BI

WISEflow
Europe/Oslo(CEST)
30 Jun 2022

Handelshøyskolen BI

GRA 19703 Master Thesis

Thesis Master of Science 100% - W

Predefinert informasjon

Startdato:	16-01-2022 09:00	**Termin:**	202210
Sluttdato:	01-07-2022 12:00	**Vurderingsform:**	Norsk 6-trinns skala (A-F)
Eksamensform:	T		
Flowkode:	202210||10936||IN00||W||T		
Intern sensor:	(Anonymisert)		

Deltaker

Simen Bentsen, Akhmed Telkhigov

Informasjon fra deltaker

Tittel *:	Forecasting Norwegian Mainland Gross Domestic Product - Using Deep Learning Algorithms
Navn på veileder *:	Emil Stoltenberg

Inneholder besvarelsen konfidensielt materiale?:	Nei	**Kan besvarelsen offentliggjøres?:**	Ja

Gruppe

Gruppenavn:	(Anonymisert)
Gruppenummer:	281
Andre medlemmer i gruppen:	

BI Norwegian Business School
Oslo, Spring 2022

Forecasting Norwegian Mainland Gross Domestic Product

Using Deep Learning Algorithms

Simen Bentsen & Akhmed Telkhigov

Supervisor: Emil Stoltenberg

GRA 1974 - Master Thesis

Business Analytics

BI NORWEGIAN BUSINESS SCHOOL

Acknowledgements

This thesis is the last chapter for both of our Master of Science degrees in Business Analytics at BI Norwegian Business School campus Oslo. It was written during the spring of 2022. Working on this thesis has been challenging, yet rewarding. We are both interested in macroeconomics and utilization of new technologies. Thus, working on this thesis have been a very enjoyable experience and we have gained a lot of valuable insights within the field of deep learning. We want to thank our supervisor Emil Stoltenberg for his passion and drive to help us throughout the whole period working on this project. He has been heavily involved all the way from the proposal to the finish product, which has resulted in what we believe is a very interesting thesis. Additionally, we would like to thank Vajira Thambawita from Simula for his excellent advice and contribution on especially the technical aspects which has lifted the quality of the thesis.

BI Norwegian Business School

Oslo, July 1, 2022

Simen Bentsen

Akhmed Telkhigov

Abstract

Gross domestic product is a measure of overall economic activity. It is therefore regarded as one of the most important summary factors for understanding the economic state of a country. Hence, an accurate prediction of the gross domestic product can lead to great advantages for individuals, businesses and institutions in financial decision-making. In this thesis, we forecast the growth of the Norwegian mainland economy using deep learning algorithms, which consists of: convolutions neural networks, recurrent neural networks, long short-term memory and encoder-decoder architectures. Specifically, our models utilizes quarterly-, monthly- and daily, macroeconomic- and financial data to predict the quarterly volume change in the Norwegian mainland gross domestic product. To evaluate the performance of our best deep learning model, we compare our predictions to leading forecasting actors and financial institutions, that are: Danske Bank, Norges Bank, Finansdepartementet, Swedbank, DNB, SSB, Handelsbanken, Nordea, SEB and NHO. In addition, we compare our predictions to a traditional time series autoregressive model, which is a commonly used forecasting tool. This model is mainly included as a benchmark for all the predictions.

The results reflects that our best deep learning model is performing very well, compared to the institutions and the autoregressive benchmark model. Although, our model shows weaknesses on forecasting for the year of 2020 where we observe a dramatic fall in the economy. In crisis periods, such as the COVID-19 pandemic, we clearly see the advantage of utilizing methods such as experience, judgment and discretion in combination with models. To summarize, based on our overall evaluation, we conclude that deep learning algorithms shows huge potential and should be considered as a valuable tool for predicting the growth of the Norwegian economy.

Abbreviations

AR - Autoregressive Model

CNN - Convolutional Neural Network

ED - Encoder-Decoder

GDP - Gross Domestic Product

LSTM - Long Short-Term Memory

MAE - Mean Absolute Error

MSE - Mean Squared Error

MLP - Multi-Layer Perceptron

RNN - Recurrent Neural Network

Table of Contents

List of Figures

List of Tables

1 Introduction

In this chapter, we cover the introduction for this master thesis, which includes a problem statement that describes our research question, a brief literature review of similar studies and the structure of the remaining chapters.

Macroeconomic forecasts are essential for understanding the economic situation in a country. Gross domestic product (GDP) is a measurable parameter that reflects the current economic state. Usually, most countries desire stable economic growth. Estimates of macroeconomic measures that anticipate future growth are important for businesses, financial institutions and decision-makers within politics and the industry in general (SSB, 2022a). For example, the expected GDP growth is an important factor for the government and the central bank to take into consideration when they are respectively making decisions on the national budget and the key interest rate. Another example is that, companies must be aware of the macroeconomic risk of investments, where the GDP growth may have a huge impact on the profitability. In other words, an accurate prediction of the GDP growth can lead to a great advantage for making the right decisions.

1.1 Problem Statement

This thesis aims to forecast the Norwegian mainland GDP growth, using deep learning algorithms. The specific algorithms we use are: convolutional neural networks (CNN), recurrent neural networks (RNN), long short-term memory (LSTM), CNN-LSTM encoder-decoder (ED), RNN-RNN ED and LSTM-LSTM ED. We train our models mainly on macroeconomic- and financial data. Our complete data set includes 83 variables (an overview of all data can be found in Appendix A) with quarterly-, monthly and daily observations from December 1995 to December 2020. The performance of our deep learning models will be compared with -each other, -the observed Norwegian mainland GDP growth, and -a traditional time series autoregressive (AR) model. After the deep learning model examination, we pick the best performing algorithm as our champion model. This model then gets evaluated based on comparisons to predictions performed by leading forecasting actors and financial institutions. These are all participants of the macroeconomic forecasting competition, “Samfunnsøkonomenes Prognosepris”, organized by associate professor at BI, Genaro Sucarrat. Hereafter, we refer to this competition by its abbreviated Norwegian name, Prognoseprisen. This is a yearly competition, where the contenders aims to predict a variety of macroeconomic measures, including Norwegian -private consumer growth, -unemployment rate, -inflation rate and -GDP growth. The participants consists of: Danske Bank, Norges Bank, Finansdepartementet, Swedbank, DNB, Handelsbanken, SSB, Nordea, SEB, NHO, LO, CAMP, NAM, IMF and OECD (an URL link to each participant can be found in Appendix C).

Deep learning is complex and is often received as a black box in contrast to traditional time series forecasting methods. Applying deep learning algorithms to forecasting problems in general is regardless a very hot topic. However, to the best of our knowledge, we have not seen any research on applying this technology to forecast the Norwegian mainland GDP growth. Thus, we believe this investigation could be very exciting and we hope this thesis can be appealing for a variety of individuals, stakeholders and institutions. Based on our timeline and scope, the following research question will be investigated:

How well does deep learning algorithms perform on forecasting Norwegian mainland GDP growth?

By exploring this research question, we examine if the complex technology of deep learning can perform on par with- or even better than the traditional time series AR model, and the predictions performed by the participant of Prognoseprisen. Historically, GDP forecast models have mostly been based on time series regressions. When there are limitations on data, which is often the case with macroeconomic time series data, this method excels and can provide very useful predictions. However, there have been some development in the later years of employing deep learning algorithms in the field of macroeconomic forecasting. Deep learning usually thrive when there are non-linearity between multiple variables. We collect macroeconomic- and financial data that we believe correlates with the Norwegian economy. Then, it is up to the deep learning algorithms to catch the underlying patterns. We believe that in the future where big data takes over and more data will be accessible, deep learning will play a key role in forecasting in general and has the potential to be implemented as common method in the macroeconomic forecasting toolbox.

As Business Analytics students, who both have bachelor's degrees in Business Administration, we think this is a very interesting and relevant topic. In this research, we are able to combine our understanding of macroeconomics with our technical skills. However, the thesis will focus on the technical aspects rather than the macroeconomic theory.

1.2 Literature Review

Forecasting time series data with machine learning algorithms and its sub field deep learning has become a popular topic in the recent years. Especially as the field has advanced and the amount of data available to businesses and researchers has grown in the past 10 to 15 years. The traditional approach to forecasting macroeconomic data has been done by using the AR model, moving-average model, and the combination of the models; autoregressive-moving-average and autoregressive integrated moving average. Research on forecasting macroeconomic data by machine learning and deep learning has increased with the growth of the open source community and the amount of information and powerful libraries created and shared, having said that, it is still considered as a relatively small field.

Tkacz and Hu (1999) and Tkacz (2001) applied a simple artificial neural networks with one hidden layer to forecast Canadian GDP growth. Both articles concluded that forecasting GDP growth with artificial neural networks, outperformed traditional approaches when it comes to forecasting GDP growth in the long run (a year ahead). However, no improvements were seen when forecasting GDP growth in the short-run (one quarter ahead). In the late 1990s and early 2000s, deep learning and artificial neural network research was generally characterized by small data samples and simple networks (usually only one hidden layer). Tkacz and Hu (1999) outline small sample size as one of the issues in their paper, since artificial neural networks generally needs big sample sizes to perform well.

Smalter Hall and Cook (2017) predicted monthly unemployment in the US, using different neural networks architectures. This paper is relevant to our master's thesis because it uses architectures that we plan to use in our thesis: the LSTM network, which is a variant of an RNN, and an ED-network, which is an extension of the LSTM architecture and is a member of a broader class of networks known as sequence-to-sequence models. Hall and Cook (2017) forecast 3, 6, 9, and 12 months ahead using only historical unemployment data, and all the models/architectures used in the research outperform the Survey of Professional Forecasters, with the ED architecture being the best.

Szafranek (2017) forecasts monthly Polish headline inflation by implementing a combination of 10 000 bagged single hidden-layer feed-forward artificial neural network and used a real-time data set of 188 potential explanatory variables. The author concluded that the forecast combination of bagged single hidden-layer artificial neural network outperforms traditional statistical models.

Jung et al. (2018) published a working paper at the International Monetary Fund in which they investigated the use of machine learning algorithms and deep learning models to forecast the GDP growth of advanced G7 economies and several emerging economies. The deep learning model they used is elman neural network, which is an extension of RNN. The machine learning algorithms they used are elastic net and super learner. In general, all machine learning algorithms outperformed traditional statistical methods, although the elman neural network model performed the poorest of all the models. This paper provides a solid foundation and understanding of relevant macroeconomic and financial data. This will provide a solid starting point, as we will be able to collect the same data for the Norwegian economy that they utilized in their paper.

1.3 Thesis Structure

The thesis proceeds as the following structure. In chapter 2, we present theory relevant for this thesis. Chapter 3 contains a description of our data and challenges that comes with it. Next, in chapter 4, we elaborate on the methodology for building our models. In chapter 5, we go thought the design of our specific algorithms. Further, in chapter 6, we analyze the results of our deep learning models and evaluate the performance of the predictions. In chapter 7, we discuss our main findings, limitations, strengths and weaknesses and a proposal for further research. Lastly, in chapter 8, we draw a conclusion based on our research question.

2 Theory

In this chapter we present relevant theory for the master thesis. First, we elaborate on the GDP. Then we cover some of the basics within time series regressions, and, in particular, the AR model. Finally, we cover relevant theory within the field of deep learning and some specific algorithms.

2.1 Gross Domestic Product

GDP is a quantitative indicator of the market value of all goods and services produced during a certain period of time in a given country. In other words, it is a measure of overall economic activity. GDP is therefore regarded as one of the most important summary factors for understanding the economic state of a country (Callen, 2020). In Norway, the GDP is calculated in the national accounts. The Norwegian national accounts adhere to strict international guidelines which enables economic comparisons with other countries (SSB, 2022b).

It is common to separate GDP by nominal- and real GDP. Nominal GDP is given in current prices, without adjustment for inflation. On the other hand, real GDP is adjusted for inflation. Inflation may vary between countries and over time. Real GDP is therefore a better measure for comparing GDP across countries and time periods. Hence, with real GDP we can determine whether a change in GDP is due to change in production of goods and services, or is simply due to a change in prices. There are three ways of calculating the GDP which all amount to the same outcome (Y, expressed in Equation (1)). The first method, called the income approach, is calculated by adding together the total income generated by goods and services. The second approach, called the value added approach, consists of calculating an industry output and subtracting its intermediate consumption (the goods and services used to produce the output) to derive its value added. Finally, what is known as the expenditure approach, consists of adding together all groups of expenditure household, business and the government (SSB, 2022b). The last calculation method is the most common and is given by

$$Y = C + I + G + NX, \tag{1}$$

where Y is the GDP, C is consumption, I is investment, G is government purchases, and NX is the net export, that is, the total value of all exported goods minus the total value of all imported goods. In Norway, due to its large offshore oil and gas sector, it is common to separate GDP by mainland GDP and total GDP. Norwegian mainland GDP includes production from all industries, except oil and gas extraction, pipeline transport and foreign shipping (Dette er Norge-redaksjonen, 2021).

In this thesis, our data analysis and predictions concern the quarterly volume change of Norwegian mainland GDP, denoted growth_q for quarter $q = 1, 2, \ldots$. This measure is calculated by taking the percentage volume change of the market value from the same quarter in the previous year (SSB, 2022b), that is

$$\text{growth}_q = \frac{\text{mGDP}_q - \text{mGDP}_{q-4}}{\text{mGDP}_{q-4}} \times 100, \quad \text{for } q = 1, \ldots, 4T, \tag{2}$$

where mGDP_q is the Norwegian mainland GDP in the qth quarter in our data set, and T is the number of years in our data set. The measure we use in our model evaluation is the yearly volume change in GDP, which we denote ygrowth_t for year $t = 1, \ldots, T$, is defined by

$$\text{ygrowth}_t = \frac{1}{4} \sum_{q=4t-3}^{4t} \text{growth}_q, \quad \text{for } t = 1, \ldots, T. \tag{3}$$

In the next section we look at one traditional time series forecasting method that is often used to forecast macroeconomic parameters such as GDP growth. We introduce this method primarily because it will serve as a benchmark for various other deep learning models that we use for forecasting.

2.2 Traditional Time Series Regression

Series of data points ordered by time is called a time series. Predictions based on time series are called forecasting. There are many different methods of forecasting. Here we discuss one of the most traditional methods. Before we jump into this method, we must be aware of an expression called stationarity. Stationary time series have constant statistical properties such as mean and variance. It does not mean that the time series does not change over time, although the way it changes does not change. If the data is non-stationary we need to perform some extra preprocessing of the data. Non-stationary time series data produces unreliable results which leads to poor understanding. The solution to this problem is to transform the time series data so it becomes stationary (Koenecke, 2020).

2.2.1 The Autoregressive Model

AR models are regression models for time series data. In the AR model, the outcome variable at time t depends linearly on its own previous values and a stochastic term. The stochastic term is a mean zero random variable, often called the error term or noise term. The AR model is lagged by a specified number of time steps (Koenecke, 2020). For example, AR(1) has a time lag of one time step. In general, an AR model of order p, AR(p), may be expressed as

$$Y_j = \theta_0 + \theta_1 Y_{t-j} + \cdots + \theta_p Y_{j-p} + \varepsilon_j, \tag{4}$$

where Y_j is the outcome variable at time j, which, in our case, is quarterly volume change in Norwegian mainland GDP, that is, $Y_j = \text{growth}_j$ for $j = 1, \ldots, 4T$, with growth_j as defined in Equation (2); $\theta_0, \ldots, \theta_p$ are the parameters of the model, p is the number of time lags, and ε_t is the stochastic error term assumed to have a mean value of zero.

