which is explained in more detail in section 3.3.4.

An activation function is a function that is added into a neural network which adds a threshold, which if reached, updates the weights of the Neural Network. This helps the model learn complex patterns within the data. The activation function is at the end of each layer deciding if the information is to be sent to the next layer of neurons. The activation function receives the output from the previous neuron in the layer before and takes the output and converts it into a form that is then taken as input into the neuron in the next layer. Activation function constrains the value of the output from the neuron before within a limit specific to the type of activation function. That is why it is important to choose the right activation function for the task for which you are using your Neural Network. It is good practice to test multiple types of activation function. There are some general guidelines that can be followed which can be found by reviewing online models used for similar tasks for which you are trying to complete. Activation functions also adds non-linearity into a neural network. My LSTM model uses a rectified linear activation function (ReLU) which is explained in more detail in section 3.3.4.

Figure 3.3.3 Detailed NN structure

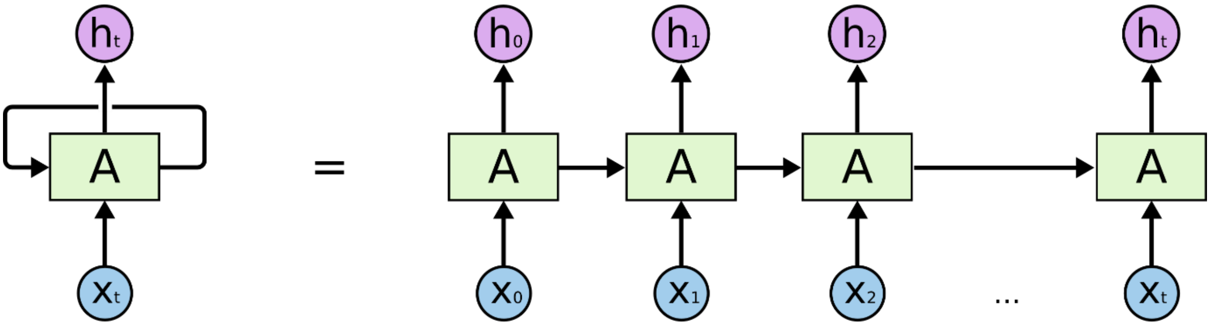


3.3.2 RNN and LSTM

Traditional neural networks cannot process sequential information but the RNN can do this by having loops in their structure. The figures are used from Colah's Blog [Colah's Blog, 2015].

Figure 3.3.4 Recurrent Neural Network structure

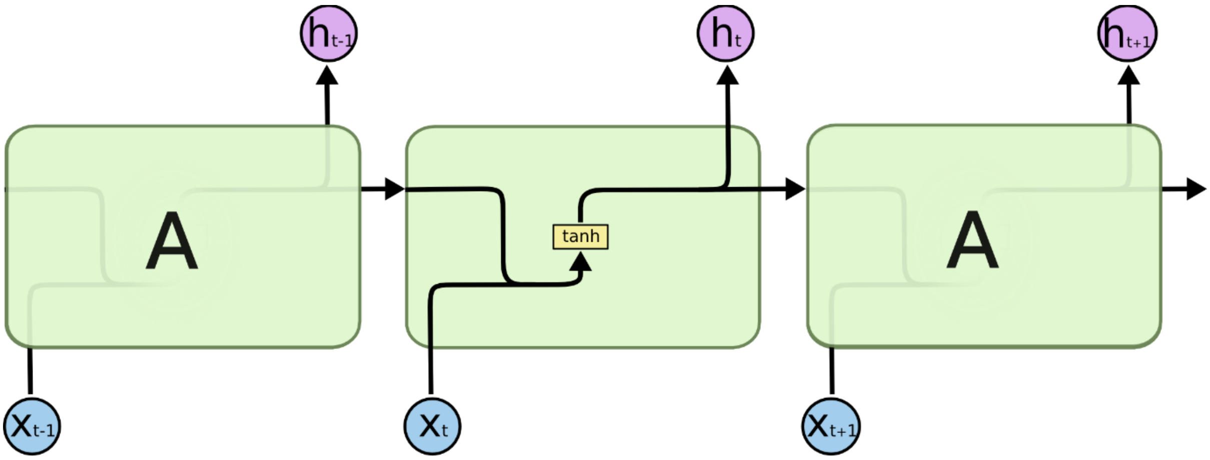
Recurrent Neural Network



Their chain-like structure shows that recurrent neural networks are built to process sequence data. The structure of the RNN is represented as A in the diagram, X is the input data into the RNN at a timestep and h is the output of the RNN in a timestep. This structure makes them the ideal model for learning patterns in data where past data points are related to future data points such as time series data. However, the limitations in this architecture discussed in chapter 2 about long term dependencies can affect RNN performance at forecasting GDP. For example, if the HPI is inflated leading up into 2008 and then GDP crashes due to an economic recession like in the 2008 GFC the RNN model will have trouble learning this long-term dependency because it happened further back in the time series data. Because RNN do not learn long term dependencies well LSTM models were invented.

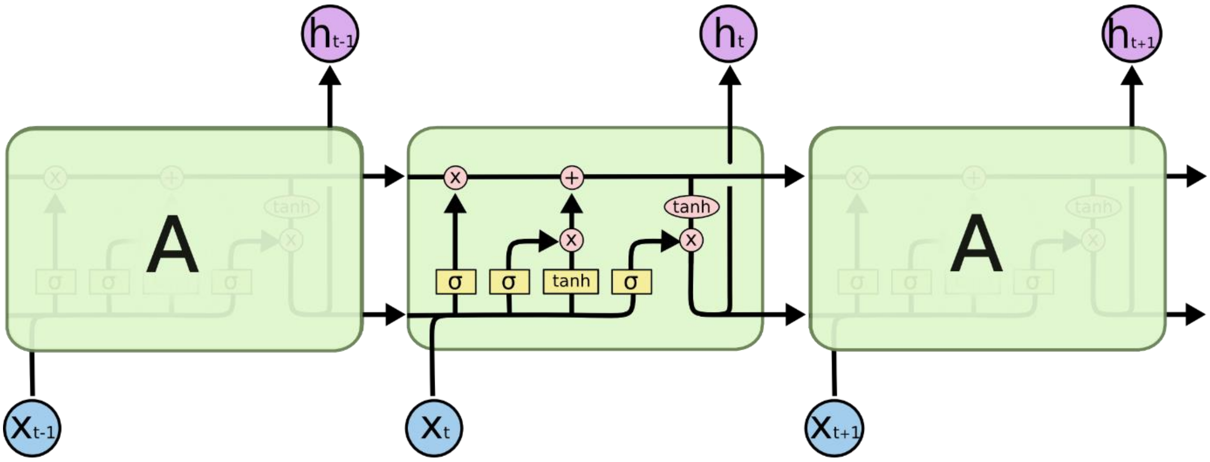
LSTMs were designed to deal with the long-term dependency problem, which is done within the structure of their neurons. RNN have a single tanh activation function within their neurons like in Figure 3.3.5.

Figure 3.3.5 RNN with Activation function



LSTMs have the same structure as RNN, but the neurons that loop have a different structure. There are four gates controlled by sigmoid layer which quantify how much information they let through with a 0 to 1 which update and control the cell state. This is show in figure 3.3.6 below.

Figure 3.3.6 LSTM cell state

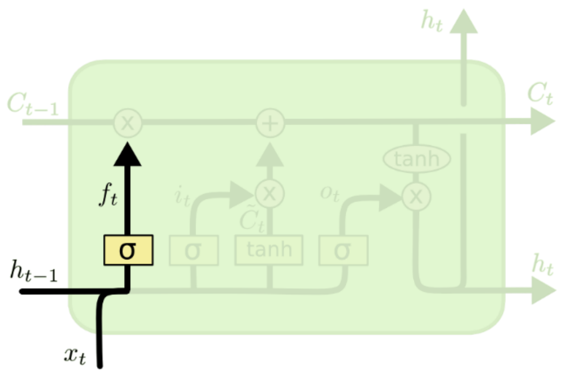


In this layer is where the mathematical operations happen that affect the models output. The main innovation of an LSTM is the cell state which is the line above running through the middle neuron. This allows old information to be passed through the network without being updated by the activation function, there by eliminating the issue of the vanishing or exploding gradient problem. The LSTM has gates that allow information to pass through this cell state. These gates are designed to optimally let information into this cell state. They use a sigmoid neural net layer and a pointwise multiplication operation. A number between zero and one is output from the sigmoid layer which is how much information from each component should be sent through. A value of zero means no information will be passed through while a value of one means all the information will be passed through. An LSTM regulates the cell state with three gates.

3.3.3 Inside look at the cell state

Taking a look into the cell state gives a good insight into the linear algebra that the LSTM model is built on. This information helps guide you when constructing an LSTM model for a particular task. First the LSTM decides which information from the cell state to remove, this is called the 'forget gate'. It looks at h_{t-1} and X_t, a number between 0 and 1 is outputted for each number in the cell state C_{t-1}. If it outputs a 1 it keeps all the information and if it output 0 it discards all the information.

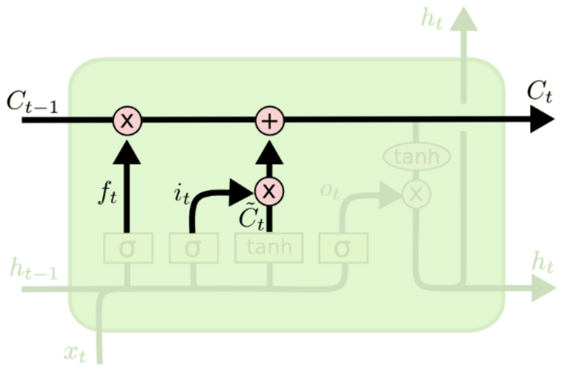
Figure 3.3.7 LSTM forget gate



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

The LSTM updates the old cell state C_{t-1} to the new cell state C_t. This is done by multiplying the old cell state by f_t which has dropped the information that it formulated earlier. The LSTM then adds $i_t * \tilde{C}_t$. This is the new value based on how much the LSTM decided to update each state value. This is illustrated in Figure 3.3.8 below:

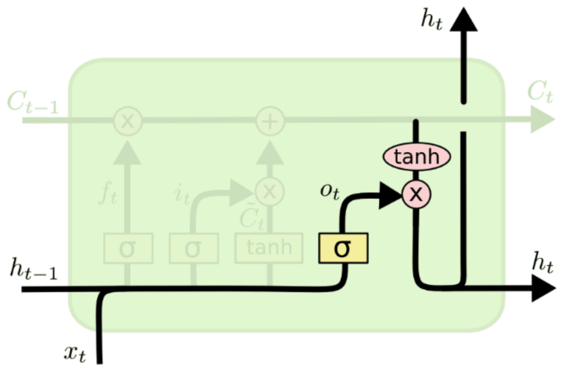
Figure 3.3.8 LSTM drop gate



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

The last step of the LSTM is the output which is the information passed to the next cell state or the output layer depending on where the information is within the LSTM model. The output is filtered information from the cell state. The LSTM decides which part of the cell state is outputted by the sigmoid layer which is run first. Then the cell state will be put through tanh function (this scales the values between −1 and 1) and multiplied by the output of the sigmoid gate, so it outputs only the information from the model that the LSTM wants to.

Figure 3.3.9 LSTM output gate



$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

This is an overview of a generic LSTM cell state. A real-world application of this would be if an LSTM model was processing data to identify an area in a city. If the image contained the Brooklyn bridge from data processed in the earlier layers and then processed images with shops with lots of Chinese characters the model could allow the vector with the Brooklyn bridge in it to pass through the other layers not having the importance of the information diminished by the other layers weights. This would allow the model to retain the information that the suburb is in New York and most likely Chinatown not a suburb in a city in China.

3.3.4 LSTM activation function and Adam optimization.

The rectified linear activation function (ReLU) activation function is a mathematically simple function that is highly effective; f(x)=max(0,x). It is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many neural networks because the models that use it are easier to train and often achieves better performance. LSTMs use hyperbolic tangent function (tanh) as their default. It is a similar shaped nonlinear activation function that outputs values between -1.0 and 1.0. However, there are some limitations to tanh functions as they tend to saturate and large numbers snap to 1 and small numbers snap -1. Once saturated it becomes hard for the model to update its weights to improve performance.

Adaptive Moment Estimation (Adam) is an optimization technique for gradient descent. It is a combination of the gradient descent with momentum algorithm and the Root Mean Square Propagation (RMSP) algorithm. The momentum algorithm accelerates the gradient descent algorithm by taking into consideration the exponentially weighted average of the gradients. Using averages makes the algorithm converge towards the minimum faster.

$$w_{t+1} = w_t - \alpha m_t$$

$$m_t = \beta m_{t-1} + (1-\beta)\left[\frac{\delta L}{\delta w_t}\right]$$

RMSP is an adaptive learning algorithm that tries to improve AdaGrad by taking the 'exponential moving average'.

$$w_{t+1} = w_t - \frac{\alpha_t}{(v_t+\varepsilon)^{1/2}} * \left[\frac{\delta L}{\delta w_t}\right]$$

$$v_t = \beta v_{t-1} + (1-\beta) * \left[\frac{\delta L}{\delta w_t}\right]^2$$

Adam Optimizer uses the positive attributes of the above two methods and builds upon them to give a more optimized gradient descent. Combining the features of the above methods to reach the global minimum efficiently. The formula for Adam is:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\left[\frac{\delta L}{\delta w_t}\right] v_t = \beta_2 v_{t-1} + (1-\beta_2)\left[\frac{\delta L}{\delta w_t}\right]^2$$

Since mt and vt have both initialized as 0 it is observed that they tend to be biased towards 0 as both β1 & β2 ≈ 1. This Optimizer fixes this problem by computing bias-corrected mt and vt. This is also done to control the weights while reaching the global minimum to prevent high oscillations around the minimum. The formulas used are:

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t} \widehat{v}_t = \frac{v_t}{1-\beta_2^t}$$

Adam optimizer is adapting to the gradient descent after every iteration so that it remains controlled and unbiased throughout the process. Instead of our normal weight parameters mt and vt, we take the bias-corrected weight parameters (m_hat)t and (v_hat)t. This gives the general equation below:

$$w_{t+1} = w_t - \widehat{m_t}\left(\frac{\alpha}{\sqrt{\widehat{v_t}+\varepsilon}}\right)$$

Adam optimizer gives a higher performance than other optimizers when performing gradient descent which is why I chose to use it in my model.

3.4 Evaluation methods

I will compare the models forecasting performance by using Root Mean Squared Error (RMSE) and the Diebold–Mariano (DM) test.

3.4.1 RMSE

RMSE is used as the errors and random variables might have Gaussian distribution with the mean μ and the standard deviation σ. $\hat{y}_i$ is the exact distance between two points. In, my models those two points will be the models forecasted GDP and the actual GDP, while y_i is the forecasted distance between the two points above with errors from miscalibrations and measurement noise. The mean of the distribution of errors corresponds to a persistent bias coming from miscalibration. The standard deviation corresponds to the amount of measurement noise. If the mean μ of the distribution for the errors is known then you can derive the formula for the standard deviation. I want the model with the smallest errors so I will add the bias-variance decomposition of mean forecast error (μ^2) to understand where the forecasting errors come from. Therefore, it allows me to search for the best model specification to minimize the forecast a tool to tune models and to measure the accuracy of trained models. The smaller the RMSE the better the trained model's prediction accuracy, the higher the RMSE the worse the trained model is at forecasting. RMSE is a standard way to measure the error of a model in predicting quantitative data. The formula for RMSE is:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y_i} - y_i)^2}{n}}$$

Where $\widehat{y_1}$ to $\widehat{y_n}$ are the predicted variables and y_1 to y_n are the observed variables and n is the number of observations.

The RMSE is a measurement that can be thought of as a tool for measuring distance between the predicted values and the observed values in vector form.

$$y_i = \hat{y_i} - \epsilon_i$$

The errors and random variables might have Gaussian distribution with the mean μ and the standard deviation σ. $\hat{y_i}$ is the exact distance between two points. In, my models those two points will be the models forecasted GDP and the actual GDP, while y_i is the forecasted distance between the two points above with errors from miscalibrations and measurement noise. The mean of the distribution of errors corresponds to a persistent bias coming from miscalibration. The standard deviation corresponds to the amount of measurement noise. If the mean μ of the distribution for the errors is known then you can derive the formula for the standard deviation. I want the model with the smallest errors so I will add the bias-variance decomposition of mean forecast error (μ^2) to understand where the forecasting errors come from. Therefore, it allows me to search for the best model specification to minimize the forecast errors.

$$var = \mathbb{E}\left[\frac{\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}{n}\right]$$

$$= \mathbb{E}\left[\frac{\sum_{i=1}^{n}{\epsilon_i}^2}{n}\right]$$

$$=\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[{\epsilon_i}^2\right]$$

$$= \mathbb{E}[\epsilon^2]$$

$$= var(\epsilon) + \mathbb{E}[\epsilon]^2 = \sigma^2 + \mu^2$$

$\mathbb{E}$ is the expectation and var is variance. We can replace the average of the expectations $\mathbb{E}[{\epsilon_i}^2]$ with $\mathbb{E}[\epsilon^2]$ where ϵ is a variable that has the same distribution as each of the ϵ_i, this is because the errors ϵ_i are identically distributed, therefore all their squares have the same expected value. We know μ which is a known bias. We subtract μ from our observations giving our errors a distribution of $\mu = 0$. Put this into the equation and the square root of both sides gives the formula below:

$$\sqrt{\mathbb{E}\left[\frac{\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}{n}\right]} = \sqrt{\sigma^2 + \mu^2} = \sigma$$

To get the RMSE formula, remove $\mathbb{E}$ from the left-hand side formula and we get the formula for the RMSE. The central limit theorem tells us that as n gets larger, the variance of the quantity should converge to a normal distribution asymptotically. The n under the square root in RMSE allows us to estimate the standard deviation σ of the error for a typical single observation rather than a total error. To keep the measure of the error consistent by dividing by n. This keeps the output more accurate as you move from a smaller dataset to a larger one with more observations.