2.3 Deep Learning Concepts

Deep learning can be seen as a subset of machine learning, and machine learning is itself a subset of artificial intelligence. This is visualized in Figure 1. Deep learning is based on learning and improving on its own by discovering patterns and underlying contexts. It uses artificial neural networks that are designed to resemble how the human brain thinks, learns, and draws conclusions. The artificial neural networks associate inputs and outputs using intermediate layers, to model non-linear relationships. One of the main differences between machine learning and deep learning is when there is lack of domain understanding. For example, analyzing data such as images, sound, video or text, deep learning techniques tends to outshine other machine learning algorithms, as you have to worry less about feature engineering

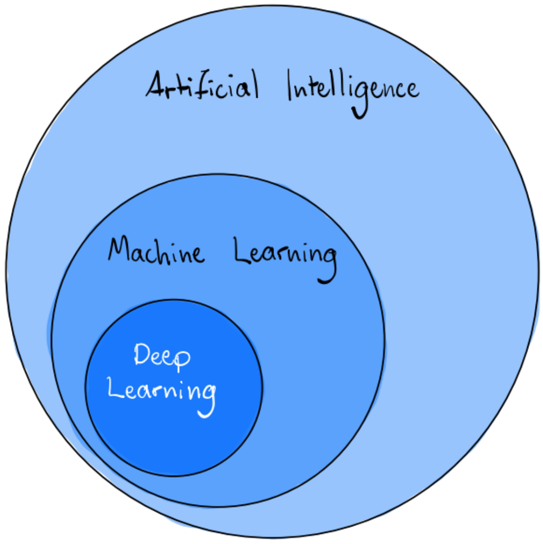

Figure 1: Deep Learning.

(Koenecke, 2020). Before we dive further into deep learning, we shift our focus towards a basic neural network, often called vanilla neural network.

2.3.1 Vanilla Neural Network

Vanilla, in the context of artificial intelligence, means standard, usual or unmodified versions of something. One can think of a vanilla neural network as an extension of regression. The difference is an extra added layer between the inputs and the output. This extra layer is called a “hidden” layer. This is because, the neural network itself takes care of all the calculations behind the scenes and the layer remains hidden (Holzbauer, 2019). A vanilla neural network with one hidden layer is illustrated in Figure 2.

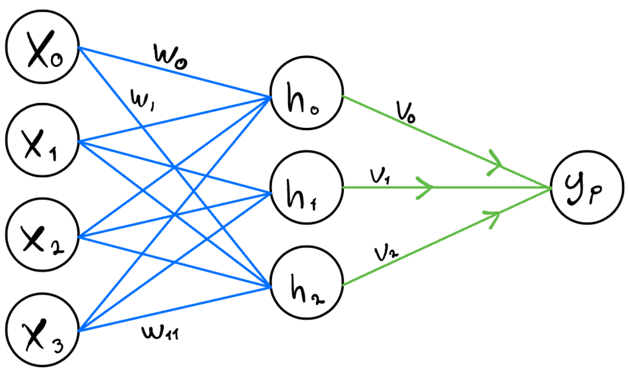

Figure 2: Vanilla Neural Network

In Figure 2, we denote the inputs $x_0, \ldots, x_3$. These are connected to weight vectors $w_0, \ldots, w_{11}$, which connects to the hidden layer containing of neurons $h_0, \ldots, h_2$, that is connected to the output layer y_p through a new set of weight vectors $v_0, \ldots, v_2$. The number of neurons in a hidden layer can vary. Thus, the designer of the network can simply create as few, or as many neurons as they wish (Holzbauer, 2019).

2.3.2 Deep Neural Network

After covering the vanilla neural network, we can continue towards deep learning. Deep neural networks consist of three different types of layers, just as in the vanilla neural network; the input layer, hidden layers and the output layer. The difference is that, in deep neural networks we have multiple hidden layers, hence "deep" learning. The more hidden layers you add, the deeper the network is. A network where each hidden layer has multiple neurons and where all the neurons in one hidden layer are connected to the neurons in the next layer is called a multi-layer perceptron, often abbreviated as MLP (Goodfellow et al., 2016). An illustration of a deep neural network can be seen in Figure 3.

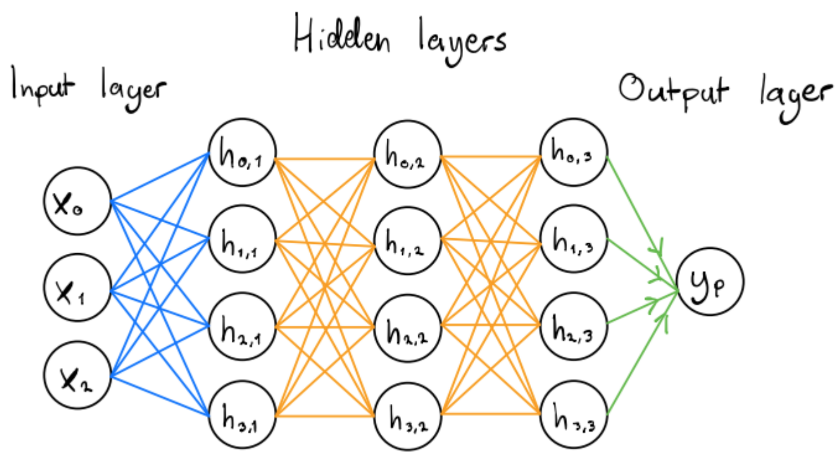

Figure 3: Deep Neural Network

Next, we cover some important concepts within deep learning, before we present some specific algorithms. These concepts are not important to understand for the results in isolation, however, they are essential for understanding how the models are built and how they work.

2.3.3 Activation Functions

What makes the neural networks architecture so powerful and fundamentally different from linear regression, is that for the output of each hidden layer, there is applied a non-linear "activation function". There are several different activation functions. In our deep learning models, we are using the hyperbolic tangent function (tanh) (Hansen, 2019), the gaussian error linear unit (GELU) (Hendrycks and Gimpel, 2016) and exponential linear unit (ELU) (Clevert et al., 2015). The shape and equation of the activation functions are visualized in Figure 4.

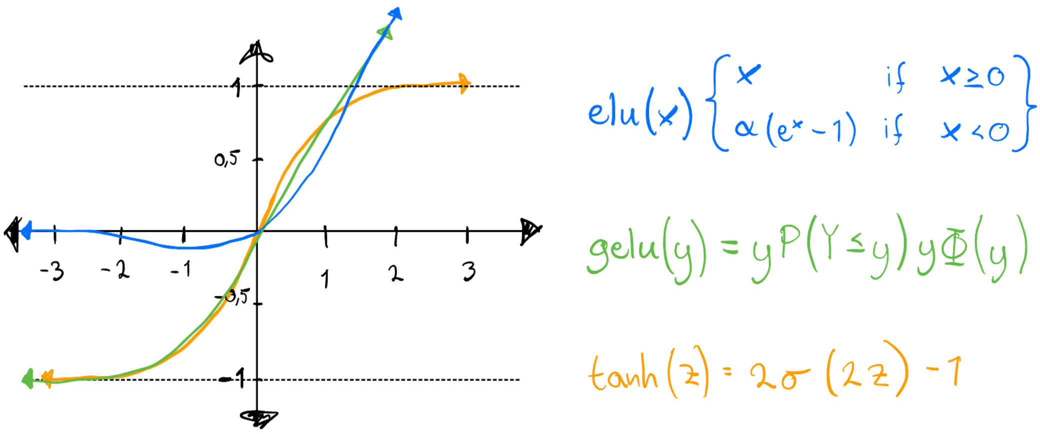

Figure 4: Activation Functions

We will not go more into detail on the specific equations for each of the activation functions as we consider it to be out of scope. However, they are important for the networks as they define how the weighted sum of the inputs is transformed into an output from a node or nodes in a layer of the network (Holzbauer, 2019).

2.3.4 Forward Propagation

Forward propagation is a quite intuitive method to feed data into the network. As the name suggests, the input data is fed in the forward direction through the hidden layers to the outputs. As the input data moves through the hidden layers, it gets processed as per current activation function, it then passes the information to the next layer and so on (Iuhaniwal, 2019).

2.3.5 Weights & Biases

Recall from Figure 3, where each neuron in the network is connected with each other and the connection arrows are associated as different weights. These weights are often called the "tuning knobs". The weights decide how much of the activation from one neuron is carried over to the next neuron. Thus, we can say that the weights convey the importance of the feature in predicting the output value. Bias is used for shifting the activation function towards right or left. The weights and the bias are both adjustable parameters in a neural network, this means that they can be adjusted by the user (Ganesh, 2021).

2.3.6 Loss Function

The loss function in a neural network quantifies the deviation between the predicted values and the observed values. There exists multiple different loss functions, a common one for regression problems is the mean squared error (MSE) (Koenecke, 2020). The calculation of MSE may be expressed as

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2, \tag{5}$$

where y_i is the observed value of observation i, and $\widehat{y_i}$ is the predicted value of observation i and n is the number of predictions. We are looking for a small MSE which in that case shows that the predictions are good (predicted values are close to observed values).

2.3.7 Gradient Descent

Gradient descent is an optimization algorithm. It is often used in both machine learning and deep learning to find a local minimum point of a function. The way the algorithm works is that it takes repeated steps in the opposite direction of the gradient (objective to minimize) of the function at the current point which will lead to a local minimum. For computing the gradient we use backpropagation, which we explain in Section 2.3.8. How much the parameters are updated per iteration is called the "learning rate" (Kwiatkowski, 2021). An illustration of the gradient descent can be seen in Figure 5.

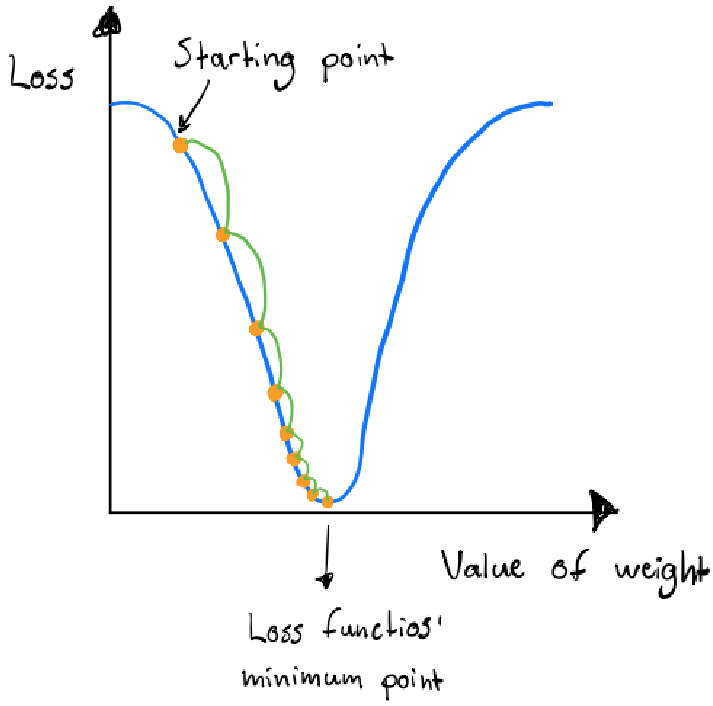

Figure 5: Gradient Descent

2.3.8 Backpropagation

Backpropagation is a fundamental building block and is used to train the neural network. While the input data is passed through the network by forward propagation, backpropagation passes the data backwards while adjusting the model's weights- and biases (Kostadinov, 2019).

2.3.9 Dropout

In a deep neural network there can in some cases be a very high amount of weight- and bias parameters that often can lead to overfitting. Dropout is a common layer in deep learning which is used to prevent this problem. The dropout layer randomly selects neurons which are simply dropped and ignored during the training of the deep neural network. It is the user that decides the dropout ratio which can vary between 0 and 1 (Singla, 2022). Dropout is randomly applied with a dropout ratio of 0.5 in the neurons with red crosses in Figure 6.

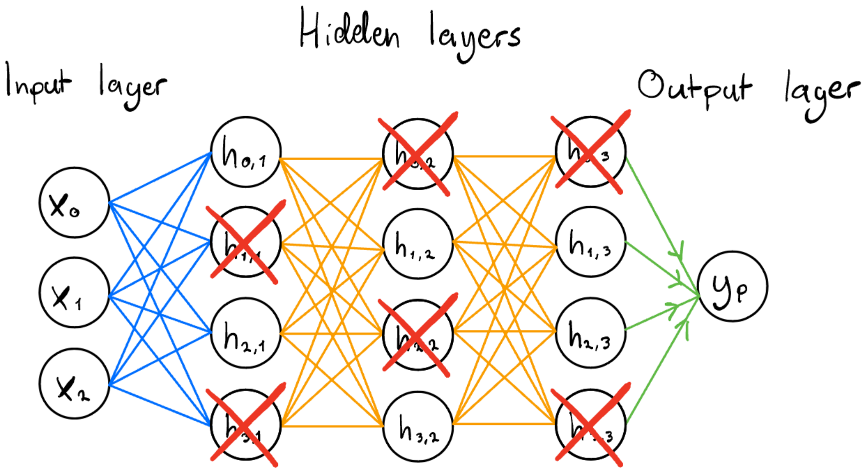

Figure 6: Dropout

2.3.10 Mask Layer

Masking is a way to tell the neural network that certain time steps in an input are missing, and thus should be skipped when processing the data. In our case of forecasting the Norwegian mainland GDP growth, at the time of our predictions (Section 4.3), we are missing the last quarter and the last month for each year. The masking layer simply enable us to utilize the last quarter and the last months for every other year in the training by masking the value of the fourth quarter of 2020 and December 2020 in the quarterly- and monthly data set (Abadi et al., 2015).

2.4 Deep Learning Algorithms

Building upon the concepts within deep learning that we covered above, we now turn to the specific algorithms that we have utilized in our forecasting models.

2.4.1 Convolutional Neural Networks

CNN is a class of an artificial neural network which is traditionally used in computer vision (analysis of digital images and videos), besides, they can be utilized in time series forecasting. The CNN consists of four different types of layers that are stacked on top of each other. These layers are the input layer, convolutional layer, pooling layer, and fully connected layer (Koenecke, 2020). The architecture of a CNN can be seen in Figure 7.

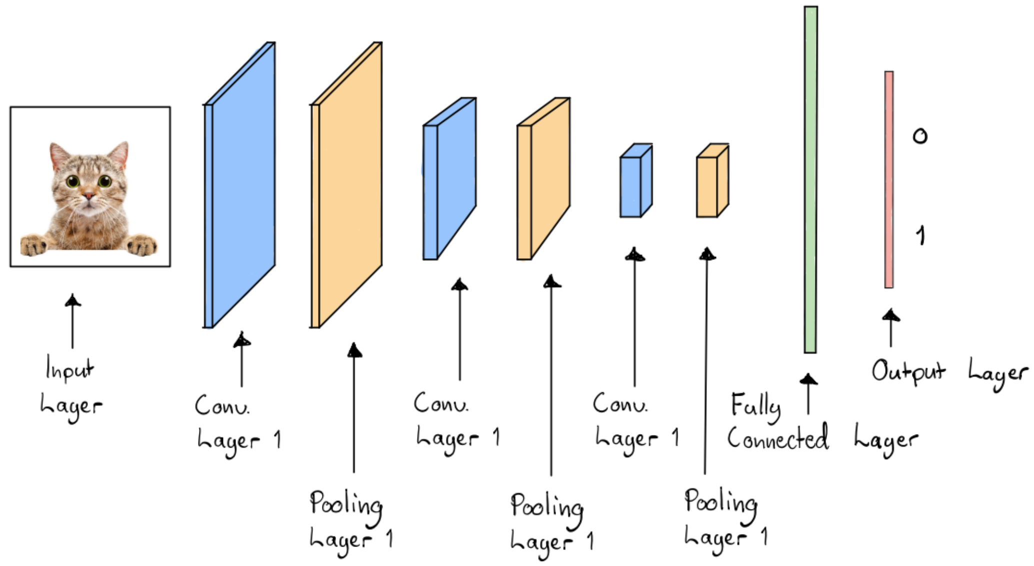

Figure 7: CNN Architecture

The input layer is the first layer in the CNN stack. There is not too much to say about this layer, other than it stores the raw input data. The convolutional layer is the key component of the CNN. This layer uses filters, often called "kernels" which is simply a smaller image than the input. Convolution is performed by taking a given section of the input and the filter in a sliding window approach until ultimately the whole input image is covered. The output of this process is called an activation- or feature map (Koenecke, 2020). An example of this operation is illustrated in Figure 8.