There is no specific cut off for RMSE and the units used in the forecast are percentages. Therefore, when interpreting RMSE I will use it for comparative purposes because it is a common measurement of the same units as the forecasts and the forecasted variables. This makes it useful for not only comparing models but also for deciding if the model forecasts are producing any useful insights.

3.4.2 DM Test

The DM test is used to compare two different forecasts. It does this by evaluating a null hypothesis that there is no difference in the accuracy of two competing forecasts. It does this by comparing the loss of the forecast errors defined as $e_{it} = \hat{y}_{it} - \mathrm{yt}, \mathrm{i} = 1, 2$. The loss associated with forecast i is assumed to be a function of the forecast error, e_{it}, and is denoted by $g(e_{it})$. g is a loss function that never takes a negative number, takes a value of 0 when no error is made and increases in size as the error gets larger. The loss differential between the two forecasts is:

$$d_t = g(e_{1t}) - g(e_{2t})$$

The forecasts have equal accuracy when $d_t = 0$. The null hypothesis is that the two forecasts have the same accuracy. The alternative hypothesis is that the two forecasts have different levels of accuracy. The test can be laid out mathematically in the below example:

If you have the quantity $\sqrt{T}(\bar{d} - \mu)$ where $\bar{d}$ is the sum of d_t from t=1 until T and μ is the mean of the loss differential and $\mu = E(d_t)$ is the population mean of the loss differential. The spectral density of the loss differential at frequency 0 is:

$$f_d(0) = \frac{1}{2\pi}\left(\sum_{k=-\infty}^{\infty} \gamma d(k)\right)$$

$\gamma d(k)$ is the autocovariance of the loss differential at lag k. You can show if the loss differential series is covariance stationary and short memory, then:

$$\sqrt{T}\left(\bar{d} - \mu\right) \to N(0, 2_{\pi} f_d(0)$$

$$\frac{\bar{d}-\mu}{\sqrt{\frac{2\pi f_{d}(0)}{T}}} \to N(0,1)$$

Under H_0: $\frac{\bar{d}}{\sqrt{\frac{2\pi f_{d}(0)}{T}}} \to N(0,1)$

For the forecasts are $h(> 1)$-step-ahead. To test the null hypothesis that the two forecasts have the same accuracy, the DM test uses the following statistic:

$$DM = \frac{\bar{d}}{\sqrt{\frac{2\pi f_{d}(0)}{T}}}$$

Where $\hat{f}_d(0)$ is a consistent estimate of $f_d(0)$ which is defined by:

$$\hat{f}_d(0) = \frac{1}{2\pi}\widehat{\gamma_d}(0)$$

for h ≥ 1

$$DM = \frac{\bar{d}}{\sqrt{\frac{\widehat{\gamma_d}(0) + 2\ \sum_{k=1}^{h-1}\widehat{\gamma_d}(k)}{T}}}$$

In practice, using:

$$\sum_{k=-M}^{M}\widehat{\gamma_d}(k)$$

Where $M = T^{1/3}$ provides an adequate estimator of $2\pi f_d(0)$ therefore:

$$DM = \frac{\bar{d}}{\sqrt{\frac{\sum_{k=1}^{m} - m\widehat{\gamma_d}(k)}{T}}}$$

Under the null hypothesis, the test statistics DM is asymptotically N(0, 1) distributed. The null hypothesis of no difference will be rejected if $|DM| > Z\_(\frac{a}{2})$.

Chapter 4: Data

All the variables used were collected from Statistics New Zealand for GDP, NZ's official data agency and the Reserve Bank of New Zealand (RBNZ) for import and export prices, House Price Index (HPI) and Overnight Interbank Cash Rate (OICR), Consumer Price Index (CPI) and Unemployment. Statistics New Zealand collects census, demographic, economic and a range of other data which made them the default data repository choice. Their data is often reviewed and updated.

Table 4.1.1

Variable	Mean	Min	Max	Std	Lag upon release
GDP	0.709	-2.42	2.8	0.744	3 month lag
Unemployment	6.062	3.3	11.2	1.924	1 month lag
CPI	2.174	-0.5	7.6	1.473	1 month lag
Import Prices	0.364	-11.9	10.6	3.712	2 month lag
Export Prices	0.058	-7.5	9.6	3.050	2 month lag

I use pseudo-real time data to train the models in this thesis. The data is taken from the time the forecasts are made but the data has been retrospectively updated as the data agencies have had more time to collect and interpret the data. This is important for later sections of this thesis when I compare to ANZs forecasts because they would have only had real time data when doing their forecasts.

Like any ML task the data you use is as important as the model itself so I will go through and discuss each of the variables which were used in the models.

4.1 GDP

The explanatory variable is quarterly percent movement in GDP. There is a two and half month delay upon release of data. I calculated this by getting Series, GDP(P), which was last updated 16 June 2022. The framework used to compile GDP in NZ is based on the System of Nation Accounts 2008 which is jointly published by international institutions such as The World Bank and United Nations The annual GDP numbers are revised quarterly and can cause the values to change. Average quarterly GDP growth is calculated quarter over quarter growth rates. For example, GDP 2019 Quarter 2 growth rate would equal GDP 2019 Quarter 2 minus GDP 2019 Quarter 1 divided by GDP 2019 Quarter 1 (GDP2019Q2_gr = (GDP2019Q2 - GDP2019Q1)/ GDP2019Q1). This provides a better indicator of trends in annual growth than measuring current quarter to previous quarter. This variable will be in my model and will be the variable my models are forecasting.

4.2 Unemployment

Unemployment is measured by the Household Labour Force Survey (HLFS). There is a one-month delay in employment data. HLFS statistics are published approximately 5 weeks after the end of quarter. I am using the current unemployment rates against current GDP. Okun's law states that when Unemployment goes up by 1% output will fall by about 2%. Because of unemployment's close association with GDP, it will be used as an independent variable because it has improved the model's predictive capability.

4.3 Consumer Price Index (CPI)

The CPI measures the changing price of a basket of goods and services purchased by NZ households. It is the primary measure of inflation. There is a one-month delay in the CPI data. The CPI is used to help set monetary policy and for monitoring economic performance. The Phillips curve states that inflation and unemployment have a stable and inverse relationship. The theory purports that with economic growth comes inflation, which leads to more jobs and less unemployment. Because of inflation's relationship with unemployment which is related to GDP and inflation's relationship to GDP it is an important variable for forecasting GDP.

4.4 Import and Export Prices

The Overseas Services Trade Price Indexes measure changes in the levels of prices of imports and exports of services trade to and from New Zealand. There is a two-month delay in the Import and Export Prices data. A country's importing and exporting activity influences its GDP, inflation, interest rates and its exchange rate. The Balance of Trade is in the GDP calculation formula.

4.5 House Price Index (HPI) and Overnight Interbank Cash Rate (OICR)

The HPI was tested in the model because increases in house prices show increased household wealth which increases aggregated demand which leads to GDP growth. Increased household wealth has a close relationship with output.

OICR[1][2] was also tested in the model because it is directly related to interest rates. Increases in interest rates decreases consumption and investment both of which are components in the GDP calculation. HPI and OICR did not add to the forecasting accuracy of the model so were dropped.

HPI and OICR are sourced from the Reserve Bank of New Zealand (RBNZ) and there is a two-and-a-half-month delay of publication of the HPI from RBNZ.

4.6 Data pre-processing

The variables used to train the models are shown to be all stationary except for unemployment. I used an Augmented Dickey-Fuller Test (ADF Test) to identify stationarity. The ADF Test states a null hypothesis that the data has a unit root and is non-stationary, if the p-value is less than 0.05 at a 5% significance, which I've used, then we can reject the null hypothesis and can conclude that the data is stationary with 95% confidence. I differenced the data to make it stationary for the VAR and VARMAX models. A function is used to inverse transform the predictions back to interpret the model output in terms of growth rates. The LSTM models can learn non-linear patterns and do not require the data to be differenced so no differencing was done for the LSTM model. Table 4.6.1 shows the output of the ADF Test.

Table 4.6.1

Augmented Dickey-Fuller Test				
Variable	**Test Statistic**	**Critical value 5%**	**P-Value**	**Stationarity**
QQ_GDP	-6.8615	-2.885	0	Stationary
Unemployment	-1.7015	-2.885	0.4303	Non-Stationary
CPI	-3.0778	-2.887	0.0282	Stationary
Export prices	-7.8082	-2.885	0	Stationary
Import prices	-5.5712	-2.886	0	Stationary

[1] OICR and HPI are highly correlated

[2] Overnight interbank cash rate (OICR) was used instead of the Official Cash Rate (OCR) because the RBNZ data only went back to March 1999 for the OCR

4.7 LSTM data setup

The LSTM model takes an input shape which is a 3d array where the first dimension is the batch size, the second dimension is time steps and the third dimension is the units of one input sequence. The batch size is the number of samples the model can process before the model updates its weights. I did not specify a batch size so the model had the flexibility to update its weights when it was fit to do so. The time steps I specified as 8 for the models in the results section 5 so it would take the last two years of data. For the third-dimension, units, I specified the number of variables being passed to the model in the dataset so the model had the power to process all the data and update its weights based on all the information provided in the dataset. For the output shape I specified the length of 16 so the model provides predictions for the test dataset passed through which has 8 timesteps plus the 8 quarters ahead of the test data passed to the model for predictions. This is what allows the LSTM model to predict 8 quarters ahead. I set the return sequences hyperparameter to true which means the model provides output for each timestep specified.

Chapter 5: Results

In this section I will display and interpret the models and the forecasts for the pre-Covid-19 period (from March 2012 up until December 2019). In Chapter 6 I will expand the models over the Covid-19 period (up to September 2022). Finally in Chapter 7 I will then make a comparison of the LSTM models to the ANZ forecasts.

5.1 Estimates

The VAR model for pre-Covid-19 predictions uses four quarters of lagged values for each model. The lags were chosen by looping multiple VAR models with different lags from one to nine and then I chose the model with the lowest Akaike Information Criterion (AIC) number and the lowest Bayesian Information Criterion (BIC) number. The VAR models were very volatile so I chose the most stable model for each forecast period. The VARMAX model uses a lag value of 7 and a moving average window of 1. An Auto-ARIMA model was used to find the optimal values for the lags and the moving average window.

The VAR and VARMAX models create a forecast for each variable in the model. Because the models are re-estimated every loop, each model will have different coefficient values for the estimators. I have displayed the model summaries for the final looped models. Below are the estimators for the VAR of the GDP forecast (QQ_GDP).

Table 5.1.1

VAR QQ_GDP Model				
	Estimate	Std Error	t-stat	P-value
const	0.037	0.058	0.64	0.522
CPIL1	0.092	0.098	0.938	0.348
CPIL2	0.031	0.096	0.325	0.745
CPIL3	-0.177	0.094	-1.891	0.059
CPIL4	0.151	0.090	1.684	0.092
Export pricesL1	-0.002	0.022	-0.080	0.936
Export pricesL2	0.022	0.024	0.929	0.353
Export pricesL3	0.000	0.025	-0.016	0.987
Export pricesL4	0.023	0.023	0.995	0.320
Import pricesL1	-0.080	0.027	-2.959	0.003
Import pricesL2	-0.117	0.033	-3.533	0.000
Import pricesL3	-0.086	0.033	-2.556	0.011
Import pricesL4	-0.037	0.028	-1.304	0.192
QQ_GDPL1	-0.654	0.092	-7.081	0.000
QQ_GDPL2	-0.296	0.103	-2.873	0.004
QQ_GDPL3	-0.374	0.104	-3.606	0.000
QQ_GDPL4	-0.265	0.089	-2.970	0.003
UnemploymentL1	-0.030	0.218	-0.140	0.889
UnemploymentL2	-0.333	0.212	-1.573	0.116
UnemploymentL3	0.520	0.214	2.432	0.015
UnemploymentL4	0.175	0.225	0.779	0.436

Table 5.1.1 shows that lagged GDP is the strongest predictor in the model having a negative impact on GDP forecasts for the all the lags and are all significant. Lag 1 and 2 of Unemployment has a

negative impact on forecasts but are not significant at a 5% significance level. Lag 3 and 4 of unemployment has a positive impact on GDP forecasts but are not significant at a 5% significance level. Import prices have a negative relationship with GDP forecasts and are significant from lag 1 to 3. This is consistent with economic theory that as the price of raw materials increases then GDP will decrease. All the other estimators are not significant.

Below Table 5.1.2 is the VARMAX estimators.

Table 5.1.2

VARMAX QQ_GDP Model				
	Estimate	**Std Error**	**t-stat**	**P-value**
intercept	0.0785	0.101	0.781	0.435
CPIL1	0.3222	0.075	4.289	0
CPIL2	0.1677	0.097	1.728	0.084
CPIL3	-0.0737	0.115	-0.641	0.522
CPIL4	0.1623	0.095	1.713	0.087
CPIL5	0.2331	0.087	2.672	0.008
CPIL6	0.1542	0.103	1.498	0.134
CPIL7	0.1473	0.1	1.475	0.14
e(CPI)L1	-0.188	0.076	-2.478	0.013
Export pricesL1	0.1332	0.072	1.855	0.064
Export pricesL2	0.0955	0.053	1.793	0.073
Export pricesL3	0.1107	0.063	1.766	0.077
Export pricesL4	0.0964	0.063	1.541	0.123
Export pricesL5	0.0418	0.048	0.863	0.388
Export pricesL6	0.0261	0.059	0.444	0.657
Export pricesL7	0.069	0.043	1.602	0.109
e(Export prices)L1	-0.1458	0.074	-1.961	0.05
Import pricesL1	-0.2252	0.057	-3.957	0
Import pricesL2	-0.2349	0.06	-3.911	0
Import pricesL3	-0.2308	0.048	-4.832	0
Import pricesL4	-0.1091	0.066	-1.65	0.099
Import pricesL5	0.0295	0.056	0.532	0.595
Import pricesL6	-0.0063	0.059	-0.107	0.915
Import pricesL7	0.027	0.049	0.548	0.584
e(Import prices)L1	0.153	0.107	1.432	0.152
QQ_GDPL1	-0.8432	0.08	-10.6	0
QQ_GDPL2	-0.3043	0.088	-3.463	0.001
QQ_GDPL3	-0.2625	0.113	-2.329	0.02
QQ_GDPL4	-0.1559	0.1	-1.555	0.12
QQ_GDPL5	0.0395	0.079	0.5	0.617
QQ_GDPL6	0.3637	0.083	4.397	0
QQ_GDPL7	0.3093	0.124	2.498	0.012
e(QQ_GDP)L1	0.2679	0.078	3.425	0.001
UnemploymentL1	0.3475	0.051	6.844	0

UnemploymentL2	-0.2782	0.093	-3.007	0.003
UnemploymentL3	0.1175	0.053	2.216	0.027
UnemploymentL4	0.0464	0.102	0.457	0.647
UnemploymentL5	0.773	0.08	9.638	0
UnemploymentL6	0.5152	0.059	8.781	0
UnemploymentL7	-0.1215	0.089	-1.368	0.171
e(Unemployment)L1	-0.67	0.078	-8.591	0

Table 5.1.3

Summary Statistics	VAR	VARMAX
Observations	114	114
Log likelihood	-876.821	-712.548
AIC	2.174	0.154
BIC	4.613	2.674
HQIC	3.165	1.176

Table 5.1.2 shows that lagged GDP is significant in the VARMAX model at a 5% significance level up until lag 3 and have a negative impact on GDP forecasts. Lag 4 and 5 are not significant and then lag 6 and 7 are significant for GDP and have a positive impact on forecasts. The moving average coefficient is also significant and has a positive impact on predictions. All of the lag 1 variables are significant at a 5% level except export prices. Only Imported prices is significant at a 5% level in the lag 1, 2 and 3 variables which has a negative impact on GDP which is consistent with the balance of trade which is exports minus imports. When Imports exceed Exports then there is a trade deficit which decreases GDP. This represents an outflow of funds from New Zealand. However, a high level of imports does represent a robust economy with increased demand as firms look to grow. The type of imports also matters, if the imports are productive machinery, then this will increase efficiency and production in the future, increasing GDP. It has a negative impact on GDP. None of the lagged 4 variables have significance at a 5% level. Unemployment has a positive impact on GDP at lag 2, 5 and 6 at a 5% significance level which is inconsistent with Okun's law. None of the 6 and 7 lagged variables are significant at a 5% level. The coefficients values are smaller for the VARMAX mode compared to the VAR model which shows up in the stability of the predictions compared to the VAR model. A change in the economic indicators will likely cause a larger change in the VAR forecast than it will for the VARMAX forecast.

The VAR and VARMAX models were differenced because unemployment variable was non-stationary. I ran the models where the only differenced variable was unemployment and the rest of the variables were not differenced. The RMSE performance is roughly the same as when I differenced all the data. The RMSE was larger for the first two horizons and then improved the further out I was forecasting with the best quarters being 3, 5 and 8. Usually, multi horizon models predict better in the earlier quarters and then the further out the horizon is, the more the forecast performance deteriorates. Below is a table comparison of the VAR RMSE for the model with difference data and the model with unemployment only differenced and the other variables not differenced. I have chosen to use the model where all the variables are differenced because the earlier time horizons are better, which is generally the area of most interest.

Table 5.1.4

RMSE		
	VAR Diff	VAR UE Diff
t-1	0.491	0.525
t-2	0.458	0.531
t-3	0.452	0.443
t-4	0.500	0.501
t-5	0.491	0.433
t-6	0.474	0.460
t-7	0.458	0.463
t-8	0.506	0.432

LSTM and Neural Network models are often referred to as black box models (refer to Appendix B for a more detailed explanation). Table 5.1.5 below is an output summary of the LSTM model built for this thesis. It has three layers listed below as bidirectional_38, lstm_77 and dense_38. The first layer is the input layer that takes in an input shape of 8 and outputs to the next layer with a shape of 240 (number of units) the next layers output shape is 120 and the final layers shape is 16. The first layer has 120,000 weights and bias, the middle or hidden layer has 173,280 weights and bias and the final output layer has 1,936 weights and bias. The model has a total of 295,216 weights and bias.

Table 5.1.5

Model: "sequential_38"

Layer (type)	Output Shape	Param #
bidirectional_38 (Bidirectio	(None, 8, 240)	120000
lstm_77 (LSTM)	(None, 120)	173280
dense_38 (Dense)	(None, 16)	1936

Total params: 295,216
Trainable params: 295,216
Non-trainable params: 0

5.3 Models

LSTM models have a lot more hyperparameters compared to traditional models like VAR where you generally only need to consider how many lags you want your model to use. When building an LSTM for time series prediction you have to set the look back period which is how far back you want past values to influence current predictions, this will define the input shape into the LSTM. You then have to decide how many layers your LSTM model will have. This is how many layers the data is passed through and manipulated by the matrix multiplications above in the cell states. You also have to set how many units each LSTM layer will use which is the dimension of the inner cells in LSTM. You then have to set the activation function for each layer. LSTM by default use tanh activation functions and sigmoid activation functions however you may want to try Relu Activation functions. The activation functions are how the model learns from the data by whether or not to update the weights as information is passed through the model. The output of the model also needs to be defined which is very important for forecasting tasks, this is how many timesteps the model will predict based on the input shape. Finally, you need to compile the model with an optimization function and the loss function which is the metric the optimizer is trying to minimise. There are a lot more hyperparameters that are used in LSTM that are too many to cover (Refer to the Appendix C for the code used to build the LSTM models and set the hyperparameters). I have just listed the main hyperparameters that all modellers have to consider when building an LSTM model. They have a complicated architecture that will likely be unique to each task the LSTM is trying to solve.

The LSTM model I have built for pre-Covid forecasts has three layers with two being LSTM state cells and one output later. Because my time series problem does not have much data the model only needs two LSTM layers and an output layer. The more layers there are the harder the model is to train. The activation functions are Relu activation functions. Relu functions were chosen because they allow the model to learn nonlinear patterns and do not suffer from limited sensitivity like tanh functions. I use an Adam optimizer with mean squared error as the loss function. Adam optimizer is an optimization function that can handle sparse gradients on noisy problems. Adam is relatively easy to configure. I have tested many variations of the model including a bidirectional model, altering the units, activation function, loss function and many other optional hyperparameters. It is impossible to test every possible combination of hyperparameters so it is best practise to use your understanding of how each of the main hyperparameters function and then choose the suitable options for your model based on the problem you are trying to solve. It is also good practise to review models built for similar problems and alter the hyperparameters to see if you can have it perform better on your data. The LSTM models do not reproduce the same outputs but do perform within a similar range. Because of this I ran multiple iterations of the same model and then kept the model that have the lowest average RMSE over multiple iterations. The bidirectional model performed the best with a better average RSME of around about 0.1 difference on average across multiple iterations.

The VAR model for pre-covid predictions uses four quarters of lagged values for each model. The VARMAX model uses a lag value of 7 and a moving average window of 1. An Auto-ARIMA model was used to find the optimal values for the lags and the moving average window.

The variables used in the models are unemployment, Consumer Price Index (CPI), export prices and import prices. Other variables such as House Price Index (HPI) and Overnight Interbank Cash Rate (OICR) were tested but did not add to the predictive power of the model so they were dropped. The same variables were used in all the models which makes it easier to make a direct comparison of model performance.

5.3.2 Variance Inflation Factor

In this section I will briefly loop back to the data with respect to the model outputs. In the VAR and VARMAX model most of the variables are not significant. Looking at the variance inflation figures in Table 5.3.1 shows the variables are not highly correlated which would cause the p-values to inflate for the explanatory variables.

Table 5.3.1

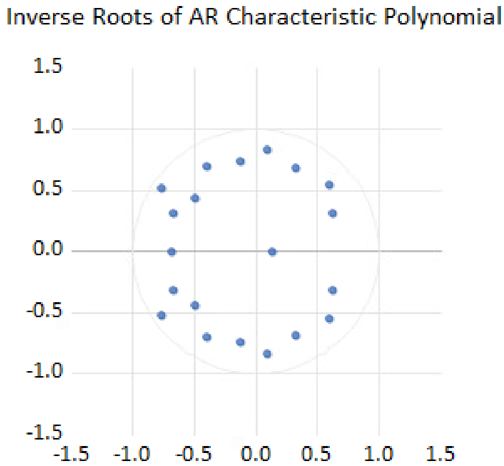
	VIF
Unemployment	3.487
CPI	2.564
Export prices	2.043
Import prices	2.025
QQ_GDP	1.697

5.3.2 AR Roots

I used an augmented Dickey–Fuller test to test for stationarity. All of the variables were tested above and the test showed them to be stationary except for unemployment. The LSTM does not require differencing because they are able to capture non-linear patterns in time series data while considering the inherent characteristics of non-stationary time series data.

Graph 5.3.1 shows an AR root test I conducted on the VAR[3] model's coefficients. The graph shows that the VAR model coefficients are stable all of them being inside the unit circle indicating that the VAR model is stable.

Graph 5.3.1



5.4 Pre-Covid-19 model output

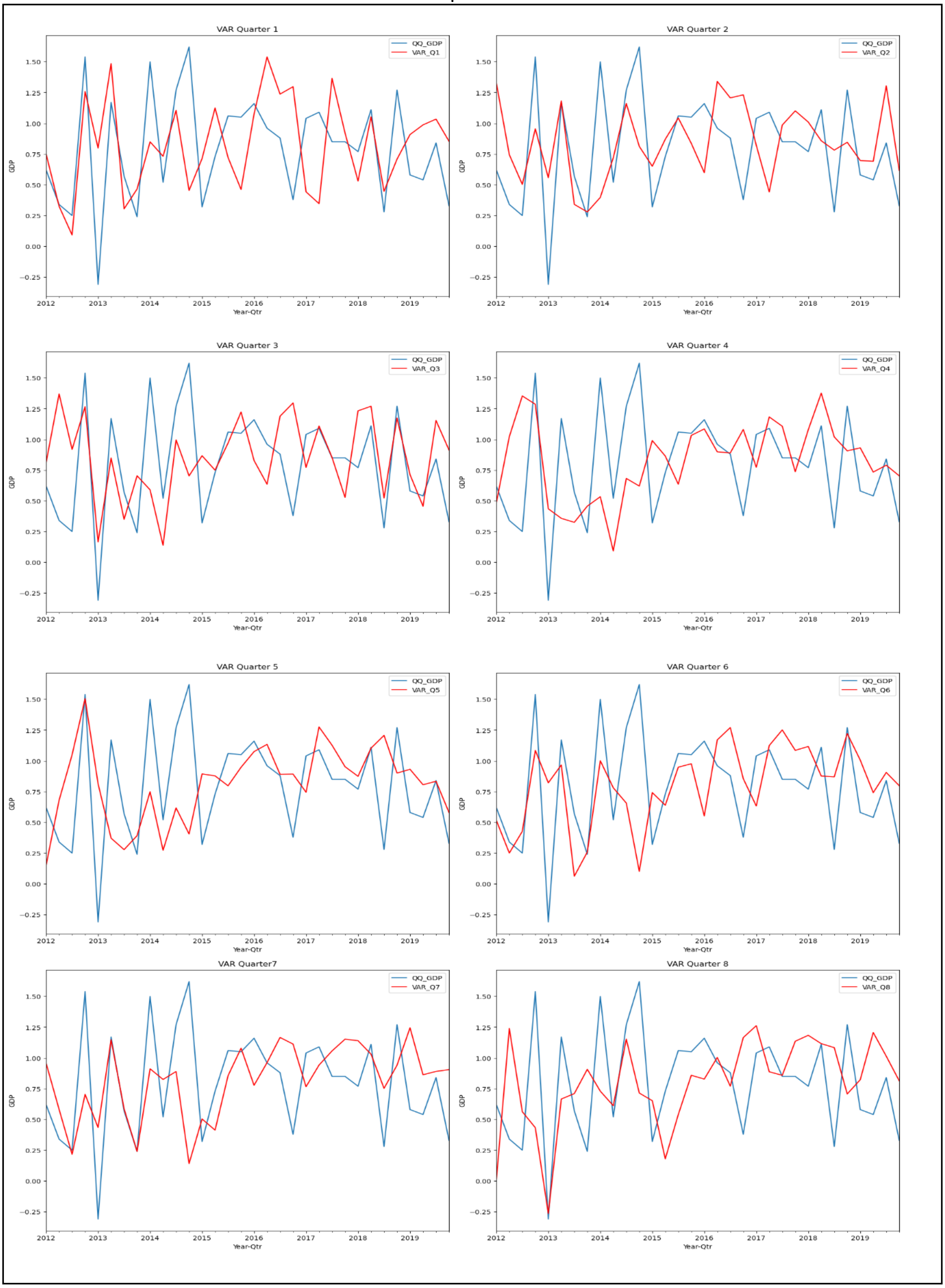
Each model forecasts eight quarters (two years) and loops over the data making a prediction each quarter from March 2012 to December 2019. Because the models forecast eight quarters there are eight predictions for each quarter. Each model is trained from March 1990 to the quarter before the forecast of 8 quarters (2 years) two years out from the final date the model was trained to. The model's predictive power for each quarter is measured by Root Mean Squared Error (RMSE). The results for each quarter forecast for the LSTM model are in Table 5.4.1 in the Appendix. Q1 represents forecasts one quarter ahead and Q2 represents forecasts two quarters ahead and so on in the graphs below.

[3] In the statistical software used to derive the AR roots does not have VARMAX model so only VAR was tested

Graphs 5.4.1

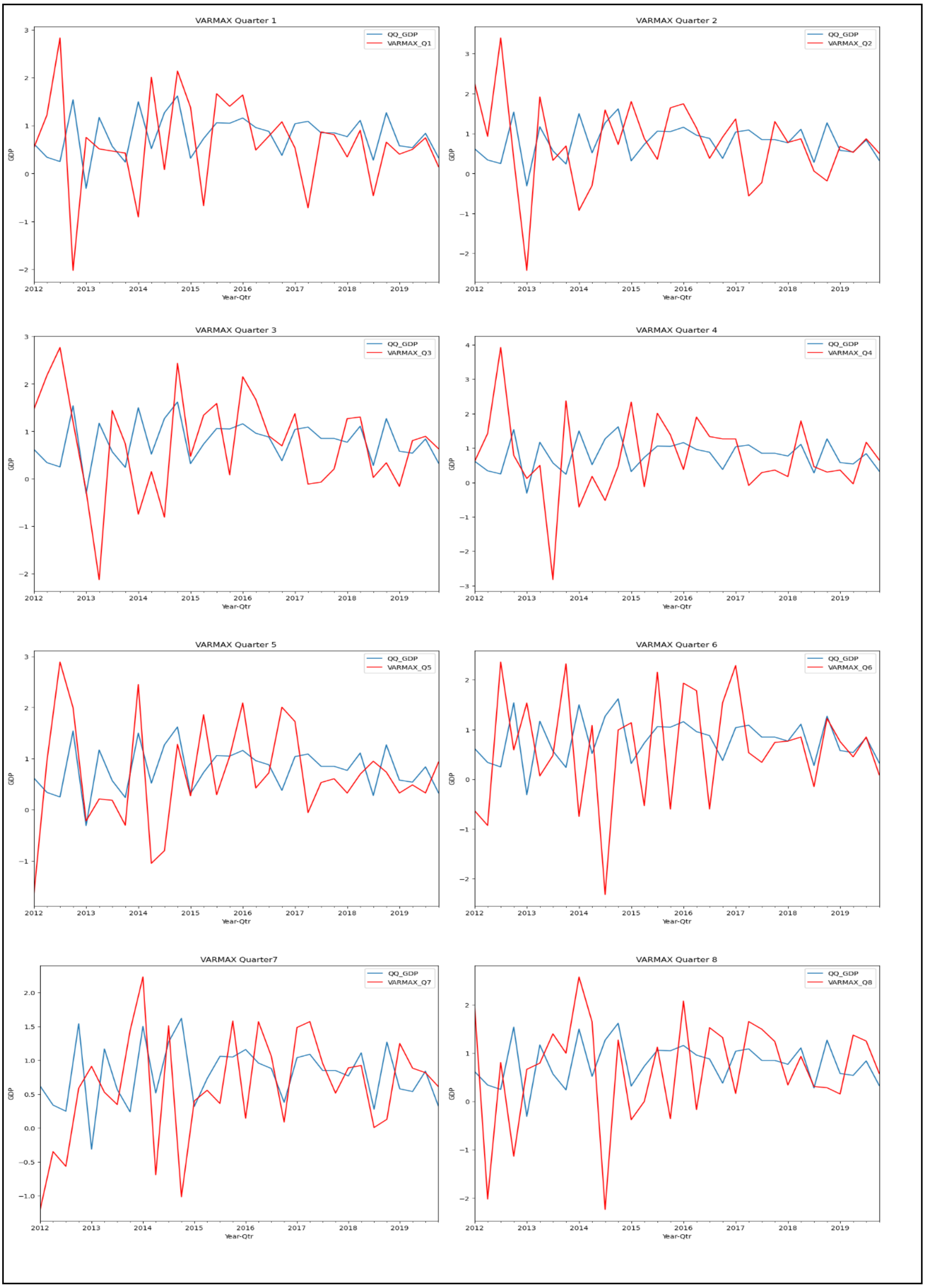
The results show that the LSTM makes relatively stable predictions except at the beginning of the period for the eight quarter forecasts. All model forecasts are provided in Appendix A. Below is a graphical representation of the VAR models performance.

Graphs 5.4.2



The graphs shows that the VAR model makes stable predictions across all 8 time horizons. This is represented in the graphs below.

Graphs 5.4.3

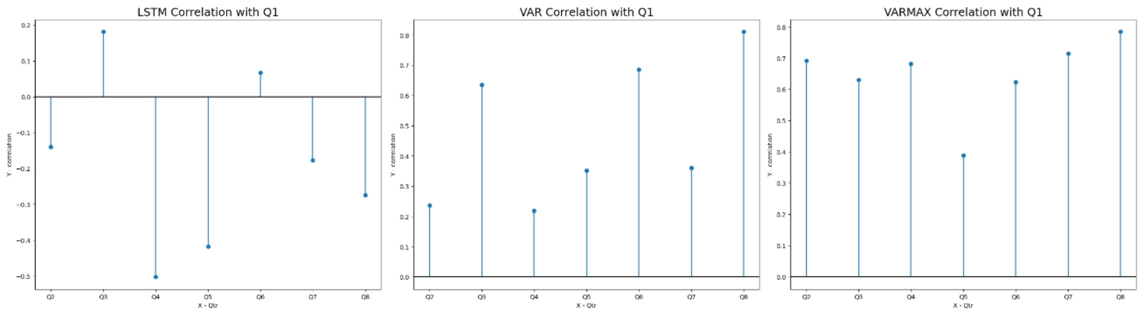

VARMAX Quarter 1
QQ_GDP
VARMAX_Q1
VARMAX Quarter 2
QQ_GDP
VARMAX_Q2
VARMAX Quarter 3
QQ_GDP
VARMAX_Q3
VARMAX Quarter 4
QQ_GDP
VARMAX_Q4
VARMAX Quarter 5
QQ_GDP
VARMAX_Q5
VARMAX Quarter 6
QQ_GDP
VARMAX_Q6
VARMAX Quarter7
QQ_GDP
VARMAX_Q7
VARMAX Quarter 8
QQ_GDP
VARMAX_Q8
GDP
Year-Qtr
2012
2013
2014
2015
2016
2017
2018
2019

The VARMAX model has relatively stable predictions with some variance 2014 quarters. Quarter 6 looks to have the most variance.