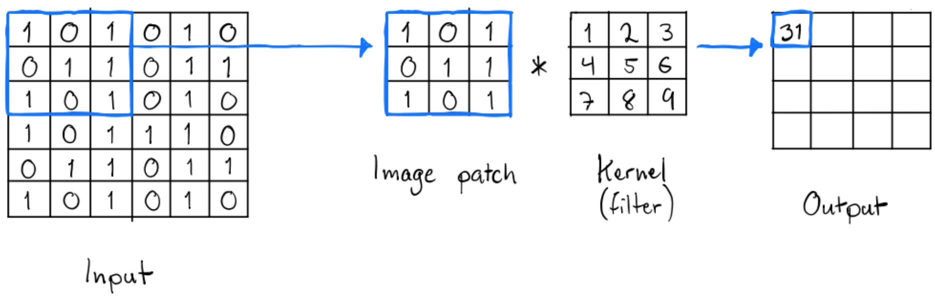

Figure 8: Convolution

The pooling layer reduces the size of the data that is fed into it. It is commonly done by a technique called max pooling, which takes the maximum value in the filter of the feature map. Stride is a parameter which modifies the amount of movement (Koenecke, 2020). An example of this operation can be seen in Figure 9.

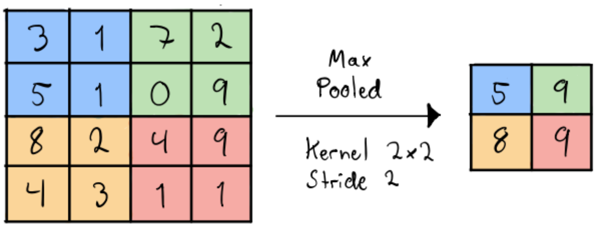

Figure 9: Max Pooling

The fully connected layer is the last layer before the output layer. The flattend matrix goes through a fully connected layer with some probabilities to classify the input (Koenecke, 2020).

2.4.2 Recurrent Neural Network

RNN is a different class of artificial neural network which is traditionally often used in natural language processing, however, it has shown its value in time series forecasting. RNN and CNN share much of the same aspects when dealing with sequences. The main difference is that RNN has the additional possibility of applying a recurrence at each of the time steps when processing sequences. In other words, RNN uses the current information on the input, along with the prediction of the last input (Koenecke, 2020). This operation is visualized in Figure 10,

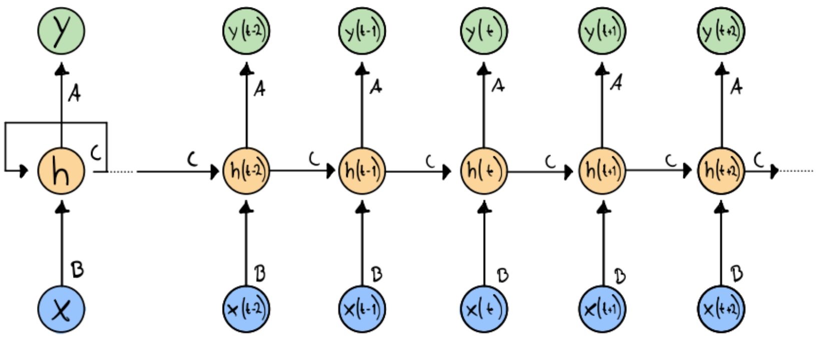

Figure 10: RNN Recurrence

where x is the input layer, h is the hidden layer, y is the output layer and A, B, C are parameters used to improve the output of the model.

In a feed-forward neural network (e.g. MLP or CNN) the decision is based on the current input. In contrast to RNN it cannot memorize the past data. However, RNNs can have trouble when the gradient of the activation function becomes very small. This is a common problem called "vanishing gradient descent". One way around this problem is to use a variant of RNN called LSTM (Koenecke, 2020).

2.4.3 Long Short-Term Memory

As we touched on above, LSTM is basically a more complex type of RNN. LSTM is traditionally used in deep learning and is well suited for classification and time series forecasting. Just as the standard RNN, LSTM contains feedback connections and can handle complete sequences in addition to single data points. It is common to break LSTM down to blocks or cells (Koenecke, 2020). An LSTM cell, is illustrated in Figure 11,

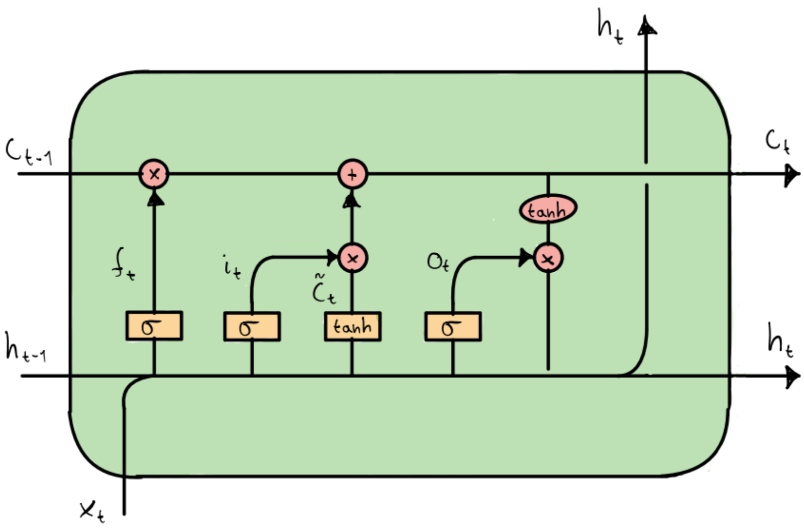

Figure 11: LSTM Cell

where x_t is the input layer vector, h_t the hidden layer vector, C_t the output layer vector, σ and $tanh$ are activation functions, i_t the input gate, f_t the forget gate, o_t the output gate and $\widetilde{C_t}$ is the cell state.

The LSTM cell can seem quite overwhelming, let us break it down. The LSTM cell memorizes values over arbitrary time intervals and the gates regulate how the information flows in and out of the cell. The forget gate decides how much information from input and the last output to keep, where 1 is to remember everything and 0 is to forget everything. The input gate decides how much information that should be stored in the current cell state, this gate prevents the cell from storing excess data. Lastly, the output gate decides how much information in the current cell state (current memory) that should be exposed to the output layer (Koenecke, 2020).

2.4.4 Encoder-Decoder

The ED model, often called the seq2seq model, is a type of an RNN as well. It is used to solve sequence to sequence problems. The ED model only accepts sequences of the data as input and returns the following sequence (not necessarily the same lengths) as output (Goodfellow et al., 2016). In our case, we can say that the model takes feature values from, for example, a twelve-month sequence and forecasts quarterly volume change of GDP growth (growth$_q$ from Equation (2)), which means it takes in a number of features, each of which contains twelve rows, and then returns a vector with four values.

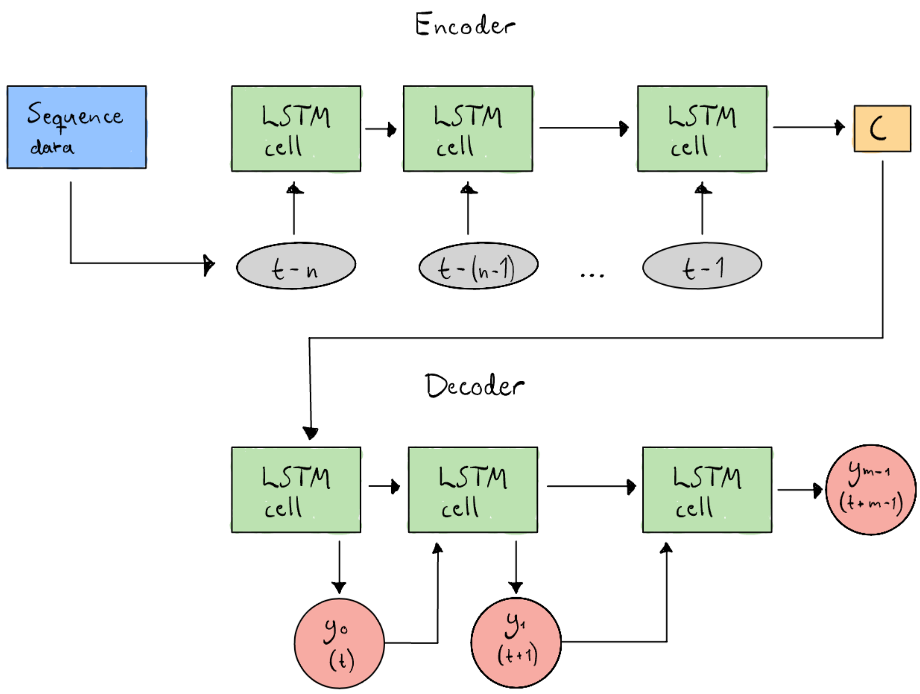

Figure 12: ED Architecture

The encoder and the decoder are two different networks that makes up the ED architecture and the model can be understood as follows. The encoder network learns or encodes the input sequence. As the network captures feature characteristics, it is stored in a vector. This vector is often called the context C. It can either be a single vector, or a sequence of vectors that summarize the input sequence. Next, the decoder network receives the context vector C (final hidden state) and learns or extract (decodes) the output sequence from it. Both the encoder and the decoder networks uses recurrent cells to handle the sequences (Goodfellow et al., 2016). An example of such a recurrent cell is the LSTM cell which we can recall from Figure 11. An example of an ED architecture can be seen in Figure 12, where t is time, n is input sequence length, C is the context vector, m is the output sequence length and y is the predicted values which in our case will be the quarterly volume change in Norwegian mainland GDP (growth_q from Equation (2)).

3 Data

Before we turn to the methodology and results of our forecast models we present the data we have used. Traditionally, deep learning algorithms perform best when handling large amounts of data, typically in the form of images, text or sound. Deep learning algorithms usually continue to increase in performance when more data is added to the model (Brownlee, 2016). The traditional economic indicators for predicting a macroeconomic measure such as GDP growth are commonly collected on low frequencies such as annual- or quarterly basis. To base our deep learning models solely on low frequency macroeconomic data would probably not give very accurate predictions. Thus, in addition to the quarterly macroeconomic data, we have collected monthly macroeconomic data and daily financial data that we believe contain valuable information about the Norwegian economy. In the following sections of this chapter, we present the data used for our predictions and evaluation, and challenges this type of data brings.

3.1 Data for Prediction

In this thesis, we forecast the Norwegian mainland GDP growth. Thus, to produce accurate predictions of this measure, it is essential to base our models on a data set with variables that captures the Norwegian economy. In addition to the variables themselves, we need a sufficient number of observations. For our predictions, we have collected data at three different frequencies, that are: quarterly, monthly and daily. We had to separate all of the data with different frequencies to be able to properly feed them into the deep learning models. We will look further into this in Chapter 4. Originally we thought of utilizing data from as far back in time as possible, however, for the sake of simplicity, variability in data availability and limitations to the period for this thesis, we decided to limit the data for the period from December 1995 to December 2020.

As we touched upon above, we have collected data at three different frequencies, which we used to build three different data sets. These three data sets contain observations on variables the are observed on a quarterly-, monthly- and daily basis. In cases where the same variables was available in multiple frequencies, we always chose to include the variable on most frequent form to maximize the amount of information. When it comes to macroeconomic data, it is important to keep in mind that the variables are usually subject to seasonal variation. All of our current variables are seasonal adjusted, which means that the variables are adjusted for their seasonal variation.

The quarterly data set contains 21 variables. This data set mostly includes macroeconomic measures including the dependent variable, Norwegian mainland GDP growth, as we recall from Equation (2). This variable is in the form of quarterly volume change, meaning that the growth represents the growth relative to the same period the year before. We have mainly collected the data for this data set from the national accounts section of SSB, which is an open source statistical bank. The national accounts provide an overview of the state and development in the Norwegian economy. Some of the key variables in addition to GDP growth are: consumption, investment, exports and imports, employment, wages, profitability in industries, and productivity. Norway have a small and open economy which is heavily dependent on trade with other countries. In other words, the economic state of Norway's trading partners will therefore be important for the growth in the Norwegian economy. Thus, this data set includes GDP growth for other countries such as Great Britain, USA, Denmark, Sweden, Germany and China. These variables are collected from Bloomberg which is a system that provides data specialized for the financial sector. This is not an open source, however, we luckily have access through our institution.

The monthly data set contains 17 variables, such as: the construction cost, inflation rate, unemployment rate and newly registered cars. These variables are typically measures which represents the consequences of the economic state. Bloomberg is the main source of this data set.

The daily data set is by far our largest and contains 45 variables. Obviously this is the data set with the most observations by a total of 6722. This data set includes exclusively financial data such as national- and international stock indices and commodity prices. The stock market are very much governed by expectations of the future, which we believe can be hold valuable information regarding the growth in the Norwegian economy. Bloomberg is the main source of this data set. A brief summary of the three data sets is given in Table 1. For a complete overview of the data see Appendix A.

Table 1: Data Set Summary

Frequency	**Features**	**Observations**
Quarterly	21	104
Monthly	17	301
Daily	45	6722

3.2 Data for Evaluation

The evaluation of our predictions is a very important process in our thesis. Recall the research question in Section 1.1, *How well does deep learning algorithms perform on forecasting Norwegian mainland GDP growth?* To measure the performance of our predictions, we first average our prediction from quarterly volume change (growth_q from Equation (2)) to obtain the yearly volume change (ygrowth_t from Equation (3)). Then, we compare on both revised- and unrevised GDP growth, to a benchmark AR(1) model and to some of the participants of Prognoseprisen. To maximize our basis for comparison, we have chose to only include the participants that have competed in Prognoseprisen for all years between 2013 and 2020, these institutions consists of: Danske Bank, Norges Bank, Finansdepartementet, Swedbank, DNB, Handelsbanken, SSB, Nordea, SEB and NHO. This approach excludes some of the participants including OECD, IMF and NAM. However, we believe this is the best way for comparison due to the fact that some years are easier to predict than others. In other words, some institutions may be unfairly penalized or rewarded for skipping a year. Norwegian mainland GDP growth are one, among other macroeconomic measures that the competitors are making yearly predictions on. The competition ranks the accuracy of the competitors predictions from lowest- to highest "absolute error". The absolute error is the difference between the predicted- and the observed value. This is the method we will use to rank the performance of our deep learning models. Likewise, this is the method we use to rank the performance of our deep learning models.

For our evaluation we use both revised- and unrevised yearly Norwegain mainland GDP growth. The revised data is collected from the national accounts in SSB and the unrevised data is colleccted from Sucarrat's own web page where he yearly publishes the results from Prognoseprisen. The evaluation data contains observations from 2013 to 2020. The financial institutions of the competition have a deadline of delivering their results by the end of December in year $t-1$. To make the comparison as fair as possible, we used the same approach for our predictions. In other words, we are excluding the fourth quarter data from year $t-1$ and monthly data from December in year $t-1$ in our predictions. Another important note is that some of the macroeconomic data we are using in our model are revised, which has to be considered as a weakness in sense of comparison. We will come back to this in the following section.

3.3 Data Challenges

Macroeconomic data are commonly revised. For example, the data in the national accounts for a specific month, quarter or year are revised in accordance to an ordinary publication and revision cycle (SSB, 2022b). Data being revised means that it is updated with more precise numbers after a certain amount of time. This creates a challenge for us as we are comparing and evaluating our forecasts based on the predictions performed by the institutions we discussed in Section 3.2. In other words, since the macroeconomic data we are using for our forecasts are more accurate then the data that was available at the time where the other institutions did their predictions, we can say that it may cause a data advantage in our favor. Hence, this has to be considered as a weakness in our evaluation of our deep learning models compared to the other predictions.

In our daily data set, we have collected financial data from multiple international stock exchanges. This creates a challenge because different exchanges in different countries have different holidays, resulting in features with missing data. There are multiple ways to solve this problem, however, we have simply imputed the missing value with the last available price. This is a method that are frequently used by many data providers and financial institutions for imputing missing values in financial time series (Kokic, 2001).

4 Methodology

In this chapter, we describe how we preprocessed the data and what Python libraries and frameworks we used to create our models.

4.1 Toolkit

We chose Python as the programming language for our thesis, the Pandas (pandas development team, 2020; Wes McKinney, 2010) and Numpy (Harris et al., 2020) libraries for data processing, and Mathplotlib (Hunter, 2007) for data visualization. To develop deep learning models, we used the Keras (Chollet et al., 2015) functional API in TensorFlow (Abadi et al., 2015) as the framework.

4.2 Data Preprocessing

The first step after downloading the data was to merge features with equal frequencies into a single data frame. For quarterly and monthly data, this was rather simple, with little to no data cleaning needed. Preprocessing was more time consuming for the daily data set. To create the daily data set, we first had to locate the financial feature with the most dates, then concatenate the remaining features onto it. The next problem with the daily data set, is that the number of trading days per year varies. The number of trading days likewise differs between exchanges. In order to input the data into the deep learning models, we had to make an equal number of trading days for each year; this is because we need to "window" the data set, which we cover in depth in Section 4.2.4.

We solved this problem by writing a function that scales down the number of days in each year to be equal. We now go through this function by an example. Let us say that the number of days (rows) in 2011 is 260, in 2012 it is 258 and in 2013 it is 256. The function sums the number of rows in a year, for each year in the data set by selecting the year with fewest number of days (2013) and calculates the difference between the other years (2011 and 2012) in the data set. The difference for 2011 would be equal to 4 ($260 - 256$) and for 2012 it would be equal to 2 ($258 - 256$). This means that for 2011 and 2012, the function deletes respectively 4-and 2 days of data. The function assesses which rows from the data set that should be deleted; in general, it deletes rows with the most missing data across all the columns/variables for each row that needs to be deleted. We can filter the rows by year, month and days because our data set is in pandas data frame and our index is in date time format.

Finally, we standardized the data. Because the majority of our macroeconomic data had already been downloaded as growth rates, we transformed all other features (that was not downloaded as growth rates) to growth rates as well.

4.2.1 Mixed Frequency Problem

One common challenge with collecting time series data from various sources and domains is that the data is often sampled at different frequencies. Most machine learning models require all data to be passed at the same frequency in order to be trained. There are a number of approaches to tackling this problem, the most common method is to scale down all features to the lowest frequency available. In our thesis, we leave the data as it is and instead utilize deep learning techniques to build models that can handle mixed-frequency input, which we go through in Chapter 5.

4.2.2 Train-, Validation- & Test Split

Forecast accuracy can only be measured by how well a model performs on new data that was not utilized while training the model. When selecting a model, it is common practice to split the available data into two parts: training- and test data. The training data is used to train the model, while the test data is used to evaluate its accuracy. Because the test data is not used in the training process of the model, it should offer a reliable indication of how well the model will forecast on new data (Hyndman and Athanasopoulos, 2021).

In machine learning it is common to add one more split to the data, which is called the validation set. The validation set is used to avoid overfitting under the training period and help it to generalize. We will come back to how we used the validation set in Section 4.2.3.

We split the data into train- and test split, each split contains respectively seventy- and thirty percent of the data. The dependent variable's training data cover the years

1997 to 2012 (this is because we forecast one year ahead), whereas the independent variables' training data cover the years 1996 to 2011. The years between 2012 and 2019 were included in the test data for the independent variables, whereas the years between 2013 and 2020 were included in the test data for the dependent variable.

4.2.3 Cross-Validation

To evaluate our models performance and to track the effect of hyperparameter tuning throughout the building process we use, time series cross-validation, often referred to as cross-validation on a rolling basis or evaluation on a rolling forecasting origin (Hyndman and Athanasopoulos, 2021).

In time series cross-validation there are a series of test sets and corresponding training sets, consisting only of observations that occurred prior to the observation that forms the test sets. (Hyndman and Athanasopoulos, 2021). Figure 13 gives intuition on how time series cross-validation works in practice. The training data are further separated into two sets: train and test. The model is trained on the training set, then the model forecasts future data points, which are evaluated on the corresponding test set. The prior test set is then incorporated in the next training set, and subsequent data points are forecasted and assessed. The forecast accuracy for each model is calculated by averaging the error for all the test sets (Hyndman and Athanasopoulos, 2021). The error measure for each test set is mean absolute error (MAE), which is given in Equation (6).

In Section 4.2.2, we mentioned that we used validation sets in our training; in time series cross-validation, the test sets function as a validation set. Each time we made changes to our model, we trained and tested it using time series cross-validation and calculated the accuracy and compared it to the previous models result to see if we made any improvements.

Randomness is present in machine learning models. This can be through gradient descent, since computing the gradient step based on the entire data set is not feasible for large data sets and models. Only one or a small number of randomly selected training samples from the training set are used by stochastic gradient descent to update for a parameter in a given iteration (Shao, 2019). We can additionally introduce randomness through our model choices and what layers we include, for example through the drop out layer (recall Section 2.3.9. To ensure that the improvements in model accuracy were due to our adjustments rather than randomness, we utilized a function in TensorFlow that allows users to set random seed for the whole process. This allowed us to attribute improvements in our models to the changes we made.

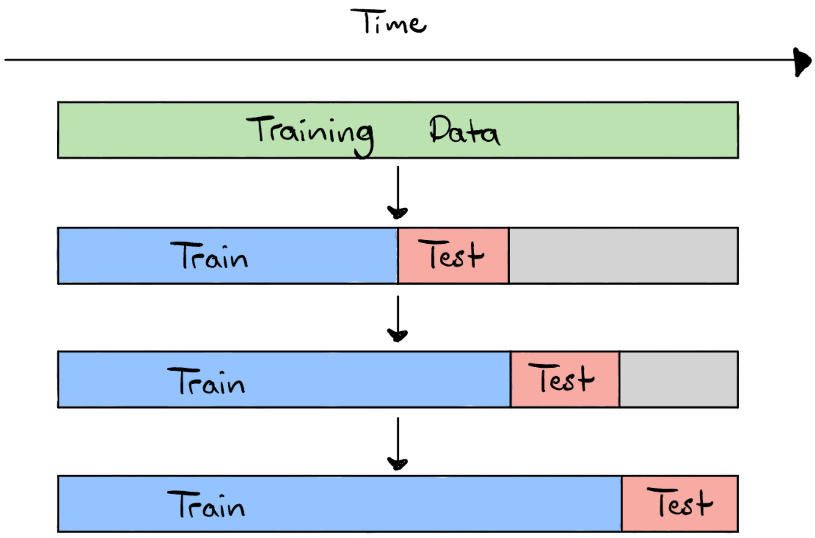

Figure 13: Time Series Cross-Validation

4.2.4 Window Method

To train and use deep learning models, we must first transform our time series data and frame it as a supervised learning problem using the moving window technique for multivariate data and multi-step forecasting (Bhatt et al., 2022). The simplest way to explain this concept is through an example of how we would preprocess the data if we were to build an univariate (one input variable) deep learning model. To learn the patterns and structure of the data, the deep learning models we use require an independent- and dependent variable, which is essentially what supervised learning means.

Figure 14 shows how vector X_1 is transformed by applying the moving window. In Transformation 1 the X_1 sequence has been changed, there are now four past values predicting one value in the future. What we did to the vector X_1 is called windowing, and the window size, often called the look back period, tells the model how many time steps into the past it should look to predict the next time step.

Transformation 2 in Figure 14 shows that we have a window size of four, however, now we predict two future values, this is called multi-step forecasting.

The transformed X_1 sequence in Transformation 1 have square brackets around the numbers for each row, this is what is referred to as a window. The window size determines how many numbers is contained in each window and the window size for each window in X_1 has to be equal, however, the window size for X_1 and Y can be different. The important part is that there should be equal amounts of windows between X_1 and Y. For example, if we consider an LSTM model that is used for language translation, a sentence in one language does not have to be the same length if translated into another language. What is important is that the number of sentences to be translated is equal in the training data. The same applies to time series.

Original Data

X1
1
2
3
4
5
6
7
8
9
10

Transformation 1

X1	Y
[1,2,3,4]	5
[2,3,4,5]	6
[3,4,5,6]	7
[4,5,6,7]	8
[5,6,7,8]	9
[6,7,8,9]	10

Transformation 2

X1	Y
[1,2,3,4]	[5,6]
[3,4,5,6]	[7,8]
[5,6,7,8]	[9,10]

Figure 14: Windowing, Univariate Time Series

In the case where there are multiple independent X values that are not in the same frequency as the dependent Y value, the windowing method can be applied. In Figure 15, we transform X_1, X_2 and Y, where three previous time steps predict one time step in the future, which can be seen in the Transformation 1. Data can likewise be windowed in such a way where we can forecast multiple time steps ahead, with multiple features. When multiple features are involved in forecasting multiple time steps it is called a multivariate multi-step forecast. The data in Transformation 2 in Figure 15 has been windowed in such way where we can forecast multiple time steps ahead. We are forecasting two values by looking back three time steps. Since the example data only contains ten rows, we have to skip some values in Transformation 2, this does not have to be the case when working with real life data.

Data in Transformation 2 in Figure 15 consist only of two rows, however, the window size for the independent variables are different from the dependent variable. This allows us to predict GDP growth, which is measured on quarterly basis, with, for example, daily data. We can window daily data and say that we want to predict growth_q (recall Equation (2)) for year 2014 by using daily financial data from year 2013. Our dependent value (growth_q) will have a window size of four and our independent values (financial data) will have a window size of 256 trading days. We apply this technique to all three data sets, where we get equal amount of windows per data set for train- and test data. In section 5.1, we explain how we utilize three different data sets in one model.

Original Data

X1	X2	Y
1	11	
2	12	31
3	13	
4	14	33
5	15	
6	16	35
7	17	
8	18	37
9	19	
10	20	39

Transformation 1

[X1, X2]	Y
$\begin{bmatrix}1,11\\2,12\\3,13\end{bmatrix}$	33
$\begin{bmatrix}3,13\\4,14\\5,15\end{bmatrix}$	35
$\begin{bmatrix}5,15\\6,16\\7,17\end{bmatrix}$	37
$\begin{bmatrix}7,17\\8,18\\9,19\end{bmatrix}$	39

Transformation 2

[X1, X2]	Y
$\begin{bmatrix}1,11\\2,12\\3,13\end{bmatrix}$	[33,35]
$\begin{bmatrix}5,15\\6,16\\7,17\end{bmatrix}$	[37,39]

Figure 15: Windowing, Multivariate Time Series

4.3 Data Structure for Training & Forecasting

Our deep learning models and the AR(1) model forecast quarterly volume change of Norwegian mainland GDP which is expressed in Equation (2). We then average these four predictions to obtain the yearly GDP growth which is expressed in Equation (3).

We use one year of data to forecast one year ahead, meaning we use the year $t-1$ to forecast the year t. Thus, we use 256 trading days of the daily financial data, 12 months of the monthly data and 4 quarters of the quarterly data. In Prognoseprisen, the competition participants had to submit forecasts for year t no later than December 31 in year $t-1$ until 2020. However, the deadline for predictions for 2020 was extended to May 31, 2020. This has some practical implications for our forecasts and what data we can include. Since the deadline for the competition until 2020 was December 31, this means that we would not have all the monthly and quarterly data at this point in time. SSB publishes the quarterly national accounts about 40 days after the end of the given quarter. Thus, we would not have the fourth quarter for all our quarterly data before the deadline. Our monthly data has the same issue, we would not have the last month of the year for all the monthly data. Hence, we do not use last quarter of our quarterly data and last month of our monthly data for year $t-1$ when forecasting the GDP growth for year t. Between each forecast period, we update our models with new data, this means if we are forecasting t with data from $t-1$, we update our model with $t-2$ data. This is because we train our models with data up until 2011, then we forecast 2013 with data from 2012. Then, when we forecast 2014, we use data from 2013 for the forecasts, although, before we do the forecast, we update our models with data from 2012. These steps are repeated for all forecasts until 2020. Due to the COVID-19 pandemic, the deadline for Prognoseprisen was extended to May 31, 2020. This meant that participants could utilize more data to forecast GDP growth for 2020. For a better basis for comparison, we had to modify our models for the 2020 prediction. Otherwise we would be at a data disadvantage. Thus, we trained our models for the 2020 forecast with data from 1996 to 2018. We used daily financial data from January 2, 2019 to May 31, 2020, monthly data from January 2019 to April 2020, and quarterly data from first quarter 2019 to first quarter 2020 to forecast GDP growth in 2020.

5 Model Design

This chapter contains the design of the different network architectures that we used in this thesis. There are numerous ways to build a neural network, and the architecture can vary based on the problem and the data set. The architectures presented in this chapter are the product of trial and error.

5.1 Standard Architecture

The building process began by attempting to develop a good prediction model using only one data set, meaning, designing models utilizing only daily-, monthly-, or quarterly data. We did not combine all three data sets until we had models that could give us acceptable results using only one data set.

Time series cross-validation technique helped us to understand what type of network designs worked for our forecasting problem. Deeper networks, meaning networks with multiple hidden layers, did not perform better than simple networks with fewer layers. We discovered that hyperparameter tuning and finding the right activation functions for different type of layers increased the model performance, hence, this is what we shifted our focus towards. After we found the network architectures that performed well on the daily-, monthly- and quarterly data sets, we combined these networks into one model.

Tensorflow's functional API allowed us to create a model that can handle multiple inputs. Thus, we could utilize daily-, monthly-, and quarterly data in the same model. We used the windowing technique to ensure that each data set had an equal number of windows. Figure 16 shows the general network design for our models. Masking layers were implemented in all networks, besides the networks that contained a CNN layer. This is because Keras does not support masking layers in CNN. For RNN and LSTM we imputed the last quarter and the last month of test data with the number 20 and masked the value. With CNN networks we just imputed last quarter and last month for all test sets with 0. The block N Layers, depending on what layers we used in the CNN/LSTM/RNN layer, can be one or multiple dense-, flatten- and/or max-pooling layer(s). The N Layers are concatenated into one layer; for a CNN model, the last N Layer is a flatten layer, whereas for an LSTM or RNN, it is a dense layer (the name “dense layer” is used in Tensorflow instead of “hidden layer” that can be seen in Figure 3). The concatenated layer is fed further down to two layers before we get our desired output which is quarterly volume GDP change (recall Equation (2)). For networks that are not ED, the K layers would just be one dense layer. For an ED-network this would be multiple layers in following order: a repeat vector layer, an RNN or LSTM layer, and a time-distributed dense layer. The last dropout layer and output layer for an ED-network would be wrapped in a time-distributed layer.