5.4.1 LSTM correlation

VAR and VARMAX models multi horizon forecasts are correlated by their construction. LSTM do not suffer from this same draw back due to their design and use of weights rather than coefficients from earlier forecast horizons. This is demonstrated through the graph 5.4.4 below:

Graph 5.4.4



Graph 5.4.4 shows the correlation between forecast horizons for Quarter 1 with the other Quarters. As the graph shows there is no strong correlation between horizons for the LSTM model. This gives the LSTM a potential advantage when forecasting across multistep horizons. This is especially true for economic downturns where there is a sudden decrease in GDP which usually follows from an expansionary period. If your forecasts are correlated with previous forecasts then this will push the model's performance

6 down when you want your model to be able to pick up the downward trend.

5.5 Performance

Looking at the RMSE from table 5.5.1 below the VAR model performs better than the LSTM and VARMAX models overall. For this time series exercise parsimonious models have outperformed the overly complex model, LSTM, which can lead to estimation uncertainty. This can happen for a range of reasons such as the models picking up patterns that are sequential rather than causal. Unfortunately, we cannot look deeper into this with the LSTM because there are too many weights. This is the classic 'black box' issue that gets attributed to machine learning models. The VAR performs well across all time horizons while the LSTM starts strong and gets worse in performance as the time horizon goes further out. The VARMAX models outperforms the LSTM model in the last two time horizons.

Table 5.5.1

RMSE			
	LSTM	VAR	VARMAX
t-1	0.629	0.491	1.128
t-2	0.593	0.458	1.082
t-3	0.689	0.452	1.126
t-4	0.992	0.500	1.301
t-5	0.825	0.491	0.989
t-6	0.990	0.474	1.244
t-7	1.132	0.458	0.842

t-8	1.314	0.506	1.152

I also ran a DM test on each forecast horizon. The DM test compares the VAR models forecasts to the LSTM forecasts and the VARMAX models forecasts to the LSTM forecasts. For the loss differences I subtracted the VAR and VARMAX loss from the LSTM loss so a negative value indicates the LSTM forecast performs better compared to the respective other model and a positive value represents the LSTM forecast performing worse. Table 5.5.2 shows the results.

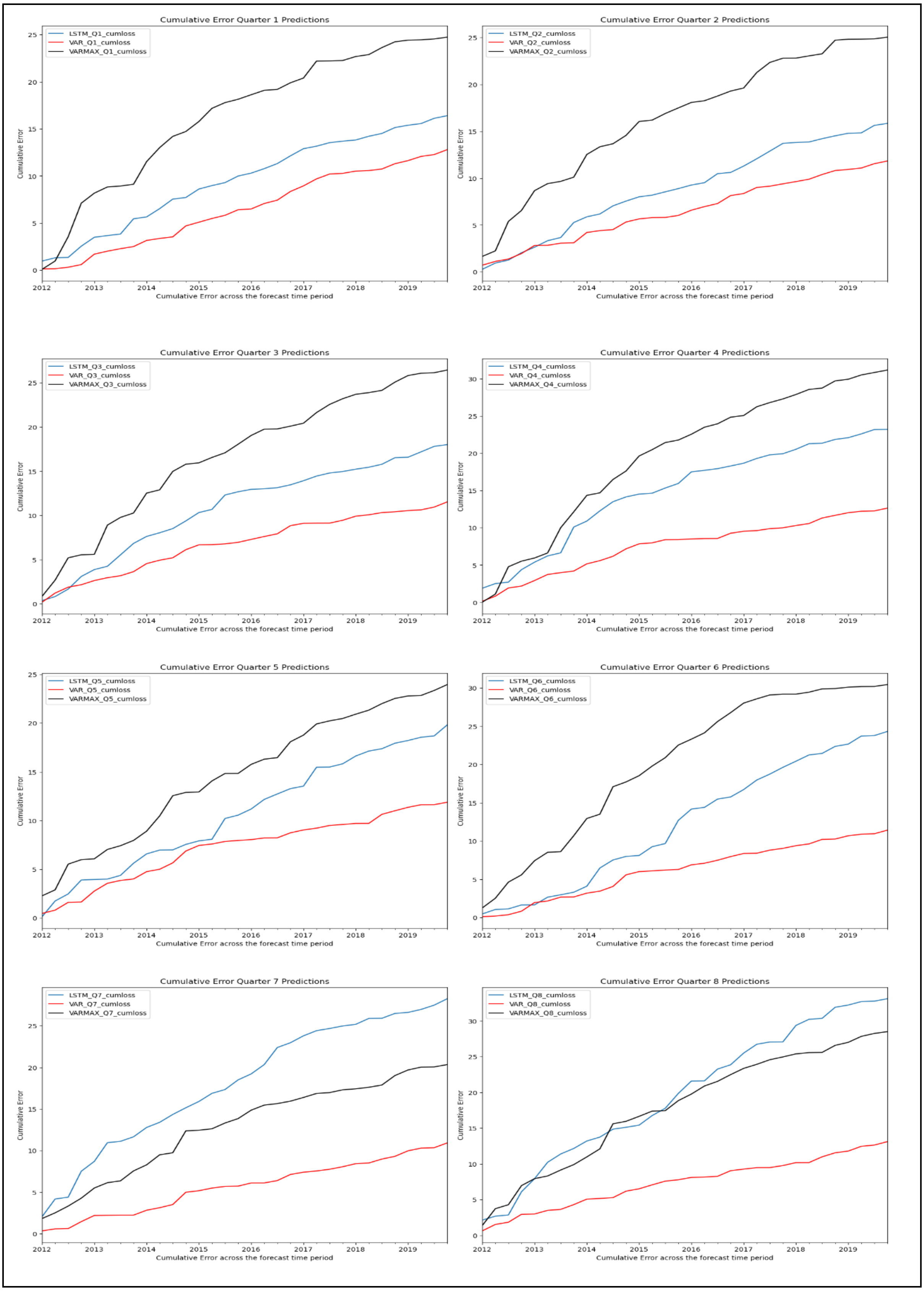
Table 5.5.2

DM Test						
	VAR			VARMAX		
	DM stat	Std Error	P-value	DM stat	Std Error	P-value
t-1	0.155	0.155	0.155	-0.876	0.498	0.078
t-2	0.142	0.081	0.079	-0.820	0.356	0.021
t-3	0.270	0.130	0.038	-0.795	0.395	0.044
t-4	0.733	0.373	0.049	-0.708	0.498	0.155
t-5	0.439	0.201	0.029	-0.795	0.304	0.328
t-6	0.755	0.346	0.029	-0.569	0.471	0.227
t-7	1.070	0.365	0.003	0.572	0.383	0.135
t-8	1.470	0.397	0.000	0.400	0.518	0.441

The DM tests show that VAR performs better than the LSTM for all quarters and is significant for all quarters except quarter 1 and 2. The LSTM model out performs VARMAX in all quarters except quarter 7 and 8, but it is only significant for quarter 2 and 3.

Below in Graph 5.5.1 shows the accumulation of errors for each forecast horizon for each model. The errors were calculated by taking the absolute difference between the forecast and the actual GDP value at each timestep and then cumulatively summed.

Graphs 5.5.1



The VAR models have smallest errors especially in the longer forecast horizons which is demonstrated in the RSME results. What is interesting is the VARMAX and LSTM model's errors are larger in the early forecast timeframe and then the models are more stable in the later forecast periods. This makes sense because the models have more data improving the model's forecasting ability.

Chapter 6: Post Covid-19 Results

In this section I will expand the models to forecast GDP growth over the Covid-19 period from March 2020 to September 2022.This will show how the models perform over an economic downturn. The variables and model hyperparameters are the same as in the Chapter 5.

6.1 Estimators

Again, I have displayed the model summaries for the final looped models. Below in Table 6.1.1 are the estimators for the VAR the GDP model (QQ_GDP). I have also differenced the data for the post Covid-19 models for both VAR and VARMAX consistent to the pre-Covid-19 models.

Table 6.1.1

VAR QQ_GDP Model				
	Estimate	Std Error	t-stat	P-value
const	0.070	0.166	0.422	0.673
CPIL1	-0.104	0.095	-1.096	0.273
CPIL2	0.150	0.261	0.574	0.566
CPIL3	-0.074	0.282	-0.263	0.792
CPIL4	0.443	0.269	1.644	0.100
Export pricesL1	0.051	0.560	0.090	0.928
Export pricesL2	-0.080	0.079	-1.008	0.313
Export pricesL3	-0.031	0.067	-0.461	0.645
Export pricesL4	-0.142	0.096	-1.487	0.137
Import pricesL1	-0.029	0.072	-0.404	0.686
Import pricesL2	-0.960	0.132	-7.284	0.000
Import pricesL3	-0.133	0.261	-0.508	0.611
Import pricesL4	0.025	0.063	0.398	0.691
QQ_GDPL1	0.580	0.585	0.993	0.321
QQ_GDPL2	-0.100	0.078	-1.271	0.204
QQ_GDPL3	-0.068	0.070	-0.961	0.336
QQ_GDPL4	-1.148	0.093	-12.391	0.000
UnemploymentL1	-0.479	0.102	-4.717	0.000
UnemploymentL2	-0.678	0.135	-5.029	0.000
UnemploymentL3	-0.487	0.580	-0.839	0.401
UnemploymentL4	0.623	0.578	1.078	0.281

The Table shows that lagged GDP is the strongest predictor in the model having a negative impact on GDP forecasts for lag 2, 3 and 4 and a positive impact on lag 1. However only GDP lag 4 is significant. Lag 1, 2 and 3 of unemployment had a negative impact on forecasts but only lag 1 and 2 are significant which again is consistent with Okun's law. Lag 4 of unemployment has a positive impact on GDP forecasts but is not significant at a 5% level. Only import prices for lag 2 are significant and has a large negative impact on GDP. This is consistent with economic theory where an increase in the cost of imports e.g. machinery or raw material, increases the cost of production, pushing up prices and decreasing consumption. This leads to a decrease in GDP. None of the lagged coefficients for export prices or CPI are significant.

Below in Table 6.1.2 are the VARMAX estimators.

Table 6.1.2

VARMAX QQ_GDP Model				
	Estimate	Std Error	t-stat	P-value
intercept	0.0391	0.22	0.178	0.859
CPIL1	-0.2771	0.154	-1.801	0.072
CPIL2	0.0805	0.152	0.531	0.596
CPIL3	0.0239	0.172	0.139	0.89
CPIL4	-0.1352	0.126	-1.07	0.285
CPIL5	-0.0043	0.158	-0.027	0.978
CPIL6	-0.2429	0.155	-1.571	0.116
CPIL7	0.0528	0.198	0.266	0.79
e(CPI)L1	0.0718	0.14	0.513	0.608
e(Export prices)L1	0.113	0.187	0.603	0.546
e(Import prices)L1	-0.0676	0.225	-0.3	0.764
e(QQ_GDP)L1	-0.3875	0.125	-3.108	0.002
e(Unemployment)L1	-0.1163	0.08	-1.446	0.148
Export pricesL1	-0.0764	0.166	-0.46	0.645
Export pricesL2	-0.133	0.163	-0.816	0.415
Export pricesL3	-0.1041	0.174	-0.597	0.55
Export pricesL4	-0.0854	0.179	-0.478	0.633
Export pricesL5	-0.031	0.159	-0.195	0.846
Export pricesL6	0.0397	0.125	0.316	0.752
Export pricesL7	-0.0048	0.169	-0.029	0.977
Import pricesL1	-0.0638	0.166	-0.383	0.702
Import pricesL2	-0.0555	0.175	-0.317	0.752
Import pricesL3	-0.061	0.186	-0.327	0.744
Import pricesL4	-0.0647	0.145	-0.447	0.655
Import pricesL5	-0.0655	0.131	-0.5	0.617
Import pricesL6	-0.1572	0.191	-0.821	0.412
Import pricesL7	-0.0382	0.19	-0.201	0.841
QQ_GDPL1	-0.9875	0.142	-6.935	0
QQ_GDPL2	-0.9349	0.151	-6.209	0
QQ_GDPL3	-0.9006	0.167	-5.401	0
QQ_GDPL4	-0.9179	0.254	-3.617	0
QQ_GDPL5	-0.4603	0.211	-2.184	0.029
QQ_GDPL6	-0.2892	0.157	-1.843	0.065
QQ_GDPL7	-0.1317	0.205	-0.644	0.52
UnemploymentL1	-1.4194	0.081	-17.603	0
UnemploymentL2	-0.3254	0.149	-2.19	0.029
UnemploymentL3	0.6963	0.078	8.947	0
UnemploymentL4	0.5773	0.097	5.93	0

UnemploymentL5	-0.7454	0.128	-5.842	0
UnemploymentL6	-0.1018	0.113	-0.905	0.366
UnemploymentL7	1.5239	0.101	15.034	0

Table 6.1.3

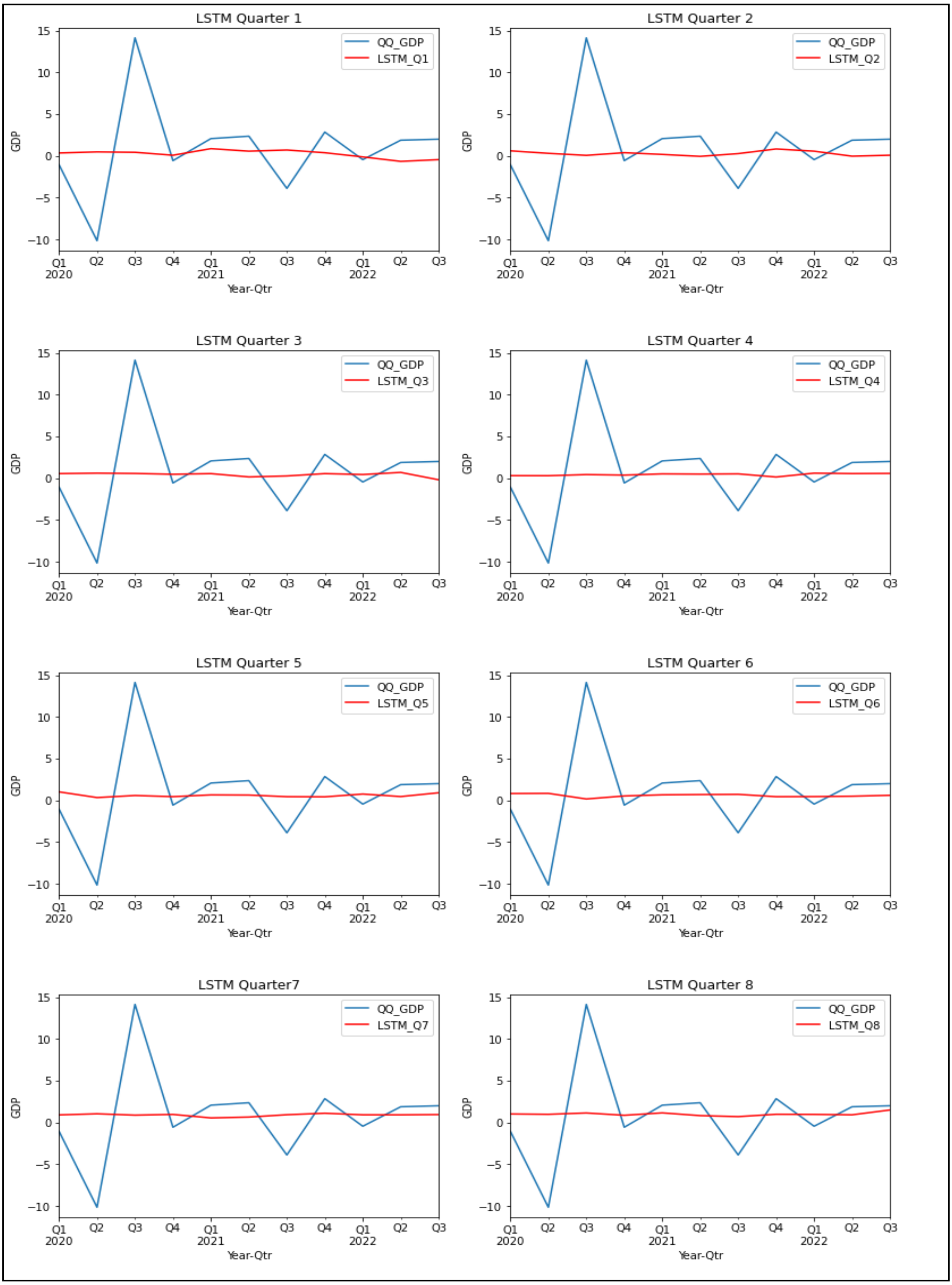
Summary Statistics	VAR	VARMAX
Observations	126	129
Log likelihood	-934.783	-894.274
AIC	2.315	2228.547
BIC	4.679	2857.706
HQIC	3.275	2484.187

Table 6.1.2 shows that lagged GDP is significant in the VARMAX model at a 5% significance level up until lag 5 and have a negative impact on GDP forecasts. Lag 6 and 7 are not significant for GDP and have a negative impact on forecasts. The moving average coefficient is significant and has a positive impact on predictions. All other moving averages for the rest of the variables are not significant at a 5% level. CPI, import prices and export prices are not significant for all lags. Unemployment is significant for all lags except lag 6 and has a negative impact on GDP forecasts for lag 1,2,5,6 with lag 1 having a strong negative impact, consistent with Okun's law. Lag 3, 4 and 7 have positive impacts on GDP with lag 7 being a strong impact. None of the 6 and 7 lagged variables are significant at a 5% level. The LSTM model summary is the same as in Chapter 5 so I have not shared it here. This is because the LSTM hyperparameters are all the same.