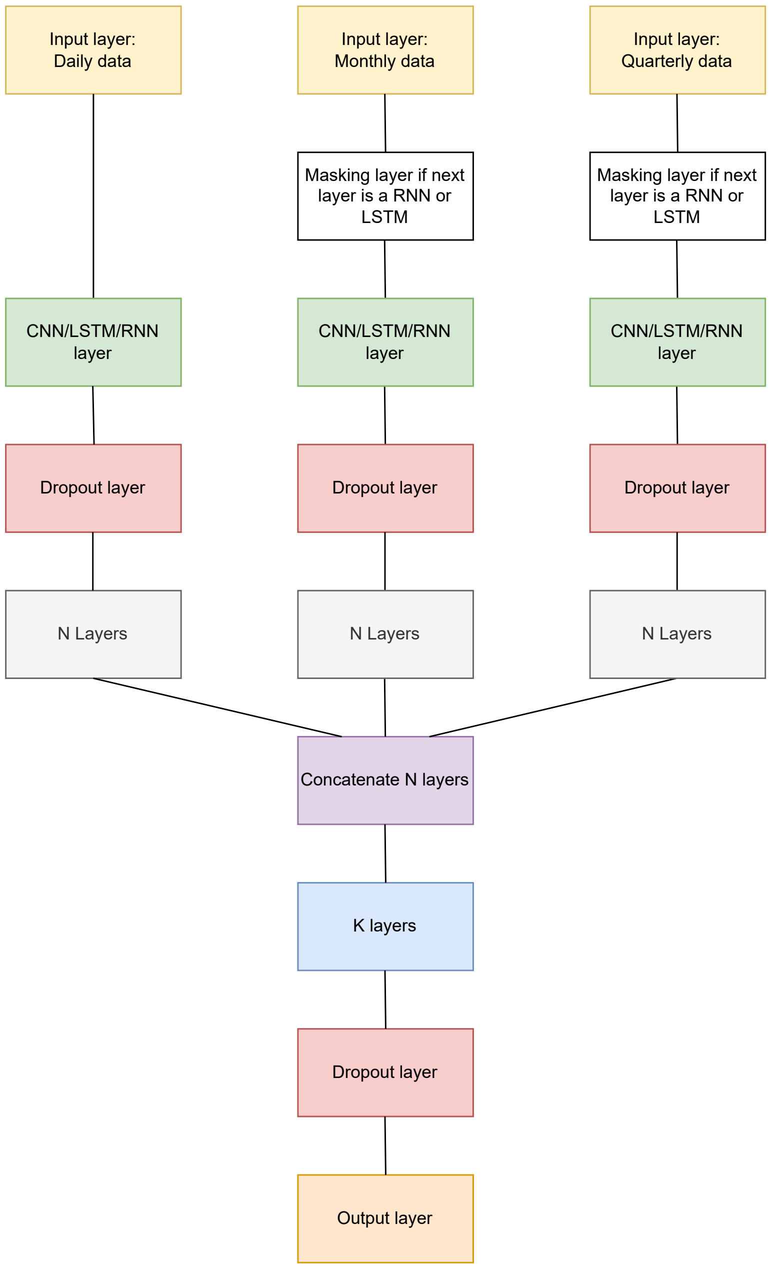

Figure 16: Standard Network Design

For all our models we used the Adam optimizer Kingma and Ba (2014) (Adam optimizer is an extension to the classical stochastic gradient descent algorithm) as the optimazation algorithm and root mean squared error for our loss function (which is the root of MSE in Equation (5)). For CNN-, LSTM- and RNN layers we used the activation function gelu, for dense layers we used the activation function elu and for our final output layer we used the activation function tanh (recall Section 2.3.3). The input shape for daily-, monthly- and quarterly data is equal across all models, except the models that forecast GDP growth for 2020. The input shape for daily data is $[256,45]$ which means that the window size is 256 and that there are 45 features. The monthly data input shape is $[12,17]$ and the quarterly data input size is $[4,21]$. The input shape for models that forecast GDP growth for 2020 is $[362,45]$ for daily data, $[17,17]$ for monthly data, and $[5,21]$ for quarterly data.

5.2 The Convolutional Neural Network Model

The 1 dimensional (Conv1D) CNN layer has a filter size of 128 and a kernel size of 72 for daily data input, a filter size of 84 and a kernel size of 8 for monthly data input, and a filter size of 64 and a kernel size of 2 for quarterly data input. The dropout layer has a 0.3 dropout rate that is equal across all input data. The pool size for the max pooling layer for daily data input is 64, 4 for monthly data, and 1 for quarterly. The last layer for all data inputs is a flatten layer. The three flatten layers are then concatenated into one and passed on to a dense layer with 112 neurons, followed by a dropout layer, and finally an output layer.

5.3 The Recurrent Neural Network Model

For the RNN model we use a simple RNN layer which has 128 neurons for daily data input, 84 neurons for monthly data and 64 neurons for quarterly data. The RNN model has a masking layer before the simple RNN layer for monthly and quarterly data where the masking value is equal to 20. The dropout layer has a 0.5 dropout rate that is equal across all input data for this model. The last layer for all data inputs is a dense layer which has 112 neurons for daily data, 72 neurons for monthly data and 56 neurons for quarterly data. The three dense layers are then concatenated into one layer and passed on to another dense layer with 100 neurons, followed by a dropout layer, and finally an output layer.

5.4 The Long Short-Term Memory Model

The LSTM model is almost an exact copy of the RNN model from the Section 5.3. The only difference is that the simple RNN layers are replaced with LSTM layers.

5.5 The Encoder-Decoder Models

The difference in hyperparameters between ED models and non-ED models are in layers that come after the concatenation layer. The part of the model before the concatenation layer is called the encoder and the part after the concatenation layer is called the decoder. For the CNN-LSTM ED model, the CNN part is the encoder and the LSTM part is the decoder. The layers until concatenation layers are exactly the same as in the CNN model described in Section 5.2. The difference is after the concatenation layer where we have a repeat vector that repeats the incoming input 4 times as seen in Figure 17, the following layer is an LSTM with 112 neurons and "return sequence" is enabled. Since we are now receiving multiple outputs from the LSTM layer, we need a way to apply a dropout layer and a dense layer to each output from the LSTM layer. We use a time-distributed layer that wraps the dropout layer and the dense layer. The dropout layers have a dropout rate of 0.3 and the dense layers have 64 neurons each. We have four output layers, where each output layer forecast single quarter GDP growth, meaning first output layer from the left in Figure 17 forecast Q1, second output layers forecast Q2, third layer forecast Q3 and fourth layer forecast Q4.

For the RNN-RNN ED model the part before the concatenation layers is exactly the same as in the RNN model described in Section 5.3 and the layers after the concatenation are almost exactly the same as in CNN-LSTM model, where the only difference is that we use a simple RNN layer instead of an LSTM layer and the dropout rate is set to 0.5.

For the LSTM-LSTM ED model the layers before the concatenation layers are exactly the same as in the LSTM model described in Section 5.4 and the layers after the concatenation layer are almost exactly the same as in the CNN-LSTM model, where the only difference is that we set the dropout rate to 0.5.

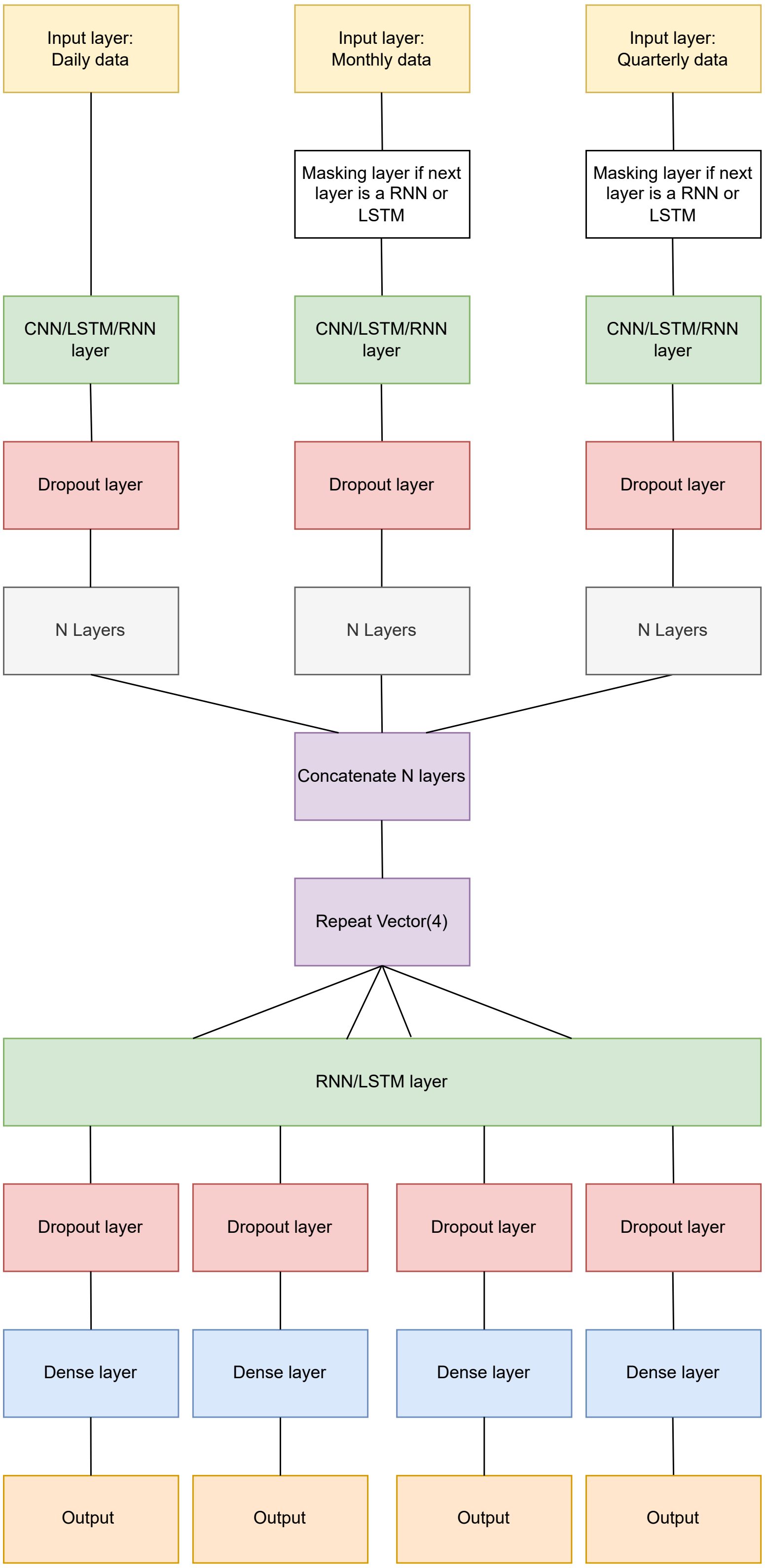

Figure 17: Encoder-Decoder network design

6 Results and Analysis

In this chapter, we first examine the overall results of our deep learning models which we discussed in the previous chapter. Next, we pick our best performing model, then we evaluate it based on comparisons with predictions performed by the participants of Prognoseprisen. In addition, we include a traditional time series AR(1) model that works as a benchmark for all the predictions. Our models are all built on the latest updated data available, that was collected in April 2022. Therefore, the possible data advantage discussed in Section 3.1, must be taken into consideration. To ensure a fair evaluation, we are performing comparisons based on both revised- and unrevised Norwegian mainland GDP growth. There are several possible measures for evaluating forecasting performance. In our analysis, we use MAE as the measure for ranking the performance of the predictions, where the average in MAE is based on our predicted years that consists of predictions from 2013 to 2020. MAE is a measure commonly used in evaluation of forecast performances. Furthermore, this is the measure Prognoseprisen uses in their evaluation. In addition to MAE, we include minimum-, maximum-, and median absolute errors in our tables as they contain information of interest for the overall evaluation. MAE is given by

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}\left|\mathrm{ygrowth}_i - \widehat{\mathrm{ygrowth}}_i\right|, \tag{6}$$

where $\mathrm{ygrowth}_i$ is the observed value of yearly GDP growth which is given by Equation (3) for year i, $\widehat{\mathrm{ygrowth}}_i$ is the predicted value of yearly GDP growth year i, and m is the number of years we are predicting (2013-2020) and the vertical lines represents the absolute value.

6.1 Deep Learning Model Examination

For the evaluation of our own deep learning models, we are measuring the performance by revised data exclusively. This makes sense, because all our models are trained on revised data. In total, we have developed six different deep learning models. This includes: CNN, RNN, LSTM, CNN-LSTM ED, RNN-RNN ED, and

LSTM-LSTM ED. In addition, we have developed an AR(1) model that is based on the same revised Norwegian mainland GDP growth values as for the deep learning models. However, recall from Section 2.2.1, the outcome variable for the AR model at time *t* depends linearly on its own previous values and a stochastic term. In other words, the AR model only uses one feature. Hence, the comparison with our deep learning models (which uses 83 features) is arguably not fair. Although, it mainly serve its purpose as a benchmark for the predictions. Figure 18 shows all our models predictions, relative to the observed revised Norwegian mainland GDP growth in all years between 2013 and 2020.

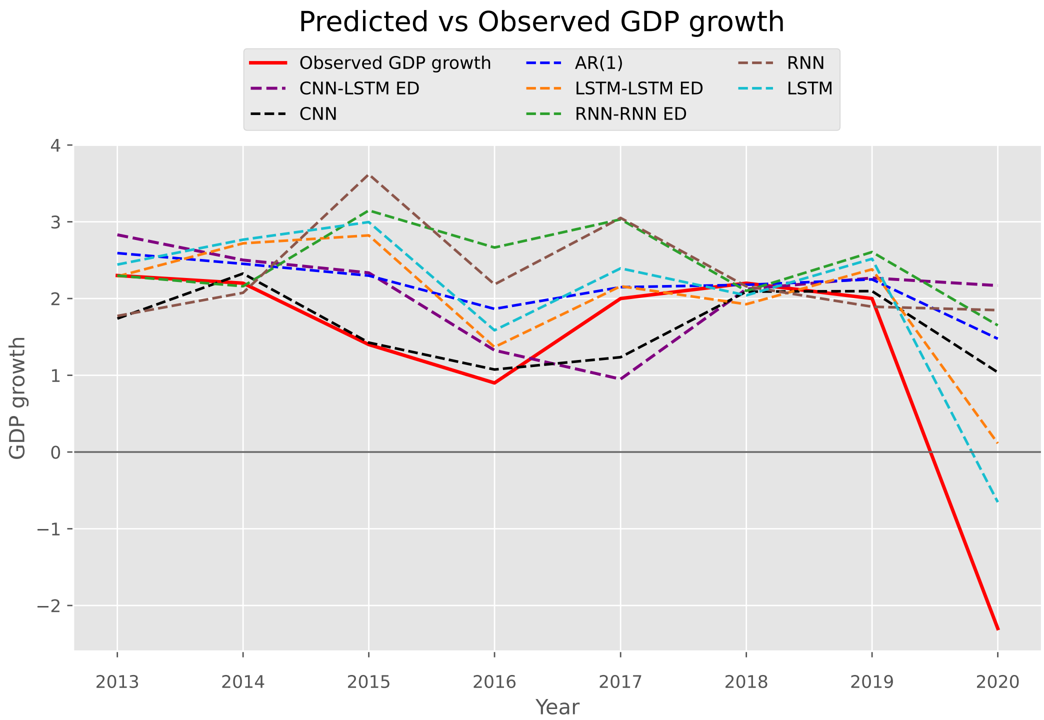

Figure 18: Deep Learning Models, Predicted vs Observed

All of the models predict fairly good for 2013 and 2014. In 2015, there is a downward trend for the observed GDP growth which separates the CNN model from the others. The CNN model produces accurate predictions in both 2015 and 2016 where the other models misses by quite a lot. In 2017, it is the AR(1) and the LSTM-LSTM ED that performs the best. In 2018, we see accurate prediction for all models. In this year, the observed GDP growth is 2.2% and all models makes predictions within the window of 1.93% and 2.18% (Table 22). In 2019, all models produces decent predictions, however, the CNN is the most accurate. In 2020, there is a dramatic fall in the graph for the observed GDP growth where it decreases from 2% to -2.3% (Table 24- and 26). Not very surprisingly, all of the models struggle to predict for this year. On the other side, they all at least predict a varying degree of decrease, although, the LSTM model is the only model that predicts a negative GDP growth by the value of -0.65% (Table 3). To summarize the key point of this figure, we can clearly see that the CNN model is the overall best model that consistently produces good predictions for most years in the period, with the exception of 2020.

Let us now study the errors of the predictions. We rank the performance of the models by the MAE which is defined in Equation (6). Table 2 shows the minimum- maximum- median- and mean absolute error for all the models in all years between 2013 and 2020. Figure 19 visualizes the yearly observations of the absolute errors between 2013 and 2020.

Table 2: Deep Learning Model Examination

	Rank	**Min**	**Max**	**Median**	**MAE**
CNN	1	0.03	3.34	0.15	0.65
LSTM - LSTM ED	2	0.02	2.41	0.43	0.71
LSTM	3	0.14	1.65	0.55	0.72
CNN - LSTM ED	5	0.08	4.47	0.48	1.01
RNN - RNN ED	6	0.00	3.95	0.82	1.18
RNN	7	0.04	4.15	0.79	1.19
AR(1)	4	0.02	3.78	0.27	0.83

Based on yearly observations between 2013 and 2020
All measures are based on the absolute error
Rank is determined by MAE, where a lower MAE results in a higher rank
AR(1) is included as a benchmark for the performances in general

As expected, based on the early impression from Figure 18, we can see in Table 2 that the CNN model ranks the best in terms of MAE by the value of 0.65. The next on the list is the LSTM-LSTM ED with a value of 0.71. The difference between these two models is small and does not fully substantiate the overall impression from Figure 18 where the CNN seemed to be superior. In this thesis, we only have 8 predicted values that can be used for evaluation. Our main measure for evaluating performance is the MAE. An average can, in general, be very sensitive to big outliers, especially with a series of few observations. As we see in Figure

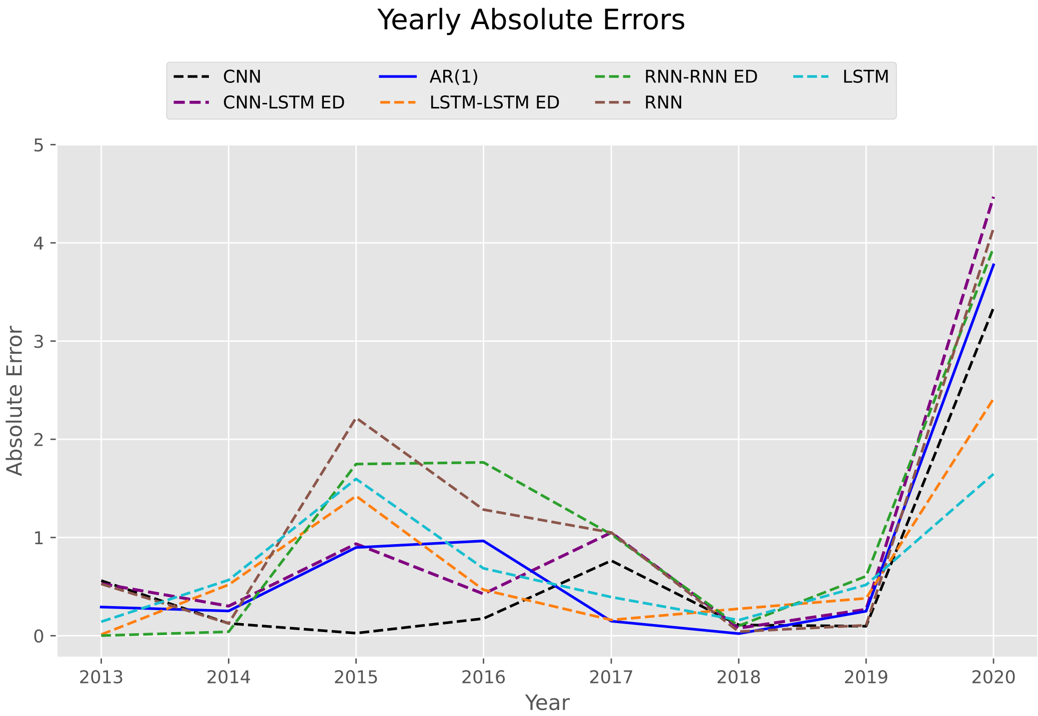

Figure 19: Deep Learning Models, Yearly Absolute Errors

19, all of the models predict poorly for 2020 making these observations outliers. Hence, maybe the median would be a better indicator for performance than MAE. If we turn to the median absolute error in Table 2, we can see that the CNN model is superior by the value of 0.15. The nearest is still the LSTM-LSTM ED model by the value of 0.43 which must be considered as a quite large gap and reflects the impression from Figure 18. To further investigate this problem we turn to Table 3, which shows all models predictions for year 2020 specifically.

Table 3: Deep Learning Models, 2020 Predictions

	Observed	**Predicted**	**Absolute Error**
LSTM	-2.30	-0.65	1.65
LSTM - LSTM ED	-2.30	0.11	2.41
CNN	-2.30	1.04	3.34
RNN - RNN ED	-2.30	1.65	3.95
RNN	-2.30	1.85	4.15
CNN - LSTM ED	-2.30	2.17	4.47
AR(1)	-2.30	1.48	3.78

Norwegian mainland GDP growth observation and predictions from year 2020
AR(1) is included as a benchmark for the performances in general

The absolute errors for our deep learning models in 2020 varies from 1.65 to 4.47, which is very poor compared with the other years. However, the LSTM, LSTM-LSTM ED and CNN at least performs better than the benchmark AR(1) model. The LSTM is interestingly the only model that predicts a negative growth by -0.65%. The year of 2020 was extraordinary due to COVID-19. The world economy in general experienced dramatic falls, resulting in the Norwegian mainland GDP to have a negative growth by $-2,3\%$. In a news article from February 2021, section chief for the national accounts in SSB, Pål Sletten stated: “This is the biggest decrease in the Norwegian mainland economy since the statistics for this measurement occurred in 1970, and it is probably the biggest decrease since World War II” (Hodne, 2021) [1].

As we have seen, the year 2020 makes it hard to evaluate our deep learning models. Hence, we now turn to the absolute errors of our models when excluding the year 2020. Table 4 shows the minimum- maximum- median- and mean absolute error for all the models in years between 2013 and 2019. Figure 20 visualizes the yearly observations of absolute errors between 2013 and 2019.

Table 4: Deep Learning Model Examination, Excluding 2020

	Rank	**Min**	**Max**	**Median**	**MAE**
CNN	1	0.03	0.76	0.13	0.27
LSTM - LSTM ED	3	0.02	1.42	0.38	0.46
CNN - LSTM ED	4	0.08	1.05	0.43	0.51
LSTM	5	0.14	1.60	0.52	0.58
RNN	6	0.04	2.22	0.53	0.77
RNN - RNN ED	7	0.00	1.76	0.60	0.78
AR(1)	2	0.02	0.97	0.25	0.40

Based on yearly observations between 2013 and 2019
All measures are based on the absolute error
Rank is determined by MAE, where a lower MAE results in a higher rank
AR(1) is included as a benchmark for the performances in general

We do not emphasize Table 4 and Figure 20 too heavily in our overall evaluation of the deep learning models, although, we think it is worth having a look because of the huge impact that the 2020 predictions have on the MAE. With that being said, Table 4 fully substantiates our impression of the CNN being the most consistent-

[1]The quote of Pål Sletten in the news article from February 2020 is a translation, thus, it is not his exact words.

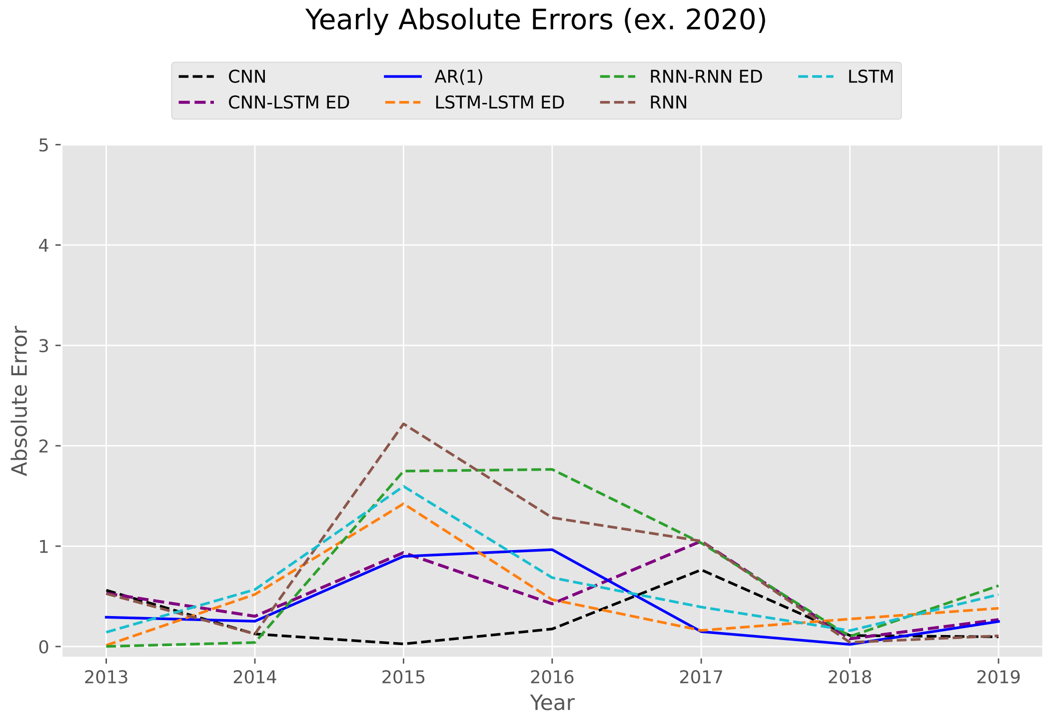

Figure 20: Deep Learning Models, Yearly Absolute Errors, Excluding 2020

and best performing model. It is superior with an MAE of 0.27 compared to the next best model at 0.46. Furthermore, this is reflected in the median- and maximum absolute errors. If we look at Figure 20, we can see that the CNN model predicts consistently well and generally outperforms the other models by the exception of 2017. Especially the prediction for 2015 is impressive, it is both accurate- and it stands out from the other models.

Based on the overall evaluation of our own deep learning models, it is clear that the CNN model is the one that performs the best on forecasting the Norwegian mainland GDP growth. Therefore, we pick the CNN model as our champion model. In the next section we compare this model to the predictions performed by the participants of Prognoseprisen.

6.2 Deep Learning Champion Model Evaluation

Now that we have picked our best performing deep learning model it is time to compare our predictions with the leading forecasting actors and major financial institutions that have competed in Prognoseprisen. As discussed in Section 3.2, to maximize our basis of comparison, we have chose to only include the participants that have competed in all years between 2013 and 2020. These institutions are: Danske Bank, Norges Bank, Finansdepartementet, Swedbank, DNB, Handelsbanken, SSB, Nordea, SEB and NHO. This approach excludes some of the participants including OECD, IMF and NAM. However, we believe this is the best way for comparison due to the fact that some years are easier to predict than others. In other words, some institutions may be unfairly penalized or rewarded for skipping a year.

As discussed in Section 3.3, we cannot rule out that we may have a data advantage on revised GDP growth, since our model utilizes the latest updated data available from April 2022. This data has perhaps been changed by varied degrees throughout the years. Furthermore, this data (revised macroeconomic variables) was obviously not accessible at the time for the institutions with whom we are comparing our predictions. Hence, we perform comparisons on both revised- and unrevised Norwegian mainland GDP growth.

Figure 21 shows our CNN model predictions, the AR(1) predictions and the four most accurate predictions from Prognoseprisen, relative to the observed revised- and unrevised Norwegian mainland GDP growth for all years between 2013 and 2020.

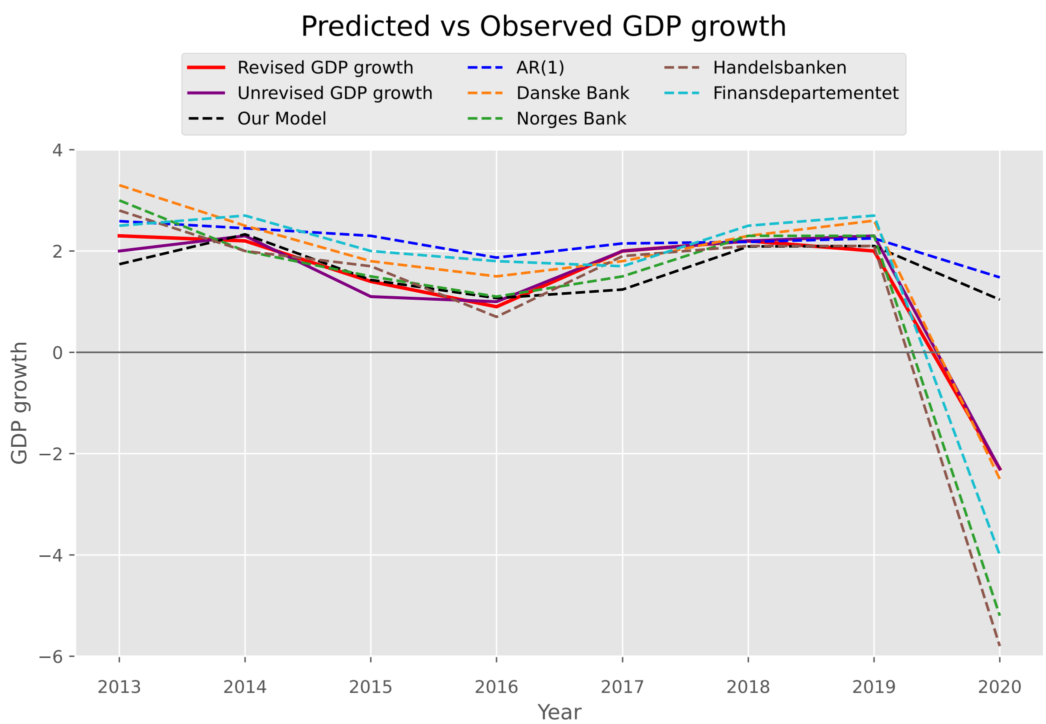

Figure 21: Model Evaluation, Predicted vs Observed

The revised- and unrevised GDP growth seems to be quite similar, although there are some years like 2013, 2015 and 2019 where we can see a slight difference. When we compare our model to the participants of Prognoseprisen, it appears that our model makes generally good predictions. The fact that these four predictions made by Danske Bank, Norges Bank, Handelsbanken and Finansdepartementet are the top four predictions in Prognoseprisen, further contributes to the positive first impression. With the exception of 2017 and 2020, we can see that our prediction is fairly accurate for each year. In 2017, our model does not perform well, and in 2020, our model merely outperforms the benchmark AR(1) model.

Let us now dive deeper into the errors and investigate further. Table 5 shows our model's minimum- maximum- median- and mean absolute error compared to AR(1) and the participants of Prognoseprisen on unrevised data with yearly observations between 2013 and 2020.

Table 5: Our Model vs Prognoseprisen, Unrevised GDP

	Rank	**Min**	**Max**	**Median**	**MAE**
Our Model	2	0.03	3.34	0.23	0.64
Danske Bank	1	0.10	1.30	0.25	0.44
Norges Bank	3	0.00	2.90	0.35	0.66
Finansdepartementet	3	0.30	1.70	0.45	0.66
Swedbank	5	0.00	3.70	0.25	0.70
DNB	6	0.10	3.60	0.30	0.73
Handelsbanken	7	0.10	3.50	0.30	0.74
SSB	8	0.10	3.20	0.35	0.80
Nordea	10	0.30	3.70	0.45	0.95
SEB	11	0.10	5.10	0.55	1.10
NHO	12	0.10	4.20	0.35	1.06
AR(1)	9	0.02	3.78	0.37	0.85

Based on observations between 2013 and 2020 on unrevised GDP growth
All measures are based on the absolute error
Rank is determined by MAE, where a lower MAE results in a higher rank
AR(1) is included as a benchmark for the performances in general

When compared to the other predictions on the unrevised observed Norwegian mainland GDP growth, our CNN model performs well. An MAE of 0.64 ranks our model second, only behind Danske Bank which outperforms the competition with an MAE of 0.44. Interestingly, we are outperforming all of the other institutions, including Danske Bank, when it comes to the median absolute error with the value of 0.23. This is something that we are going to investigate further. Next, we turn to the revised data. Table 6 shows our model's minimum- maximum- median- and mean absolute error compared to AR(1) and the participants of Prognoseprisen on revised data with yearly observations between 2013 and 2020.

Table 6: Our Model vs Prognoseprisen, Revised GDP

	Rank	**Min**	**Max**	**Median**	**MAE**
Our Model	4	0.03	3.34	0.15	0.65
Danske Bank	1	0.10	1.00	0.35	0.43
Norges Bank	2	0.10	2.90	0.25	0.63
Handelsbanken	2	0.10	3.50	0.20	0.63
Finansdepartementet	4	0.20	1.70	0.55	0.65
DNB	6	0.00	3.60	0.20	0.66
Swedbank	7	0.00	3.70	0.30	0.69
NHO	8	0.10	4.20	0.25	0.83
SSB	10	0.10	3.20	0.50	0.84
Nordea	11	0.20	3.70	0.60	0.91
SEB	12	0.20	5.10	0.60	1.09
AR(1)	8	0.02	3.78	0.27	0.83

Based on observations between 2013 and 2020 on revised GDP growth
All measures are based on the absolute error
Rank is determined by MAE, where a lower MAE results in a higher rank
AR(1) is included as a benchmark for the performances in general

In theory, since our model is built on revised data it should perform better on revised- than unrevised GDP growth. However, our model's MAE is almost the same on the revised data at 0.65 compared to the unrevised data at 0.64. Interestingly, on revised GDP growth our model is ranked at fourth place. In other words, we dropped two ranks on the unrevised- compared to the revised GDP growth. This means that some of the other predictions actually performs better on revised GDP growth. Hence, the predictions for some of the participants in Prognoseprisen are actually better then what they received credit for in the forecasting competition. Especially Handelsbanken makes a big jump in the rankings, going from seventh on unrevised data to second on revised data. If we turn to the median absolute error, we get the same impression as for the unrevised GDP growth. The median absolute error value in our model is 0.15, which is the best of all the models.