6.2 Post Covid-19 model output

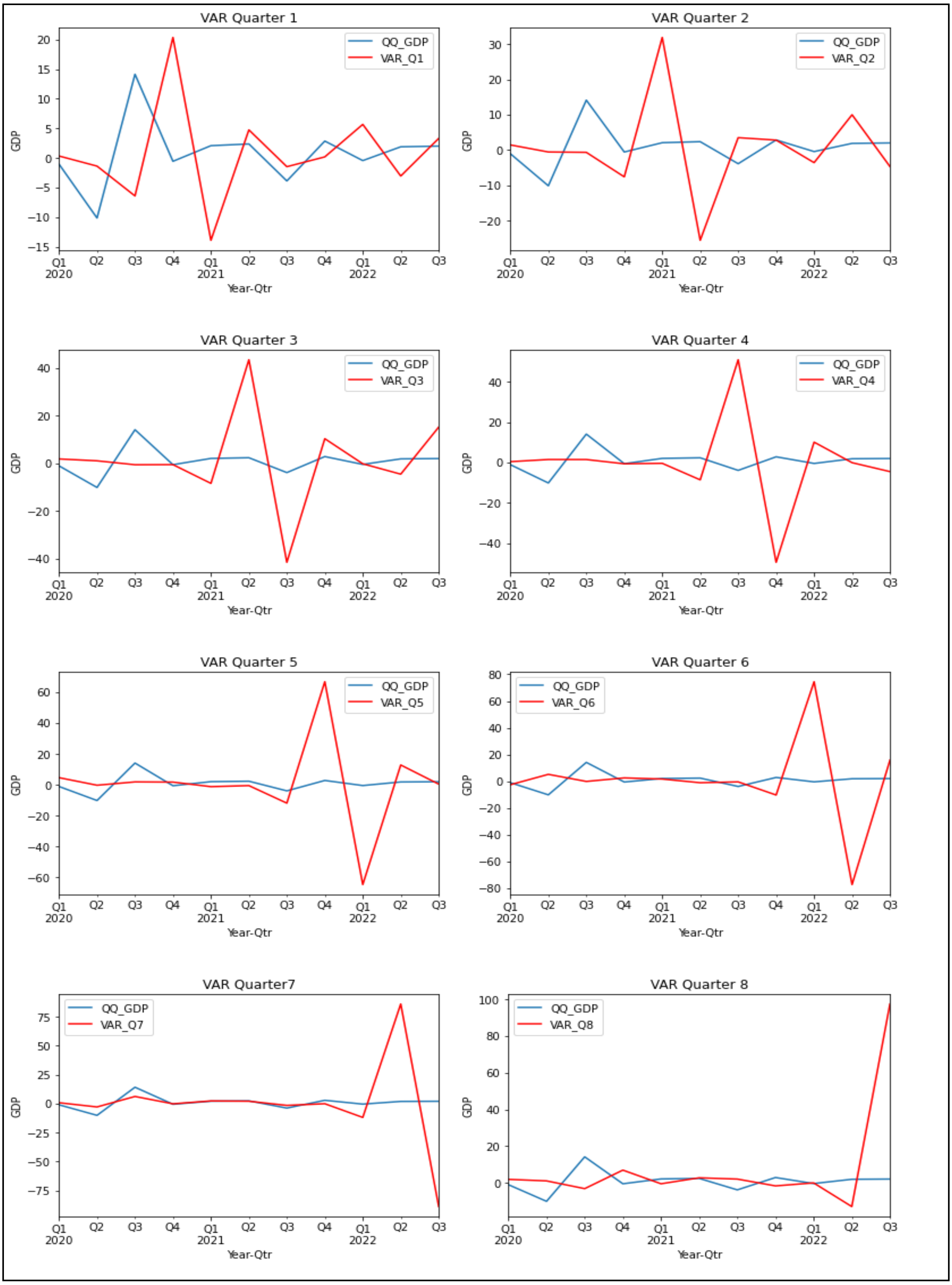
Each model forecasts eight quarters (two years) and loops over the data making a prediction each quarter from March 2020 to September 2022. Because the models forecast eight quarters there are eight predictions for each quarter. Each model is trained from March 1990 to the quarter before the forecast of 8 quarters (2 years) two years out from the final date the model was trained the same as stated in Chapter 5. The model's predictive power for each quarter is measured by Root Mean Squared Error (RMSE). The results for each quarter forecast for the LSTM model are in Table 6.2.1 in Appendix D.

Graphs 6.2.1


LSTM Quarter 1
QQ_GDP
LSTM_Q1
LSTM Quarter 2
QQ_GDP
LSTM_Q2
LSTM Quarter 3
QQ_GDP
LSTM_Q3
LSTM Quarter 4
QQ_GDP
LSTM_Q4
LSTM Quarter 5
QQ_GDP
LSTM_Q5
LSTM Quarter 6
QQ_GDP
LSTM_Q6
LSTM Quarter7
QQ_GDP
LSTM_Q7
LSTM Quarter 8
QQ_GDP
LSTM_Q8
GDP
Year-Qtr
Q1 2020
Q2
Q3
Q4
Q1 2021
Q2
Q3
Q4
Q1 2022
Q2
Q3
15
10
5
0
−5
−10

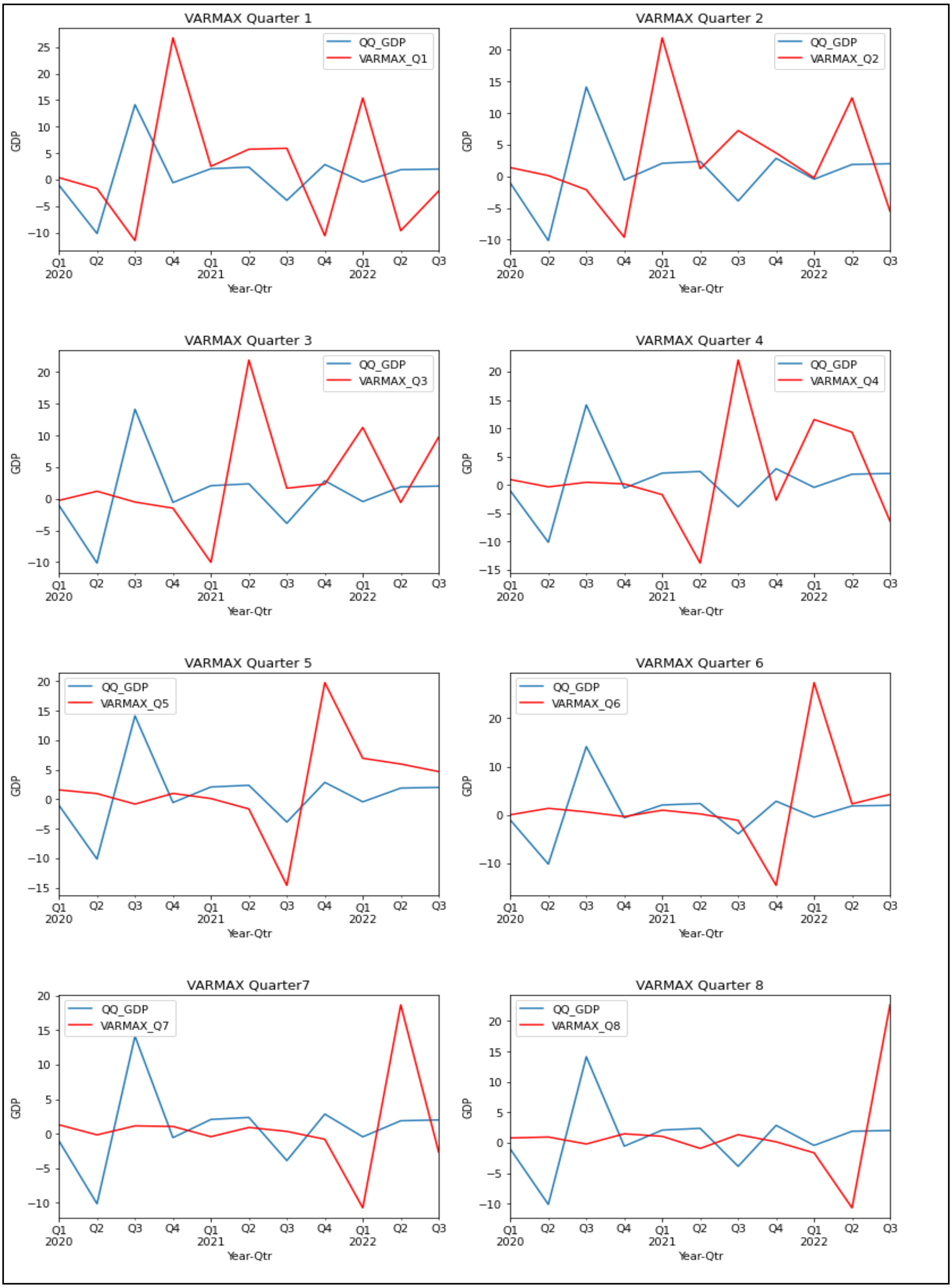
The results show that the LSTM cannot pick up any trends in the data to make forecasts across any quarters. This is not surprising because GDP is very unstable and nothing in the data would be able to connect the effects of Covid-19 lockdowns on GDP.

Graphs 6.2.2



Graph 6.2.2 shows that the VAR model picks up a trend at the beginning but has more trouble identifying trends in the later quarter predictions.

Graphs 6.2.3



Graph 6.2.3 shows that the VARMAX model's also picks up a trend at the beginning but has more trouble identifying trends in the later quarter predictions much like the VAR models.

6.3 Performance

Looking at the RMSE from Table 6.3.1 below the LSTM model performs better than the VAR and VARMAX models overall but in all quarters however the RMSE is so high and we can tell from the graphs above that no model is of any use over the Covid-19 period for forecasting GDP.

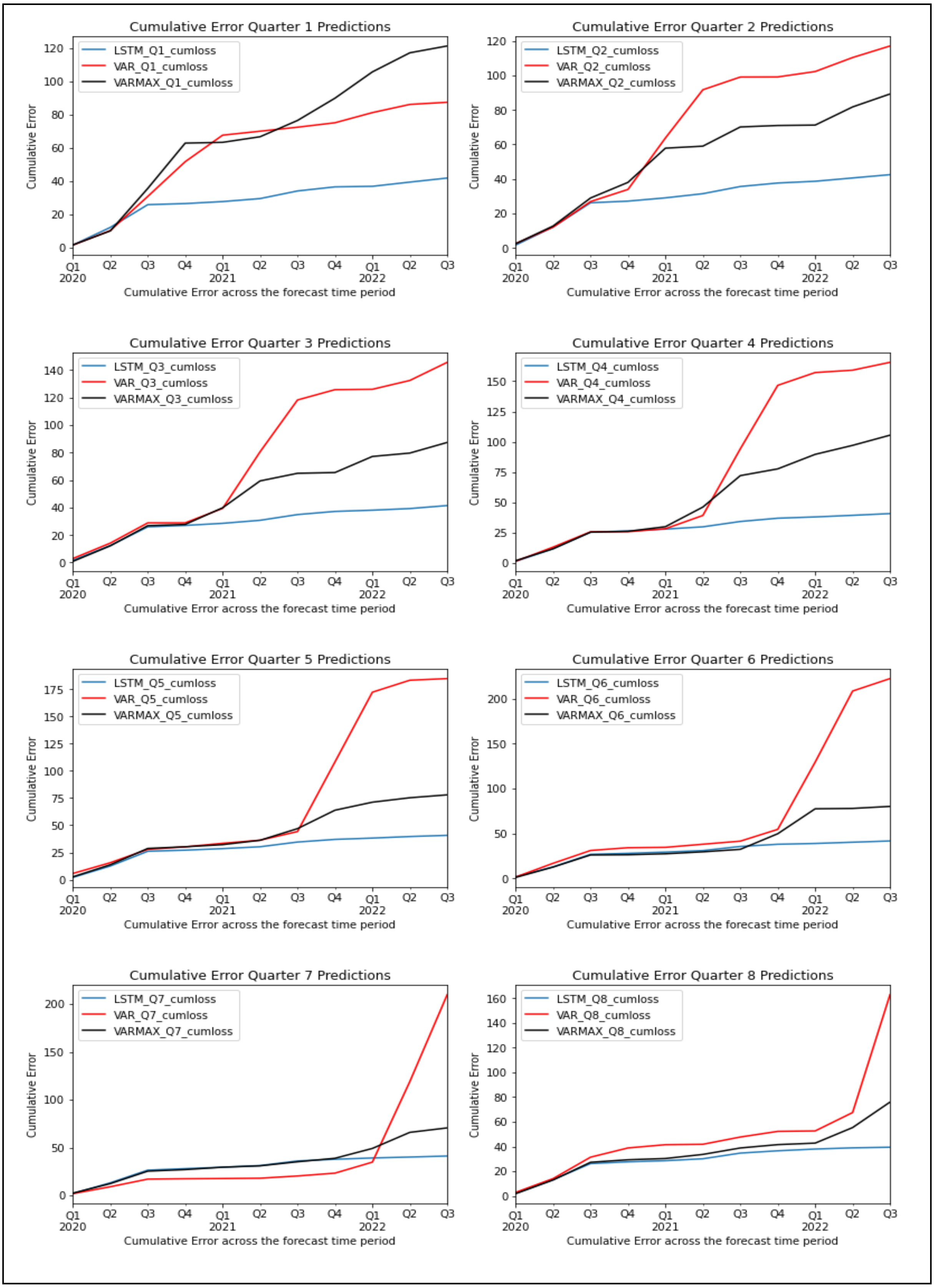
Table 6.3.1

RMSE			
	LSTM	**VAR**	**VARMAX**
t-1	5.621	12.120	14.025
t-2	5.641	17.316	10.199
t-3	5.560	16.618	10.032
t-4	5.536	27.684	11.819
t-5	5.503	26.164	8.819
t-6	5.697	38.680	11.319
t-7	5.579	34.557	8.123
t-8	5.477	35.365	9.388

I did not run a DM test on the post covid models because comparing the forecasts is not going to provide any useful insights since we are dismissing the models based on the performance above.

Below in graphs 6.3.1 shows the accumulation of errors for each forecast horizon for each model. The errors were calculated by taking the absolute difference between the forecast and the actual GDP value at each timestep and then cumulatively summed, the same as discussed in Chapter 5. The VAR models accumulative predictions show that a few bad forecasts throughout their errors. The LSTM shows consistent accumulation of errors because it picked up no trend from the data.

Graphs 6.3.1


Cumulative Error Quarter 1 Predictions
LSTM_Q1_cumloss
VAR_Q1_cumloss
VARMAX_Q1_cumloss
Cumulative Error
Cumulative Error across the forecast time period
Cumulative Error Quarter 2 Predictions
LSTM_Q2_cumloss
VAR_Q2_cumloss
VARMAX_Q2_cumloss
Cumulative Error Quarter 3 Predictions
LSTM_Q3_cumloss
VAR_Q3_cumloss
VARMAX_Q3_cumloss
Cumulative Error Quarter 4 Predictions
LSTM_Q4_cumloss
VAR_Q4_cumloss
VARMAX_Q4_cumloss
Cumulative Error Quarter 5 Predictions
LSTM_Q5_cumloss
VAR_Q5_cumloss
VARMAX_Q5_cumloss
Cumulative Error Quarter 6 Predictions
LSTM_Q6_cumloss
VAR_Q6_cumloss
VARMAX_Q6_cumloss
Cumulative Error Quarter 7 Predictions
LSTM_Q7_cumloss
VAR_Q7_cumloss
VARMAX_Q7_cumloss
Cumulative Error Quarter 8 Predictions
LSTM_Q8_cumloss
VAR_Q8_cumloss
VARMAX_Q8_cumloss
Q1 2020 Q2 Q3 Q4 Q1 2021 Q2 Q3 Q4 Q1 2022 Q2 Q3

Chapter 7: ANZ Comparison:

In this chapter I compare the LSTM model to ANZ forecasts that they publish in their Economic Outlook. I have tried to replicate the models as much as possible from Chapter 5.