We now look at the difference in MAE for revised- and unrevised GDP growth for all the predictions. Table 7 shows our model's MAE compared to the AR(1) and the participants of Prognoseprisen on both revised- and unrevised data based on yearly observations between 2013 and 2020.

Table 7: Revised vs Unrevised

	Revised MAE	**Unrevised MAE**	**Difference**
Our Model	0.65	0.64	-0.01
Danske Bank	0.43	0.44	0.02
Norges Bank	0.63	0.66	0.04
Handelsbanken	0.63	0.74	0.12
Finansdepartementet	0.65	0.66	0.01
DNB	0.66	0.73	0.07
Swedbank	0.69	0.70	0.01
NHO	0.83	1.06	0.23
SSB	0.84	0.80	-0.04
Nordea	0.91	0.95	0.04
SEB	1.09	1.10	0.01
AR(1)	0.83	0.85	0.02

Based on observations between 2013 and 2020
AR(1) is included as a benchmark for the performances in general

For all the MAE's except for our model and SSB, we observe a varying degree of improvement for the unrevised- compared to the revised GDP growth. NHO and Handelsbanken have the predictions with the largest improvements by respectively 0.23 and 0.12. Despite the fact that our model uses more accurate data, the outcome of the forecasts do not reflect the data advantage we may have.

Recall from Section 6.1, where we observed that the year of 2020, arguably unfairly penalized the MAE for some of our models. Hence, to get a better overall evaluation of our model compared to the predictions of the participants in Prognoseprisen, we now look at the same revised- and unrevised GDP growth, where 2020 is excluded from the series of absolute errors. Table 8 shows the mean absolute error of our model compared to AR(1) and the participants of Prognoseprisen on both unrevised- and revised data with yearly observations between 2013 and 2019. Interestingly, when 2020 is excluded from the series of absolute errors on the unrevised GDP growth, our model achieves the best result out of all the predictions with an MAE value of 0.26. If we turn to the revised GDP growth, our model remains in fourth place, the same as when we include 2020 in Table 6, however, the gap between our model and rank 1 model is drastically decreased.

Table 8: Our Model vs Prognoseprisen, Excluding 2020

	UR Rank	UR MAE	R Rank	R MAE
Our Model	1	0.26	4	0.27
Swedbank	2	0.27	3	0.26
DNB	3	0.31	2	0.24
Norges Bank	4	0.34	5	0.30
Handelsbanken	4	0.34	1	0.21
SSB	7	0.46	9	0.50
Danske Bank	8	0.47	8	0.46
Finansdepartementet	9	0.51	9	0.50
SEB	10	0.53	11	0.51
Nordea	11	0.56	11	0.51
NHO	12	0.61	6	0.35
AR(1)	6	0.43	7	0.40

Based on observations between 2013 and 2019
Unrevised = UR, Revised = R
All measures are based on the absolute error
Rank is determined by MAE, where a lower MAE results in a higher rank
AR(1) is included as a benchmark for the performances in general

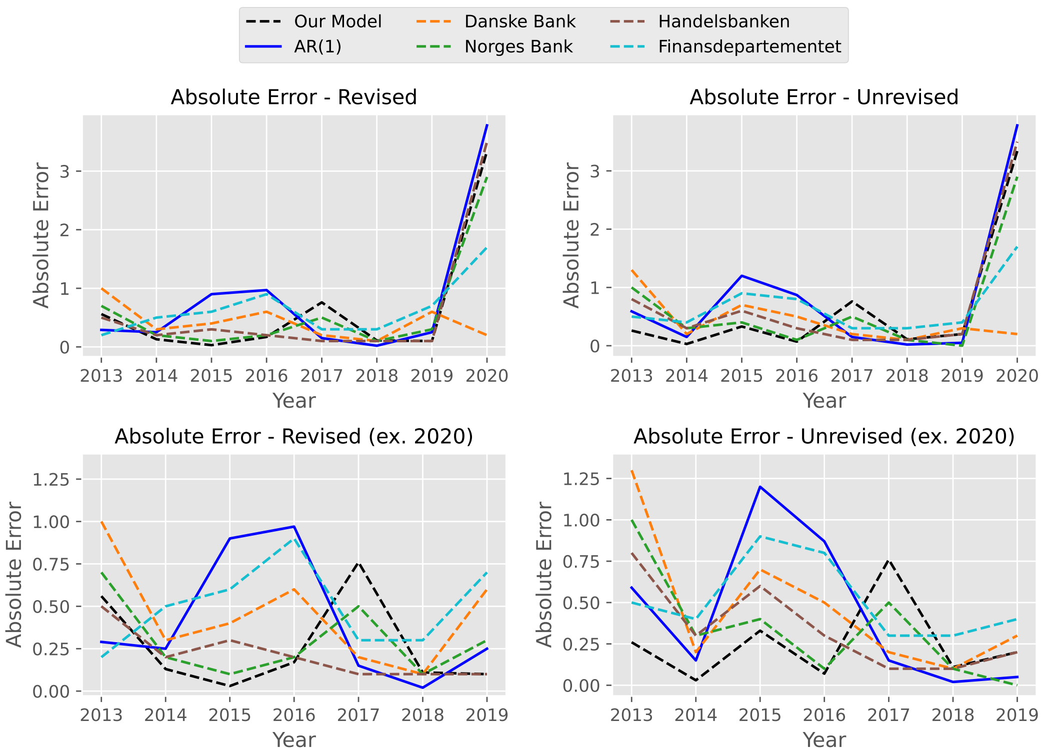

Figure 22: Model Evaluation, Revised vs Unrevised, Absolute Error

Finally, we look at the absolute errors across all evaluation scenarios for each year. Figure 22 visualizes four different variations of our model's overall assessment compared to the AR(1) model and the top four participants of Prognoseprisen.

Based on the MAE, we rank between first and fourth place in various parts throughout our evaluation. To summarize the results and analysis of our best deep learning CNN model, it is performing very well and would position at the top when compared to some of the leading forecasting actors and major financial institutions that have participated in Prognoseprisen.

7 Discussion

In this chapter, we discuss our main findings of our results and analysis, some limitations for our work, key strengths and weaknesses for our thesis and a proposal for further research.

7.1 Main Findings

Forecasting in general can be a very useful tool for individuals, businesses, and institutions making decisions in a variety of contexts. Accurate forecasts of the Norwegian mainland GDP growth can give major benefits in both small- and large scale negotiations and financial decisions. However, the future is very uncertain, which the COVID-19 pandemic is a prime example of. This uncertainty makes the future difficult to forecast. Thus, the need for tools that can improve predictions are in high demand. Today, there are numerous methods and tools available to assist in the prediction of various macroeconomic measures. We could purely guess, use our gut feeling, experience and judgment, time series regressions, machine learning algorithms, or maybe a combination of all mentioned. In this thesis we are investigating: *How well does deep learning algorithms perform on forecasting Norwegian mainland GDP growth?*. To our best knowledge, this specific question has not been researched very much in the past. A positive outcome on this question could possibly change the current work practice for a lot of major institutions that are striving to make predictions of such macroeconomic measures.

The results and analysis indicates that deep learning algorithms absolutely should be mentioned in the debate of forecasting the Norwegian mainland GDP growth. We do not possess any insights on the predictions performed by the participants of Prognoseprisen. However, we can confidently state that our deep learning CNN model is on par with- and even outperforms multiple forecasting actors and financial institutions with extensive resources, skill, and understanding within finance and economics. The year 2020 is the one that our model has the most trouble predicting. The Norwegian economy saw a significant decline that year. Our model, however, was unable to forecast a negative growth. Thus, we clearly see the advantage of the ability to forecast outlier years using other methods, such as experience, judgment, and discretion. With that being said, our model is outperforming the traditional time series forecasting model AR(1) by quite a large margin. This should be considered as a positive, as the AR model is often used as a benchmark and are already well implemented in the macroeconomic forecasting toolbox.

7.2 Limitations

Another important aspect that should be mentioned is the limitations we have in this thesis which we believe substantiates what we discussed in the section above. As last year Business Analytics master students we have limitations by a varying degree on especially, data, competence and time. The data quality and the amount of data are the most important factors for a successful deep learning forecast model. During our thesis we have had access to a Bloomberg terminal where we have collected most of our data. Except for this, we have collected data exclusively from open sources. Hence, we can arguably say that we have a data disadvantage compared to some of the participants in Prognoseprisen that may collect their own non-public data. Next, due to the fact that we are students with no experience working with forecasting we would like to point out our disadvantage in terms of competence. At last, our time frame for this thesis has been around one semester, which obviously sets a lot of limitations for us to be able to deliver the thesis by deadline.

7.3 Strengths & Weaknesses

We believe that we have a good basis for comparison of our model. We have done a good job obtaining what we think is close to the same starting point for prediction, making the comparison as fair as possible. This may be considered as a strength of this thesis. In addition, we would like mention that the evaluation of our model is solid and should be considered as a strength itself. Our evaluation involves comparison to what may be considered as some of the best macroeconomic forecasters in the country. We believe this is as a huge advantage and makes us comfortable drawing conclusion based on our deep learning model performance.

What may be considered as a weakness, is that we are not exactly sure what is behind the predictions we are comparing our model with. We do not know what the different predictions are based upon and what methods and tools that are used. As we touched upon in Section 3.2, other than the absolute errors we do not possess any insights about the predictions. Our research question revolves around how well does deep learning perform on predicting Norwegian mainland GDP growth, however, there is a chance that some of the predictions we have compared to likewise utilizes deep learning. Furthermore, the problem regarding revised data that we are discussing in Section 3.3, may be considered as a weakness for this study. This is a problem which is very hard to avoid. As data gets revised over time, the old versions of the data gets overwritten and thus, unobtainable.

7.4 Further Research

The main findings of this thesis suggests that we are investigating something very interesting and exciting. We think this topic has the potential to change the future work practice on making predictions of Norwegian mainland GDP growth. We have two proposals for further research that builds upon this thesis. First, feed the deep learning algorithms a lot more data. As we touched upon in Chapter 3, deep learning algorithms tends to increase in performance when more data is added (Brownlee, 2016). This is one of the key points that makes deep learning so exciting. For the further research it would be interesting to collect data such as google trends, newspaper documents and other text data in combination with macroeconomic- and financial data. Additionally, inventing more data based on the current variables by data augmentation. Our second proposal for further research is, predicting other types of macroeconomic measures such as, inflation, unemployment rate, housing price and interest rate. To investigate if the deep learning models could perform at the same level, or maybe even better on other macroeconomic measures would be a very exciting topic to research.

8 Conclusion

In this thesis we have presented results and analysis which reflects the huge potential of deep learning algorithms as a valuable tool of predicting the growth in the Norwegian economy. Our deep learning algorithms, uses 83 different variables observed at quarterly-, monthly- and daily basis to capture non-linear relationships and underlying patterns that explains the growth in the Norwegian mainland GDP. Our best performing algorithm, the CNN model is producing high performance predictions in comparison to some of the leading forecasting actors and major financial institutions that over the years have competed in Prognoseprisen. Our model has additionally proved that it generally outperforms the traditional time series regression model AR(1), which is already implemented as a common tool within predicting macroeconomic measures.

To answer our research question, *How well does deep learning algorithms perform on forecasting Norwegian mainland GDP growth?* we would conclude that based on our research, deep learning algorithms definitely can perform at the same level, and even better than predictions made by the biggest financial institutions in the country. However, in state of emergency such as the COVID-19 pandemic, our deep learning algorithms struggles and we see the value of methods such as experience, judgment and discretion in combination with models in such periods. If we look away from the year of 2020, in the period between 2013 and 2019, our CNN model produce some of the most accurate predictions.

References

Abadi, M., Agarwal, A., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Bhatt, A., Ongsakul, W., et al. (2022). Sliding window approach with first-order differencing for very short-term solar irradiance forecasting using deep learning models. *Sustainable Energy Technologies and Assessments*.

Brownlee, J. (2016). How To Improve Deep Learning Performance. `https://machinelearningmastery.com/improve-deep-learning-performance/`. [Accessed 03-Mars-2022].

Callen, T. (2020). Gross Domestic Product: An Economy's All. `https://www.imf.org/external/pubs/ft/fandd/basics/gdp.htm#:~:text=GDP%20measures%20the%20monetary%20value,the%20borders%20of%20a%20country`. [Accessed 03-Mars-2022].

Chollet, F. et al. (2015). Keras.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus).

Dette er Norge-redaksjonen (2021). Hva er egentlig BNP? `https://www.ssb.no/nasjonalregnskap-og-konjunkturer/nasjonalregnskap/statistikk/nasjonalregnskap/artikler/hva-er-egentlig-bnp`. [Accessed 03-Mars-2022].

Ganesh, S. (2021). What's The Role Of Weights And Bias In a Neural Network? `https://towardsdatascience.com/whats-the-role-of-weights-and-bias-in-a-neural-network-4cf7e9888a0f`. [Accessed 03-Mars-2022].

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Hansen, C. (2019). Activation functions explained. `https://mlfromscratch.com/activation-functions-explained/#/`.

Harris, C. R., Millman, K. J., et al. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (gelus).

Hodne, A. M. (2021). Stort fall i fastlands-BNP: – Sannsynligvis det største fallet siden krigen. https://www.tv2.no/a/11952084/. [Accessed 12-June-2022].

Holzbauer, L. (2019). A Quick Introduction to Vanilla Neural Networks. https://blog.insightdatascience.com/a-quick-introduction-to-vanilla-neural-networks-b0998c6216a1. [Accessed 03-Mars-2022].

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Hyndman, R. and Athanasopoulos, G. (2021). *Forecasting: principles and practice, (3rd ed)*. OTexts: Melbourne Australia. [Accessed 12-June-2022].

Iuhaniwal, V. (2019). Forward propagation in neural networks. https://blog.insightdatascience.com/a-quick-introduction-to-vanilla-neural-networks-b0998c6216a1. [Accessed 03-Mars-2022].

Jung, J.-K., Patnam, M., and Ter-Martirosyan, A. (2018). An algorithmic crystal ball: Forecasts-based on machine learning. *IMF Working Papers*, 18:1.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Koenecke, A. (2020). Applying deep neural networks to financial time series forecasting.

Kokic, P. (2001). Standard methods for imputing missing values in financial panel/time series data.

Kostadinov, S. (2019). Understanding Backpropagation Algorithm. https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd. [Accessed 03-Mars-2022].

Kwiatkowski, R. (2021). Gradient Descent Algorithm. https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd. [Accessed 03-Mars-2022].

pandas development team, T. (2020). pandas-dev/pandas: Pandas.

Shao, C. (2019). Properly Setting the Random Seed in ML Experiments. Not as Simple as You Might Imagine. https://opendatascience.com/properly-setting-the-random-seed-in-ml-experiments-not-as-simple-as-you-might-imagine/. [Accessed 12-June-2022].

Singla, S. (2022). Why Dropout is so effective in Deep Neural Network. https://towardsdatascience.com/introduction-to-dropout-to-regularize-deep-neural-network-8e9d6b1d4386. [Accessed 03-Mars-2022].

Smalter Hall, A. and Cook, T. R. (2017). Macroeconomic indicator forecasting with deep neural networks. *Federal Reserve Bank of Kansas City Working Paper*, (17-11).

SSB (2022a). Economic Trends. https://www.ssb.no/en/nasjonalregnskap-og-konjunkturer/konjunkturer/statistikk/konjunkturtendensene. [Accessed 17-June-2022].