7.1 Data

ANZ forecast percent change of real GDP annually. I have retrieved the chain volume GDP annually from the same source as earlier. I have used the same explanatory variables in Chapter 5; CPI, unemployment, import prices and export prices but annual figures instead of quarterly figures. The data only goes back to 1994 for unemployment so the training data is small which will limit the model's ability to train its weights.

ANZ publish their Economic Outlook report quarterly so I have used forecast from their September edition for each year because that is the latest of the year that aligns to my GDP data. ANZ use different data for their models which I am unable to obtain due to intellectual property concerns. However, they have been collecting and refining their data collection and usage for longer than ten years so their data will be a lot richer than what I have collected for this thesis.

7.2 Models

The LSTM model I use is the exact same structure I used in Chapter 5. The prediction period is from 2012 to 2019. All the hyperparameters are the same. The input shape has changed due to the reduction in the dataset size. I use an input shape of 3 which looks back on the last three data points and an output shape of 6 which makes six predictions, 3 for the input periods, and 3 steps into the future which are the 3 years ahead for which I am making the forecasts. This leads to the model weights being different. Model summary below:

Table 7.2.1

```
Model: "sequential_72"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 bidirectional_72 (Bidirectio (None, 3, 240)            120000
_________________________________________________________________
 lstm_145 (LSTM)              (None, 120)               173280
_________________________________________________________________
 dense_72 (Dense)             (None, 6)                 726
=================================================================
Total params: 294,006
Trainable params: 294,006
Non-trainable params: 0
_________________________________________________________________
```

ANZ use a mix of modelling with a judgemental overlay (this information was obtained through correspondence with their economics team).

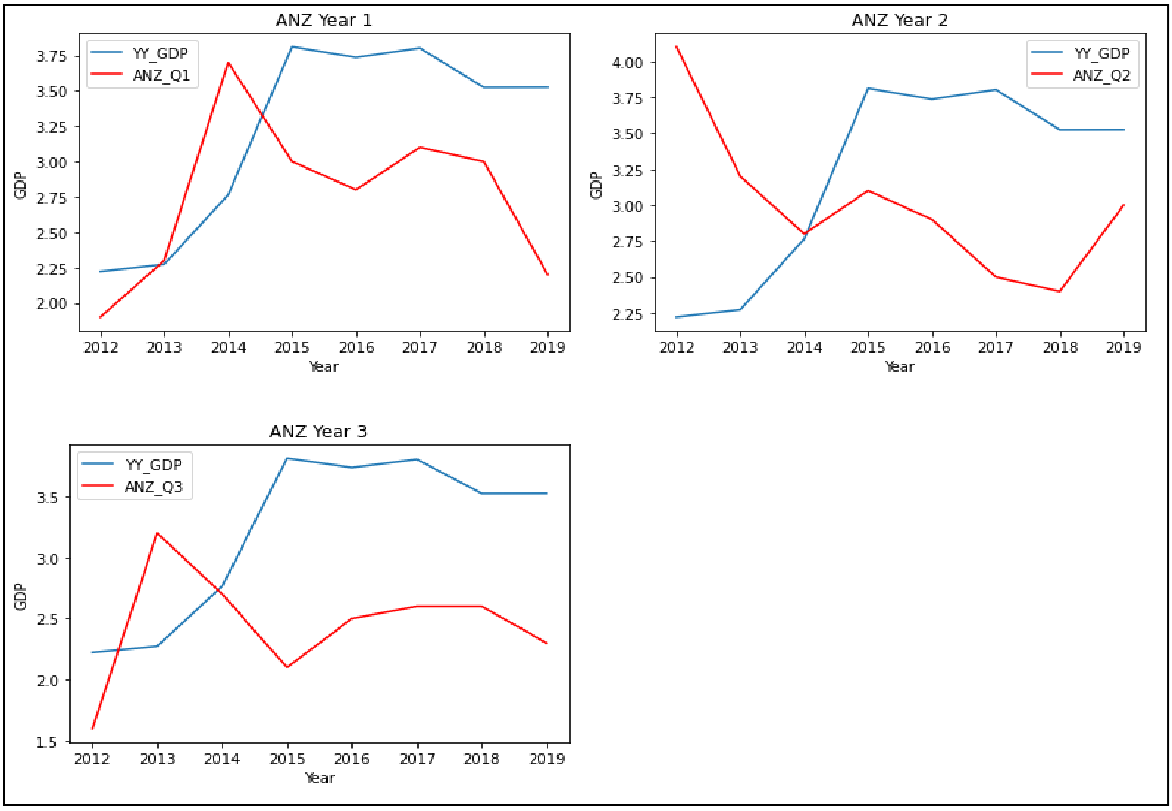
7.3 Model Outputs

The LSTM model for annual forecasting performs well but not as well as in Chapter 5 due to many factors including the dataset being a lot smaller.

Graph 7.3.1

The model had trouble making predictions for the earlier years in year 2 and year 3 model which is understandable given how small the data set was for earlier periods.

Graph 7.3.2



ANZ forecasts are much better one year ahead than two or three years ahead. Again, this is understandable because they use a judgemental overlay the longer the window of prediction, the longer timeframe key economic indicators can change, this makes it hard to apply judgemental overlays on GDP forecasts.

7.4 Performance

Again, I have used the RMSE and the DM test to compare the forecasts.

Table 7.4.1

RMSE		
	LSTM	**ANZ**
t-1	1.001	0.793
t-2	1.085	1.050
t-3	1.129	1.090

The RMSE for ANZ's forecasts are slightly lower for all three prediction periods with their predictions performing best one year ahead.

Table 7.4.2

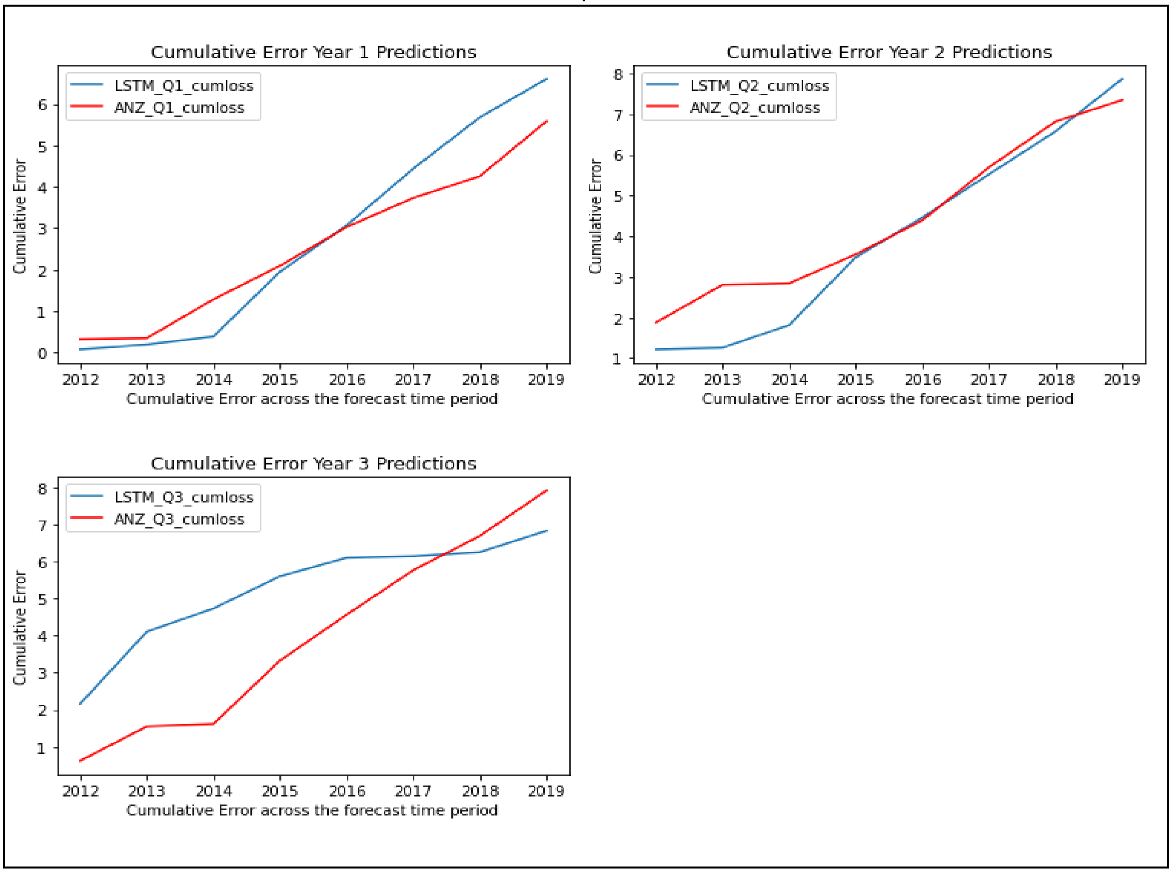
DM Test	
	ANZ

	DM stat	Std Error	P-value
t-1	0.373	0.304	0.219
t-2	0.073	0.480	0.879
t-3	0.086	0.946	0.928

All the DM stats are positive however none of them are significant so we cannot say that ANZ forecasts are performing statistically better than the LSTM model.

Below is the accumulation of errors for each of the year forecasts. They show that over the years the errors are consistent for all forecast periods and for both the LSTM model and ANZ.

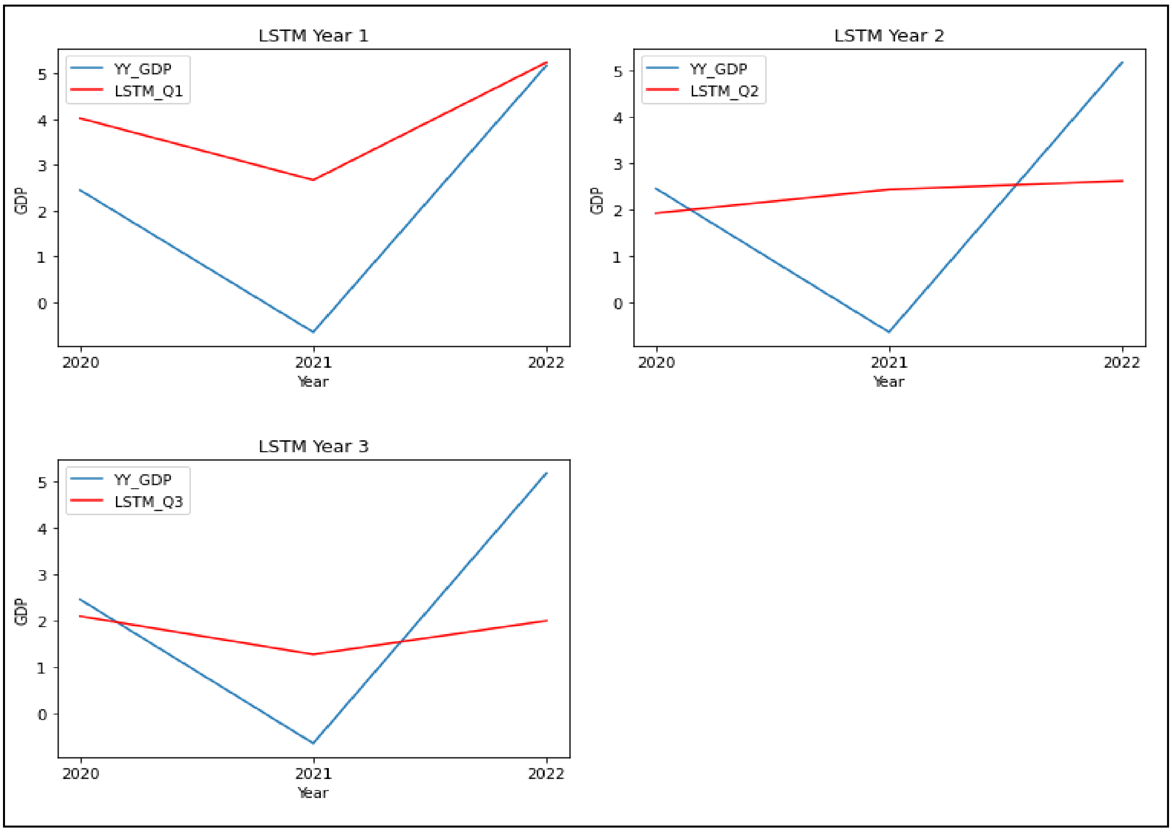
Graph 7.4.1



7.5 Post Covid-19 Comparison

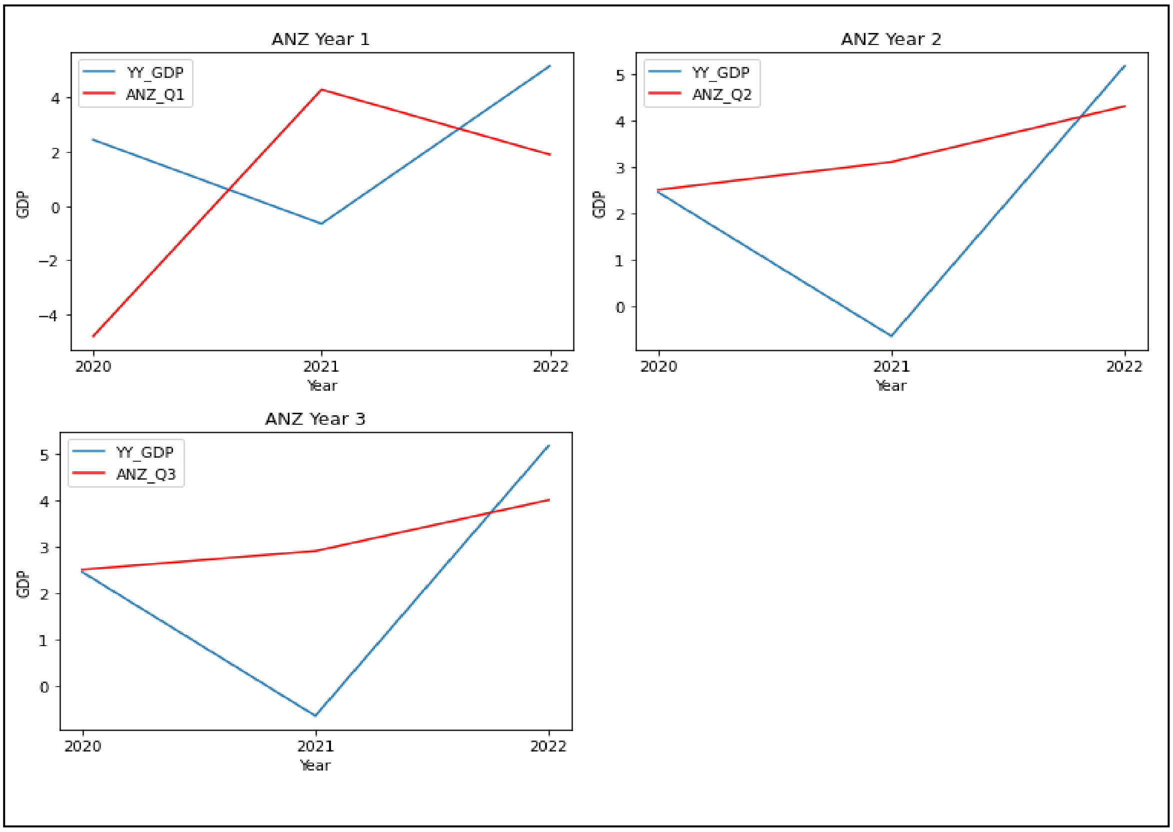
I have also compared the LSTM model against ANZ forecast for the Covid-19 period from 2020 to 2022. The data used for training the models is the same as the models before Covid-19 period and all the performance metrics are the same. The model hyperparameters and input and output shapes are all the same. Below are the LSTM models forecasts.

Graph 7.5.1



Graph 7.5.1 show that the LSTM model picks up the trends in the first-year forecasts but fails to pick up any trend over a longer forecast horizon. The models also don’t predict a negative GDP growth. ANZ forecasts are depicted in Graph 7.5.2 below.

Graph 7.5.2



ANZ forecasts, in Graph 7.5.2, over the first-year horizon move in the opposite direction to the way GDP growth is going. This shows how hard it is to predict GDP movements even with judgemental overlays. The longer horizon forecasts are good for 2020 but fail to pick up trends as the ANZ economists seem to make stable predictions between 2-4% over the long term.

Table 7.5.1

RMSE		
	LSTM	ANZ
t-1	2.122	5.408
t-2	2.332	2.225
t-3	2.154	2.160

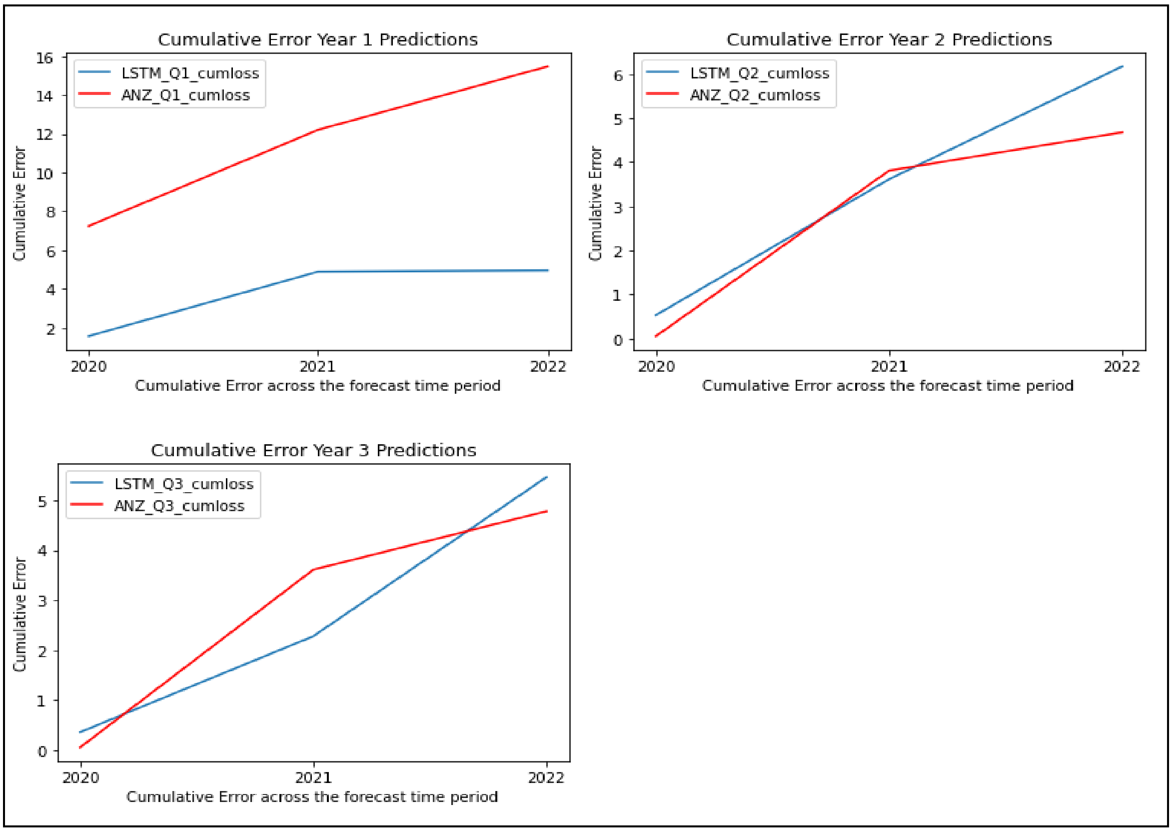
The RMSE metrics show for horizon 1 that the LSTM performs much better than ANZ forecasts and similar to ANZ forecasts for horizon 2 and 3. The LSTM model performs much better on annual figures rather than monthly because annual figures are less volatile than quarterly figures in an economic downturn. Economic downturns happen suddenly which can be illustrated when reviewing stock market index movements at times of critical events.

Table 7.5.2

DM Test			
	ANZ		
	DM stat	Std Error	P-value
t-1	-24.738	9.649	0.010
t-2	0.488	1.768	0.783
t-3	-0.026	2.926	0.993

Table 7.5.2 shows that the LSTM forecast for horizon 1 is better at a 5% level of significance. The other two horizons are not significant so no insights can be drawn. The cumulative errors are shown in Graph 7.5.3.

Graph 7.5.3



Chapter 8: Conclusion

The experiment in this thesis, to compare LSTM models predictive capability to traditional multivariate econometric models, shows that LSTM models can contribute to macroeconomic forecasting precision. However, I have not been able to get the LSTM models to outperform the traditional VAR model with the data I had collated. The results show that further work needs to be done in exploring more data sources or using ensemble models to see if this can boost performance of the LSTM models. Zhang, Wen & Yang (2022) have shown some promising results in their paper, China's GDP forecasting using Long Short Term Memory Recurrent Neural Network and Hidden Markov Model where they use a Long Short Term Memory Recurrent Neural Network and Hidden Markov Model (LSTM-HMM) to predict China's Gross Domestic Product (Zhang, et al, 2022).

Machine Learning is a new field that has come about in the age of big data and improved computer processing power. While researching for this thesis topic I did not come across many papers in academia that applied Machine Learning to forecasting economic indicators. There was some research in this area however the models used in the papers reviewed were not applicable or of practical use for forecasting macroeconomic indicators. This thesis shows the value of Machine Learning when it is applied to economic problems and I believe will soon be an essential approach for any econometrician.

For most Machine Learning problems data is split into training data and test data, usually an 80:20 split. For example, if your data was from 2010 to 2020, you'd train your model on the data from 2010 to 2018 and then use the independent variables from 2019-2020 to test the model's predictive power. The issue with this approach, when forecasting GDP for 2023 (or any other timestep in the future) using a multivariate model, is that you don't have information for the consumption or interest rates (or whatever other independent variables you are using) in 2023. The aim of this thesis has been to compare multivariate models forecasting over multiple horizons. This method can then be applied by commercial or government institutions, such as banks or central banks.

The research in this thesis can be expanded upon (and will be as time goes on) in a number of ways. Firstly, LSTM models have many hyperparameters so these could be altered a number of ways to improve the model's performance. I don't expect much uplift from this approach since every hyperparameter was chosen for logical reasons and I tested a range of hyperparameters. The main area of improvement would be the data used in the models. I used standard macroeconomic indicators that were available from reputable sources but this can be expanded upon by gathering data at a more granular level and building it up by key industries. This maybe industries that are key drivers to the NZ economy such as tourism or dairy farming which may have a strong correlation with GDP growth rates. Finding more data to improve the model's performance would be a natural first step. Another area of interest would be to apply the model to an economic downturn. I tested the model on the current Covid-19 economic downturn but due to its unique nature the results of the forecasts were not useful for any form of insight. The lockdown dramatically reduced output and when lockdown was lifted output spiked again. More research into data and appropriate judgemental overlays is required for this period.

The field of Machine Learning is continuously evolving and new models will continue to be invented and with the further application of the internet of things (IOT) data will be more widely and accurately collected and be made accessible. It is important for econometricians to be across the developments of these fields to see how it can further benefit the field of economics.

References:

Abaidoo, R. (2016). Inflation expectations, economic policy ambiguity and micro-level consumer behavior. *Journal of Financial Economic Policy*, 8(3), 377-395. https://doi-org.ezproxy.aut.ac.nz/10.1108/JFEP-12-2015-0074

ANZ. (2020). *Nz forecast update: Gdp and labour market*. https://www.anz.co.nz/content/dam/anzconz/documents/economics-and-market-research/2020/ANZ-GDP-LabourMarket-Update-20200819.pdf

Arnx, A. (2019). First neural network for beginners explained (with code). *Towards Data Science*. First neural network for beginners explained (with code) | by Arthur Arnx | Towards Data Science

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with graident descent is difficult. *IEEE transactions on neural networks, 5*(2).

Carriero, A., Carriero, A., & Carriero, G. (2019). A comprehensive evaluation of macroeconomic forecasting methods. *International Journal of Forecasting, 35*(4), 1226-1239. https://doi.org/https://doi.org/10.1016/j.ijforecast.2019.02.007

Chauvet, M., & Potter, S. (2013). *Handbook of Economic Forecasting* (Vol. Volume 2A). Elsevier. https://doi.org/https://doi.org/10.1016/B978-0-444-53683-9.00003-7

Colah's Blog, 2015. Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Diebold, F. Mariano, R (1995). Comparing Predictive Accuracy. *Journal of Business and Economic Statistics*, 13, 253-265. pa.dm.pdf (upenn.edu)

Edirisuriya, P. (2015). The predictive power of financial variables: New evidence from australia. *Australasian Accounting, Business and Finance Journal 9*(1), 57-70. https://doi.org/10.14453/aabfj.v9i1.5

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780.

Hopp, D. (2022). *Benchmarking Econometric and Machine Learning Methodologies in Nowcasting* United Nations Conference on Trade and Development,

Hu, M., & Root, H. (1964). An adaptive data processing system for weather forecasting. *Journal of Applied Meteorology and Climatology, 3*(5), 513–523. https://doi.org/https://doi.org/10.1175/1520-0450(1964)003<0513:AADPSF>2.0.CO;2

Kascha, C., & Mertens, K. (2009). Business cycle analysis and VARMA models. *Journal of Economic Dynamics & Control, 33*, 267-282. https://doi.org/https://doi.org/10.1016/j.jedc.2008.05.006

Koec, K. E. (2022). *How Does Back-Propagation Work in Neural Networks?* Towards Data Science. How Does Back-Propagation Work in Neural Networks? | by Kiprono Elijah Koech | Towards Data Science

Kuosmanen, P., & Vataja, J. (2017). The return of financial variables in forecasting GDP growth in the G-7. *Economic Change and Restructuring, 50*, 259-277. https://doi.org/10.1007/s10644-017-9212-7

Lees, K., Matheson, T., & Smith, C. (2011). Open economy forecasting with a DSGE-VAR: Head to head with the RBNZ published forecasts. *International Journal of Forecasting, 27*(3), 512-528. https://doi.org/https://doi.org/10.1016/j.ijforecast.2010.01.008

Lefkowitz, M. (2019). *Professor's perceptron paved the way for AI – 60 years too soon*. https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon

Longo, L., Riccaboni, M., & Rungi, A. (2022). A neural network ensemble approach for GDP forecasting. *Journal of Economic Dynamics & Control, 134*. https://doi.org/https://doi.org/10.1016/j.jedc.2021.104278

Moody, J. (2019). *What does RMSE really mean?*. Towards Data Science. What does RMSE really mean?. Root Mean Square Error (RMSE) is a… | by James Moody | Towards Data Science

Namin, S., & Namin, A. (2018). Forecasting economic and financial time series: Arima vs lstm.

Pradhan, R., & Kumar, A. (2008). Forecasting economic growth using an artificial neural network model. *Journal of Financial Managcnieni and Analysis, 21*(1), 24-31.

RBNZ. (n.d.). *Forecasting*. https://www.rbnz.govt.nz/challenge/team-resources/forecasting

Richardson, A., Mulder, T., & Vehbi, T. (2019). *Nowcasting gdp using machine learning algorithms: A real-time assessment*. https://www.rbnz.govt.nz/-/media/ReserveBank/Files/Publications/Discussion%20papers/2019/dp2019-03.pdf?revision=6505b3a5-634e-45bd-9938-a7af3b4317fb

Rumelhart, D., Hinton, G., & Williams, R. (1985). Learning internal representations by error propagation. *Explorations in the Microstructure of Cognition, 1*.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning representations by back-propagating errors *Nature, 323*, 533–536. https://doi.org/https://doi.org/10.1038/323533a0

Schuster, M., & Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on signal processing, 45*(11).

Stock, J., & Watson, M. (2001). Vector autoregressions. *Journal of Economic Perspectives, 15*, 101-115.

Treasury, T. (2021). *Budget Economic and Fiscal Update 2021*. https://www.treasury.govt.nz/publications/efu/budget-economic-and-fiscal-update-2021-html#section-5

Triacca, U. (n.d.). *Lesson19: Comparing Predictive Accuracy of two Forecasts: The Diebold-Mariano Test*. Lesson19: Comparing Predictive Accuracy of two Forecasts: The Diebold-Mariano Test (sssup.it)

Waldrop, M. (2009). News Feature: What are the limits of deep learning? *Proceedings of the National Academy of Sciences of the United States of America, 116*(4), 1074-1077. https://doi.org/https://doi.org/10.1073/pnas.1821594116

Watson, M. (2000). Macroeconomic forecasting using many predictors. *Econometric Society Monographs, 37*, 87-114.

Zhang, G., Patuwo, B., & Hu, M. (1998). Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting, 14*(1), 35-62. https://doi.org/https://doi.org/10.1016/S0169-2070(97)00044-7

Zhang, J., Wen, J., & Yang, Z. (2022). China's GDP forecasting using Long Short Term Memory Recurrent Neural Network and Hidden Markov Model. *PLoS ONE 17*(6): e0269529. https://doi.org/10.1371/journal.pone.0269529

Appendices:

Appendix A: Pre Covid Model Forecasts

Table 5.4.1 LSTM

Target Forecast Period (t)	NZ GDP growth (t)	1-Quarter-Ahead (Base period: t−1)	2-Quarters-Ahead (Base period: t−2)	3-Quarters-Ahead (Base period: t−3)	4-Quarters-Ahead (Base period: t−4)	5-Quarters-Ahead (Base period: t−5)	6-Quarters-Ahead (Base period: t−6)	7-Quarters-Ahead (Base period: t−7)	8-Quarters-Ahead (Base period: t−8)
2012Q1	0.62	-0.341	0.350	0.253	-1.267	0.754	1.087	2.659	2.740
2012Q2	0.34	-0.004	-0.329	-0.122	0.961	1.956	0.923	2.473	-0.215
2012Q3	0.25	0.292	-0.067	-0.592	0.052	0.973	0.329	0.028	0.084
2012Q4	1.54	0.348	0.756	0.114	-0.133	0.114	1.028	-1.577	-1.707
2013Q1	-0.31	0.635	0.280	0.469	0.696	-0.370	-0.325	-1.498	1.519
2013Q2	1.17	1.352	1.864	1.543	0.327	1.132	0.168	-1.079	-1.131
2013Q3	0.57	0.407	0.908	1.859	0.979	0.949	0.882	0.406	-0.591
2013Q4	0.24	1.860	1.842	1.536	3.708	1.487	0.574	0.783	1.015
2014Q1	1.5	1.301	0.892	0.710	0.700	0.540	0.713	0.374	0.457
2014Q2	0.52	1.400	0.824	0.935	1.892	0.911	2.900	-0.098	1.067
2014Q3	1.27	0.260	0.404	0.800	0.042	1.251	0.224	0.327	0.161
2014Q4	1.62	1.793	2.129	0.753	0.974	1.056	1.154	0.815	1.358
2015Q1	0.32	1.217	0.775	1.267	0.695	-0.041	0.442	1.078	0.018
2015Q2	0.73	1.097	0.551	1.086	0.605	0.562	1.878	1.708	-0.612
2015Q3	1.06	1.376	1.414	-0.571	0.365	-1.071	1.472	0.602	0.046
2015Q4	1.05	1.750	0.707	0.688	1.661	1.393	4.083	-0.112	-1.032
2016Q1	1.16	0.852	0.779	0.885	-0.401	1.789	-0.312	0.440	-0.577
2016Q2	0.96	1.436	0.710	0.895	0.760	1.945	0.737	-0.163	0.982
2016Q3	0.88	1.418	1.843	0.757	0.652	0.322	-0.201	-1.170	-0.770
2016Q4	0.38	1.198	0.511	0.060	0.021	0.924	0.101	0.955	-0.221
2017Q1	1.04	0.278	0.361	0.574	0.669	0.778	0.073	0.207	-0.599
2017Q2	1.09	0.823	0.312	0.566	0.450	-0.847	-0.153	0.471	-0.143

2017Q3	0.85	1.224	0.032	0.501	0.363	0.871	0.073	1.113	1.172
2017Q4	0.85	0.691	0.026	1.017	0.997	0.521	-0.033	1.154	0.863
2018Q1	0.77	0.646	0.666	1.032	0.171	-0.028	-0.031	0.575	-1.536
2018Q2	1.11	0.715	1.156	0.886	0.364	0.598	0.290	0.400	0.254
2018Q3	0.28	0.580	0.630	-0.052	0.213	0.518	0.479	0.298	0.151
2018Q4	1.27	0.644	0.963	0.528	0.761	0.702	0.352	0.685	-0.284
2019Q1	0.58	0.334	0.853	0.637	0.813	0.305	0.877	0.458	0.269
2019Q2	0.54	0.361	0.582	-0.065	1.051	0.875	1.592	0.901	1.025
2019Q3	0.84	0.284	0.039	0.222	1.424	0.977	0.774	0.327	0.903
2019Q4	0.33	0.612	0.120	0.530	0.361	1.460	0.866	1.102	0.666

Note: Reported magnitudes are in percentages.

Table 5.3.2 VAR

Target Forecast Period (t)	NZ GDP growth (t)	1-Quarter-Ahead (Base period: t–1)	2-Quarters-Ahead (Base period: t–2)	3-Quarters-Ahead (Base period: t–3)	4-Quarters-Ahead (Base period: t–4)	5-Quarters-Ahead (Base period: t–5)	6-Quarters-Ahead (Base period: t–6)	7-Quarters-Ahead (Base period: t–7)	8-Quarters-Ahead (Base period: t–8)
2012Q1	0.62	0.751	1.329	0.810	0.484	0.150	0.516	0.957	0.005
2012Q2	0.34	0.326	0.745	1.370	1.022	0.680	0.251	0.582	1.239
2012Q3	0.25	0.092	0.503	0.918	1.354	1.049	0.427	0.216	0.563
2012Q4	1.54	1.258	0.956	1.266	1.286	1.506	1.085	0.704	0.433
2013Q1	-0.31	0.798	0.557	0.165	0.435	0.813	0.821	0.434	-0.265
2013Q2	1.17	1.485	1.181	0.848	0.357	0.371	0.967	1.148	0.666
2013Q3	0.57	0.305	0.341	0.349	0.325	0.278	0.062	0.587	0.710
2013Q4	0.24	0.466	0.279	0.704	0.457	0.392	0.259	0.240	0.908
2014Q1	1.5	0.849	0.398	0.589	0.534	0.749	1.001	0.912	0.730
2014Q2	0.52	0.732	0.720	0.138	0.092	0.273	0.781	0.825	0.612
2014Q3	1.27	1.105	1.161	0.996	0.682	0.617	0.657	0.889	1.153
2014Q4	1.62	0.454	0.812	0.703	0.620	0.405	0.101	0.141	0.715
2015Q1	0.32	0.713	0.651	0.867	0.990	0.893	0.742	0.504	0.652
2015Q2	0.73	1.125	0.870	0.750	0.864	0.878	0.639	0.413	0.180
2015Q3	1.06	0.722	1.043	0.970	0.635	0.797	0.948	0.855	0.539

2015Q4	1.05	0.462	0.837	1.223	1.032	0.948	0.976	1.079	0.859
2016Q1	1.16	1.075	0.599	0.828	1.085	1.075	0.552	0.777	0.828
2016Q2	0.96	1.540	1.340	0.635	0.899	1.134	1.170	0.962	1.004
2016Q3	0.88	1.237	1.206	1.189	0.890	0.890	1.270	1.166	0.771
2016Q4	0.38	1.297	1.231	1.296	1.080	0.892	0.855	1.113	1.165
2017Q1	1.04	0.443	0.821	0.772	0.774	0.745	0.633	0.766	1.262
2017Q2	1.09	0.347	0.442	1.109	1.183	1.275	1.125	0.942	0.887
2017Q3	0.85	1.365	0.985	0.853	1.107	1.127	1.251	1.057	0.859
2017Q4	0.85	0.921	1.101	0.528	0.737	0.952	1.085	1.152	1.135
2018Q1	0.77	0.531	1.010	1.232	1.078	0.874	1.117	1.140	1.184
2018Q2	1.11	1.053	0.858	1.270	1.376	1.103	0.876	1.030	1.116
2018Q3	0.28	0.447	0.782	0.524	1.020	1.207	0.870	0.753	1.084
2018Q4	1.27	0.708	0.846	1.174	0.907	0.900	1.222	0.941	0.707
2019Q1	0.58	0.908	0.697	0.713	0.931	0.931	1.006	1.245	0.823
2019Q2	0.54	0.987	0.691	0.456	0.734	0.806	0.742	0.863	1.206
2019Q3	0.84	1.034	1.304	1.154	0.790	0.834	0.905	0.890	1.014
2019Q4	0.33	0.854	0.617	0.913	0.703	0.580	0.798	0.905	0.814

Note: Reported magnitudes are in percentages.

Table 5.3.3 VARMAX

Target Forecast Period (t)	NZ GDP growth (t)	1-Quarter-Ahead (Base period: t−1)	2-Quarters-Ahead (Base period: t−2)	3-Quarters-Ahead (Base period: t−3)	4-Quarters-Ahead (Base period: t−4)	5-Quarters-Ahead (Base period: t−5)	6-Quarters-Ahead (Base period: t−6)	7-Quarters-Ahead (Base period: t−7)	8-Quarters-Ahead (Base period: t−8)
2012Q1	0.62	0.538	2.267	1.465	0.609	-1.657	-0.630	-1.200	1.981
2012Q2	0.34	1.222	0.930	2.181	1.427	0.955	-0.926	-0.346	-2.021
2012Q3	0.25	2.830	3.394	2.769	3.924	2.888	2.359	-0.564	0.807
2012Q4	1.54	-2.023	0.372	1.196	0.790	1.996	0.592	0.589	-1.133
2013Q1	-0.31	0.755	-2.425	-0.256	0.118	-0.219	1.537	0.912	0.665
2013Q2	1.17	0.514	1.918	-2.129	0.497	0.211	0.071	0.530	0.795
2013Q3	0.57	0.470	0.329	1.441	-2.822	0.186	0.481	0.348	1.401
2013Q4	0.24	0.427	0.689	0.740	2.372	-0.302	2.323	1.436	0.999

2014Q1	1.5	-0.906	-0.923	-0.749	-0.712	2.451	-0.747	2.230	2.577
2014Q2	0.52	2.010	-0.300	0.149	0.178	-1.046	1.084	-0.687	1.650
2014Q3	1.27	0.083	1.592	-0.810	-0.521	-0.801	-2.316	1.513	-2.235
2014Q4	1.62	2.140	0.727	2.436	0.481	1.280	0.993	-1.014	1.272
2015Q1	0.32	1.381	1.802	0.474	2.336	0.273	1.139	0.400	-0.377
2015Q2	0.73	-0.671	0.878	1.340	-0.122	1.860	-0.528	0.559	0.000
2015Q3	1.06	1.668	0.357	1.587	2.013	0.295	2.154	0.363	1.124
2015Q4	1.05	1.404	1.646	0.083	1.387	1.045	-0.596	1.581	-0.356
2016Q1	1.16	1.641	1.747	2.153	0.378	2.090	1.935	0.147	2.079
2016Q2	0.96	0.491	1.142	1.673	1.902	0.426	1.783	1.570	-0.167
2016Q3	0.88	0.781	0.380	0.903	1.336	0.723	-0.595	1.061	1.528
2016Q4	0.38	1.080	0.911	0.694	1.269	2.006	1.542	0.088	1.323
2017Q1	1.04	0.535	1.367	1.375	1.268	1.726	2.290	1.486	0.164
2017Q2	1.09	-0.716	-0.560	-0.113	-0.084	-0.055	0.536	1.572	1.654
2017Q3	0.85	0.869	-0.226	-0.071	0.289	0.533	0.342	0.954	1.497
2017Q4	0.85	0.813	1.302	0.207	0.362	0.607	0.743	0.516	1.242
2018Q1	0.77	0.346	0.779	1.268	0.173	0.328	0.770	0.889	0.343
2018Q2	1.11	0.904	0.870	1.303	1.792	0.697	0.852	0.924	0.933
2018Q3	0.28	-0.463	0.061	0.027	0.460	0.950	-0.146	0.009	0.310
2018Q4	1.27	0.656	-0.187	0.337	0.303	0.736	1.225	0.130	0.285
2019Q1	0.58	0.405	0.681	-0.162	0.362	0.328	0.761	1.250	0.155
2019Q2	0.54	0.504	0.529	0.805	-0.038	0.486	0.452	0.885	1.374
2019Q3	0.84	0.746	0.871	0.895	1.171	0.329	0.853	0.819	1.251
2019Q4	0.33	0.142	0.508	0.633	0.657	0.933	0.091	0.615	0.581

Note: Reported magnitudes are in percentages.

Appendix B: LSTM Weights

You can take a look into the layer's weights. For simplicity (if you can use that word when trying to understand LSTMs) I will look into the first layer which has 60,000 weights and bias. The first layer has three tensors: 'lstm_154/lstm_cell_154/kernel:0' shape=(4, 480), 'lstm_154/lstm_cell_154/recurrent_kernel:0' shape=(120, 480), 'lstm_154/lstm_cell_154/bias:0' shape=(480,). That is a kernel, recurrent kernel and bias. The shape gives the number of parameters (weights and bias): (4*480)+(120*480)+480 = 60,000. The 480 is one of the dimensions of each tensor which is 120 * 4. The 120 is the number of units and 4 units of the LSTM cell: input, forget, cell state and output (refer back to the methodology for a refresher of these terms. Below in table 4.5.4 of the weights from one of the dimensions from the forget gate of one of the tensors that transforms the inputs to initial values for the next layer. The forget gate decides, based on the previous hidden state (h_{t-1}) and the input (x), which values to forget from the previous internal state of the cell (C_{t-1}) : f_t = sigmoid(W_f * x + U_f * h_{t-1} + b_f). f_t is a vector of values between 0 and 1 that will encode what to keep (=1) and what to forget (=0) from the previous cell state. Table 4.5.4 is one dimension of W_f. For all practical purposes I cannot say how the variables in the model are interacting to influence the predictions of GDP. This is why LSTMs and Neural Networks are known as black boxes.

Table 4.5.4

1-side of tensor in Forget gate					
-0.104	-0.104	-0.036	0.152	-0.088	-0.124
-0.119	0.066	0.043	-0.101	0.085	-0.043
0.007	0.033	-0.065	-0.148	0.056	-0.020
0.008	0.061	0.126	-0.009	-0.106	-0.040
-0.032	-0.027	0.037	0.092	0.021	0.040
-0.100	-0.062	-0.024	0.083	0.012	-0.097
-0.106	0.077	0.130	0.083	0.078	-0.030
0.002	0.137	-0.013	-0.088	-0.036	-0.026
0.076	0.162	0.063	0.096	0.047	0.081
0.035	0.024	-0.037	-0.100	-0.065	0.069
-0.092	-0.084	-0.037	0.051	0.076	0.069
0.057	0.017	0.025	0.045	0.068	-0.117
0.050	0.040	-0.110	0.000	-0.145	0.116
-0.042	0.109	0.124	0.100	0.137	0.034
0.097	-0.010	-0.004	-0.016	-0.058	-0.034
-0.027	0.041	-0.035	-0.096	-0.006	0.087
-0.044	-0.113	0.047	-0.028	-0.065	-0.087
-0.119	-0.106	-0.067	0.080	0.161	0.093
0.052	-0.074	0.148	0.159	-0.031	0.139
0.076	0.000	-0.016	-0.064	-0.048	0.171

Appendix C: GDP Data Source & Model Code

Chain volume, Seasonally adjusted, Total data from Statistics New Zealand (Table reference SNE070AA)

LSTM Python code:

```
model = Sequential()

model.add(keras.layers.Bidirectional(LSTM(120, activation='relu',return_sequences=True,
input_shape=(n_steps_in, n_features))))

model.add(LSTM(120, activation='relu'))

model.add(Dense(n_steps_out))

model.compile(optimizer='adam', loss='mse')
```

Appendix D: Post Covid Model Forecasts

Table 6.2.1 LSTM

Target Forecast Period (t)	NZ GDP growth (t)	1-Quarter-Ahead (Base period: t−1)	2-Quarters-Ahead (Base period: t−2)	3-Quarters-Ahead (Base period: t−3)	4-Quarters-Ahead (Base period: t−4)	5-Quarters-Ahead (Base period: t−5)	6-Quarters-Ahead (Base period: t−6)	7-Quarters-Ahead (Base period: t−7)	8-Quarters-Ahead (Base period: t−8)
2020Q1	-1.05	0.355	0.613	0.562	0.317	1.026	0.828	0.918	1.028
2020Q2	-10.14	0.490	0.315	0.619	0.309	0.342	0.847	1.049	0.988
2020Q3	14.14	0.439	0.073	0.580	0.444	0.590	0.174	0.889	1.146
2020Q4	-0.56	0.087	0.391	0.482	0.382	0.452	0.534	0.973	0.872
2021Q1	2.08	0.875	0.184	0.559	0.529	0.663	0.671	0.557	1.158
2021Q2	2.37	0.571	-0.049	0.164	0.504	0.644	0.704	0.658	0.830
2021Q3	-3.88	0.714	0.282	0.280	0.530	0.448	0.728	0.938	0.712
2021Q4	2.86	0.386	0.837	0.560	0.153	0.430	0.447	1.116	0.992
2022Q1	-0.44	-0.114	0.572	0.452	0.618	0.766	0.459	0.926	0.979
2022Q2	1.89	-0.648	-0.027	0.704	0.565	0.463	0.504	0.924	0.926
2022Q3	2.01	-0.438	0.090	-0.180	0.585	0.919	0.613	0.956	1.506

Note: Reported magnitudes are in percentages.

Table 6.2.2 VAR

Target Forecast	NZ GDP growth (t)	1-Quarter-Ahead (Base period: t−1)	2-Quarters-Ahead (Base period: t−2)	3-Quarters-Ahead (Base period: t−3)	4-Quarters-Ahead (Base period: t−4)	5-Quarters-Ahead (Base period: t−5)	6-Quarters-Ahead (Base period: t−6)	7-Quarters-Ahead (Base period: t−7)	8-Quarters-Ahead (Base period: t−8)

Period (t)									
2020Q1	-1.05	0.070	1.498	1.174	-0.239	4.377	-2.443	0.957	1.552
2020Q2	-10.14	-1.412	-0.288	1.820	1.306	-0.627	4.963	-2.891	0.996
2020Q3	14.14	-15.005	-1.877	-0.493	2.204	1.629	-0.659	5.800	-3.222
2020Q4	-0.56	8.047	-19.607	-1.925	-0.648	2.438	1.801	-0.945	6.463
2021Q1	2.08	-20.019	37.292	-22.245	-2.080	-0.968	2.609	1.837	-1.186
2021Q2	2.37	-5.168	-32.260	22.930	-24.196	-2.244	-1.073	2.967	2.101
2021Q3	-3.88	-2.731	0.275	-42.277	60.117	-27.754	-2.477	-1.382	3.146
2021Q4	2.86	4.658	5.404	7.193	-54.442	38.286	-30.804	-2.644	-1.605
2022Q1	-0.44	5.440	-1.171	-0.432	6.596	-73.785	84.300	-34.503	-2.713
2022Q2	1.89	-3.546	9.475	-1.132	1.563	4.991	-85.649	51.594	-37.475
2022Q3	2.01	3.049	-5.372	14.544	-0.057	1.938	5.364	-94.642	109.905

Table 6.2.3 VARMAX

Target Forecast Period (t)	NZ GDP growth (t)	1-Quarter-Ahead (Base period: t−1)	2-Quarters-Ahead (Base period: t−2)	3-Quarters-Ahead (Base period: t−3)	4-Quarters-Ahead (Base period: t−4)	5-Quarters-Ahead (Base period: t−5)	6-Quarters-Ahead (Base period: t−6)	7-Quarters-Ahead (Base period: t−7)	8-Quarters-Ahead (Base period: t−8)
2020Q1	-1.05	0.385	1.394	-0.245	0.938	1.585	0.060	1.299	0.784
2020Q2	-10.14	-1.667	0.131	1.199	-0.355	0.956	1.385	-0.155	0.942
2020Q3	14.14	-11.469	-2.104	-0.507	0.465	-0.821	0.651	1.150	-0.205
2020Q4	-0.56	26.746	-9.615	-1.473	0.173	0.988	-0.317	1.064	1.471
2021Q1	2.08	2.532	21.914	-10.028	-1.732	0.119	1.002	-0.417	1.053
2021Q2	2.37	5.744	1.217	21.903	-13.815	-1.641	0.224	0.919	-0.934
2021Q3	-3.88	5.924	7.258	1.678	22.055	-14.598	-1.101	0.346	1.331

2021Q4	2.86	-10.561	3.708	2.321	-2.712	19.775	-14.522	-0.791	0.152
2022Q1	-0.44	15.411	-0.213	11.266	11.540	6.937	27.353	-10.729	-1.659
2022Q2	1.89	-9.609	12.422	-0.576	9.301	5.970	2.303	18.683	-10.704
2022Q3	2.01	-2.124	-5.511	9.756	-6.405	4.687	4.260	-2.652	22.669

Note: Reported magnitudes are in percentages.

Appendix E: Annual Forecasts

LSTM Pre-covid

Target Forecast Period (t)	NZ GDP growth (t)	1- Year -Ahead (Base period: t–1)	2- Year -Ahead (Base period: t–2)	3- Year -Ahead (Base period: t–3)
2012	2.22	3.723	3.187	3.256
2013	2.27	2.015	3.873	2.759
2014	2.76	2.224	-0.228	3.184
2015	3.81	1.862	1.067	1.442
2016	3.74	3.646	1.219	1.920
2017	3.80	1.950	2.212	1.224
2018	3.52	3.826	2.683	1.591
2019	3.53	15.368	4.153	2.687

Note: Reported magnitudes are in percentages.

Post Covid

Target Forecast Period (t)	NZ GDP growth (t)	1-Year-Ahead (Base period: t–1)	2- Year -Ahead (Base period: t–2)	3- Year -Ahead (Base period: t–3)
2020	2.45	4.017	1.914	2.087
2021	-0.65	2.669	2.426	1.264
2022	5.17	5.237	2.610	1.988

Note: Reported magnitudes are in percentages.