SSB (2022b). Nasjonalregnskaper. https://www.ssb.no/nasjonalregnskap-og-konjunkturer/faktaside/norsk-okonomi. [Accessed 03-Mars-2022].

Szafranek, K. (2017). Bagged artificial neural networks in forecasting inflation.

Tkacz, G. (2001). Neural network forecasting of canadian gdp growth. *International Journal of Forecasting*, 17(1):57–69.

Tkacz, G. and Hu, S. (1999). Forecasting gdp growth using artificial neural networks. Technical report, Bank of Canada.

Wes McKinney (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

A Appendix - Data Overview

Table 9: Norwegian Data Overview

Feature name	Source	Freaquency
10-Year Bond Yield	Bloomberg	Daily
Bankruptcies	SSB	Monthly
Capacity Utilazation Rate	Bloomberg	Quarterly
Construction Cost Detached Houses	Bloomberg	Monthly
Construction Cost Multi Dwelling Houses	Bloomberg	Monthly
Construction Cost Residential Buildings	Bloomberg	Monthly
Consumer Price Index	SSB	Monthly
Employment Rate	SSB	Quarterly
Exchange rate EU/NOK	Bloomberg	Daily
Exchange rate GBP/NOK	Bloomberg	Daily
Exchange rate NOK/DKK	Bloomberg	Daily
Exchange rate SEK/NOK	Bloomberg	Daily
Exchange rate USD/NOK	Bloomberg	Daily
Exports	SSB	Quarterly
First-time Registered Passenger Cars	SSB	Monthly
Gross Domestic Product Growth, Mainland	SSB	Quarterly
Gross Domestic Product Growth, Mainland	SSB	Yearly
Gross Domestic Product Growth, Mainland	Prognoseprisen	Yearly
Gross Oil Investments	SSB	Quarterly
House Prices All Homes	Bloomberg	Quarterly
Imports	SSB	Quarterly
Industrial Production Manufacturing	Bloomberg	Monthly
Industrial Production Overall	Bloomberg	Monthly
Interest Rate	Norges Bank	Quarterly
Mortgage Average Lending Rate	Global Financial Data	Monthly
OBX Basic Materials GR	Bloomberg	Daily
OBX Consumer Discretionary GR	Bloomberg	Daily
OBX Consumer Staples GR	Bloomberg	Daily
OBX Energy GR	Bloomberg	Daily
OBX Financials GR	Bloomberg	Daily
OBX Health Care PR	Bloomberg	Daily
OBX Industrials GR	Bloomberg	Daily
OBX Technology GR	Bloomberg	Daily
OBX Telecommunications GR	Bloomberg	Daily
Oslo Børs Benchmark index	Bloomberg	Daily
Overnight Lending Rate	Bloomberg	Daily
Passenger Car Registration	OECD	Monthly
Private Consumption	SSB	Quarterly
Real Investments	SSB	Quarterly
The Expectation Barometer	Finance Norway	Quarterly
Total Demand	SSB	Quarterly
Unemployment Rate	NAV	Monthly
Unemployment Rate	SSB	Quarterly
Wage Growth	SSB	Quarterly

Table 10: International Data Overview

Country	Feature name	Source	Freaquency
International	Aluminium	Bloomberg	Daily
UK	Bank of England Bank Rate	Bloomberg	Daily
International	Brent Oil	Bloomberg	Daily
USA	Conference Board Leading Index	Bloomberg	Monthly
International	Copper	Bloomberg	Daily
International	CRB Commodity Index	Bloomberg	Daily
International	Crude Oil WTI	Bloomberg	Daily
Germany	DAX Index	Bloomberg	Daily
USA	Dow Jones Industrial Average	Bloomberg	Daily
USA	Dow Jones Transportation Average	Bloomberg	Daily
USA	Dow Jones Utility Average	Bloomberg	Daily
Eurozone	Euro STOXX 50	Bloomberg	Daily
Eurozone	European Commission Economic S.I.	Bloomberg	Monthly
USA	Federal Funds Rate	Bloomberg	Daily
UK	FTSE 100 Index	Bloomberg	Daily
International	Gold	Bloomberg	Daily
UK	Goverment Bonds Yield (10Y)	Bloomberg	Daily
Germany	Gross Domestic Product Growth	Bloomberg	Quarterly
Sweden	Gross Domestic Product Growth	Bloomberg	Quarterly
UK	Gross Domestic Product Growth	Bloomberg	Quarterly
USA	Gross Domestic Product Growth	Bloomberg	Quarterly
China	Gross Domestic Product Growth	Bloomberg	Quarterly
Denmark	Gross Domestic Product Growth	Bloomberg	Quarterly
UK	ICE LIBOR	Bloomberg	Daily
Eurozone	Industrial Production (ex. Construction)	Bloomberg	Monthly
Eurozone	Industrial Production Overall	Bloomberg	Monthly
USA	ISM Manufacturing PMI	Bloomberg	Monthly
International	Lead	Bloomberg	Daily
World	MSCI World Index	Bloomberg	Daily
International	Nickel	Bloomberg	Daily
Japan	NIKKEI 225 Index	Bloomberg	Daily
Sweden	OMX Index	Bloomberg	Daily
Sweden	PMI Manufacturing Swedbank	Bloomberg	Monthly
Portugal	PSI-20 Index	Bloomberg	Daily
USA	S&P 500 Index	Bloomberg	Daily
International	S&P Commodity Index	Bloomberg	Daily
China	Shanghai SE Composite Index	Bloomberg	Daily
International	Silver	Bloomberg	Daily
USA	VIX Index	Bloomberg	Daily
USA	Yield Spread (10Y-2Y)	Bloomberg	Daily
International	Zinc	Bloomberg	Daily

B Appendix - Yearly Predictions

B.1 Revised

Table 11: Our Model vs Prognoseprisen, Revised, 2013

	Rank	Observed	Prediction	Absolute Error
Our Model	9	2.30	1.74	0.56
DNB	1	2.30	2.40	0.10
Finansdepartementet	2	2.30	2.50	0.20
NHO	3	2.30	2.50	0.20
NAM	5	2.30	2.70	0.40
Swedbank	6	2.30	2.80	0.50
Handelsbanken	7	2.30	2.80	0.50
CAMP	8	2.30	2.85	0.55
SSB	10	2.30	2.90	0.60
Norges Bank	11	2.30	3.00	0.70
Nordea	12	2.30	3.00	0.70
SEB	13	2.30	3.10	0.80
Danske Bank	14	2.30	3.30	1.00
OECD	15	2.30	3.60	1.30
IMF	16	0.60	2.35	1.75
AR(1)	4	2.30	2.59	0.29

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2013
Rank is determined by absolute error

Table 12: Deep Learning Models, Revised, 2013

	Rank	Observed	Prediction	Absolute Error
RNN - RNN ED	1	2.30	2.30	0.00
LSTM - LSTM ED	2	2.30	2.29	0.01
LSTM	3	2.30	2.44	0.14
RNN	5	2.30	1.77	0.53
CNN - LSTM ED	6	2.30	2.83	0.53
CNN	7	2.30	1.74	0.56
AR(1)	4	2.30	2.59	0.29

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2013
Rank is determined by absolute error

Table 13: Our Model vs Prognoseprisen, Revised, 2014

	Rank	Observed	Prediction	Absolute Error
Our Model	4	2.20	2.33	0.13
IMF	1	2.20	2.28	0.08
Swedbank	2	2.20	2.30	0.10
SSB	2	2.20	2.10	0.10
CAMP	5	2.20	2.38	0.18
SEB	6	2.20	2.40	0.20
DNB	6	2.20	2.00	0.20
Handelsbanken	6	2.20	2.00	0.20
Norges Bank	6	2.20	2.00	0.20
NHO	11	2.20	2.50	0.30
Danske Bank	11	2.20	2.50	0.30
LO	11	2.20	2.50	0.30
NAM	14	2.20	2.55	0.35
EU	15	2.20	2.60	0.40
Finansdepartementet	16	2.20	2.70	0.50
Nordea	17	2.20	1.30	0.90
AR(1)	10	2.20	2.45	0.25

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2014
Rank is determined by absolute error

Table 14: Deep Learning Models, Revised, 2014

	Rank	Observed	Prediction	Absolute Error
RNN - RNN ED	1	2.20	2.16	0.04
RNN	2	2.20	2.07	0.13
CNN	2	2.20	2.33	0.13
CNN - LSTM ED	5	2.20	2.50	0.30
LSTM - LSTM ED	6	2.20	2.72	0.52
LSTM	7	2.20	2.77	0.57
AR(1)	4	2.20	2.45	0.25

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2014
Rank is determined by absolute error

Table 15: Our Model vs Prognoseprisen, Revised, 2015

	Rank	Observed	Prediction	Absolute Error
Our Model	2	1.40	1.43	0.03
Swedbank	1	1.40	1.40	0.00
Norges Bank	3	1.40	1.50	0.10
NHO	4	1.40	1.25	0.15
DNB	5	1.40	1.20	0.20
Nordea	6	1.40	1.60	0.20
IMF	7	1.60	1.86	0.26
Handelsbanken	8	1.40	1.70	0.30
SSB	9	1.40	1.00	0.40
NAM	9	1.40	1.80	0.40
Danske Bank	9	1.40	1.80	0.40
EU	12	1.60	2.2	0.60
Finansdepartementet	12	1.40	2.00	0.60
SEB	14	1.40	2.10	0.70
LO	16	1.40	2.50	1.10
OECD	17	1.40	2.90	1.50
AR(1)	15	1.40	2.30	0.90

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2015
Rank is determined by absolute error

Table 16: Deep Learning Models, Revised, 2015

	Rank	Observed	Prediction	Absolute Error
CNN	1	1.40	1.43	0.03
CNN - LSTM ED	3	1.40	2.34	0.94
LSTM - LSTM ED	4	1.40	2.82	1.42
LSTM	5	1.40	3.00	1.60
RNN - RNN ED	6	1.40	3.15	1.75
RNN	7	1.40	3.62	2.22
AR(1)	2	1.40	2.30	0.90

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2015
Rank is determined by absolute error

Table 17: Our Model vs Prognoseprisen, Revised, 2016

	Rank	Observed	Prediction	Absolute Error
Our Model	4	0.90	1.07	0.17
EU	1	1.10	1.10	0.00
Swedbank	2	0.90	1.00	0.10
IMF	3	1.10	1.26	0.16
Norges Bank	5	0.90	1.10	0.20
Handelsbanken	5	0.90	0.70	0.20
DNB	7	0.90	1.20	0.30
CAMP	8	0.90	0.50	0.40
Nordea	9	0.90	1.40	0.50
Danske Bank	10	0.90	1.50	0.60
LO	10	0.90	1.50	0.60
Finansdepartementet	12	0.90	1.80	0.90
SEB	12	0.90	1.80	0.90
NAM	15	0.90	1.99	1.09
NHO	16	0.90	2.00	1.10
SSB	16	0.90	2.00	1.10
AR(1)	14	0.90	1.87	0.97

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2016
Rank is determined by absolute error

Table 18: Deep Learning Models, Revised, 2016

	Rank	Observed	Prediction	Absolute Error
CNN	1	0.90	1.07	0.17
CNN - LSTM	2	0.90	1.33	0.43
LSTM - LSTM	3	0.90	1.37	0.47
LSTM	4	0.90	1.59	0.69
RNN	6	0.90	2.18	1.28
RNN - RNN	7	0.90	2.66	1.76
AR(1)	5	0.90	1.87	0.97

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2016
Rank is determined by absolute error

Table 19: Our Model vs Prognoseprisen, Revised, 2017

	Rank	Observed	Prediction	Absolute Error
Our Model	15	2.00	1.24	0.76
NHO	1	2.00	2.10	0.10
Handelsbanken	1	2.00	1.90	0.10
Danske Bank	4	2.00	1.80	0.20
OECD	4	2.00	2.20	0.20
Finansdepartementet	6	2.00	1.70	0.30
SSB	6	2.00	1.70	0.30
Nordea	6	2.00	1.70	0.30
SEB	6	2.00	1.70	0.30
EU	10	2.00	1.6	0.40
Swedbank	11	2.00	1.50	0.50
CAMP	11	2.00	1.50	0.50
Norges Bank	11	2.00	1.50	0.50
DNB	14	2.00	1.30	0.70
IMF	16	2.00	1.20	0.80
LO	17	2.00	1.00	1.00
AR(1)	3	2.00	2.15	0.15

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2017
Rank is determined by absolute error

Table 20: Deep Learning Model, Revised, 2017

	Rank	Observed	Prediction	Absolute Error
LSTM - LSTM ED	2	2.00	2.16	0.16
LSTM	3	2.00	2.39	0.39
CNN	4	2.00	1.24	0.76
RNN - RNN ED	5	2.00	3.03	1.03
CNN - LSTM ED	6	2.00	0.95	1.05
RNN	7	2.00	3.05	1.05
AR(1)	1	2.00	2.15	0.15

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2017
Rank is determined by absolute error

Table 21: Our Model vs Prognoseprisen, Revised, 2018

	Rank	Observed	Prediction	Absolute Error
Our Model	9	2.20	2.09	0.11
LO	2	2.20	2.25	0.05
OECD	3	2.20	2.30	0.10
Norges Bank	3	2.20	2.30	0.10
Danske Bank	3	2.20	2.30	0.10
NHO	3	2.20	2.10	0.10
Swedbank	3	2.20	2.10	0.10
Handelsbanken	3	2.20	2.10	0.10
CAMP	10	2.20	2.40	0.20
SEB	10	2.20	2.40	0.20
DNB	10	2.20	2.00	0.20
Finansdepartementet	13	2.20	2.50	0.30
SSB	13	2.20	2.50	0.30
Nordea	13	2.20	2.50	0.30
EU	13	1.30	1.6	0.30
IMF	17	1.30	1.62	0.32
AR(1)	1	2.20	2.18	0.02

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2018
Rank is determined by absolute error

Table 22: Deep Learning Models, Revised, 2018

	Rank	Observed	Prediction	Absolute Error
RNN	2	2.20	2.16	0.04
CNN - LSTM ED	3	2.20	2.12	0.08
RNN - RNN ED	4	2.20	2.10	0.10
CNN	5	2.20	2.09	0.11
LSTM	6	2.20	2.04	0.16
LSTM - LSTM ED	7	2.20	1.93	0.27
AR(1)	1	2.20	2.18	0.02

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2018
Rank is determined by absolute error

Table 23: Our Model vs Prognoseprisen, Revised, 2019

	Rank	Observed	Prediction	Absolute Error
Our Model	2	2.00	2.10	0.10
DNB	1	2.00	2.00	0.00
Handelsbanken	2	2.00	2.10	0.10
OECD	4	2.00	2.20	0.20
LO	4	2.00	2.20	0.20
Norges Bank	7	2.00	2.30	0.30
CAMP	8	2.00	2.40	0.40
NHO	9	2.00	2.50	0.50
Swedbank	10	2.00	2.50	0.50
SEB	10	2.00	2.50	0.50
Danske Bank	12	2.00	2.60	0.60
Finansdepartementet	13	2.00	2.70	0.70
SSB	13	2.00	2.70	0.70
Nordea	13	2.00	2.70	0.70
EU	16	0.90	1.9	1.00
IMF	17	0.90	2.06	1.16
AR(1)	6	2.00	2.25	0.25

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2019
Rank is determined by absolute error

Table 24: Deep Learning Models, Revised, 2019

	Rank	Observed	Prediction	Absolute Error
CNN	1	2.00	2.10	0.10
RNN	2	2.00	1.89	0.11
CNN - LSTM ED	4	2.00	2.27	0.27
LSTM - LSTM ED	5	2.00	2.38	0.38
LSTM	6	2.00	2.52	0.52
RNN - RNN ED	7	2.00	2.61	0.61
AR(1)	3	2.00	2.25	0.25

Observed Norwegian mainland GDP growth is based on revised data
Norwegian mainland GDP growth predictions of year 2019
Rank is determined by absolute error

Table 25: Our Model vs Prognoseprisen, Revised, 2020

	Rank	Observed	Prediction	Absolute Error
Our Model	6	-2.30	1.04	3.34
Danske Bank	1	-2.30	-2.50	0.20
Finansdepartementet	2	-2.30	-4.00	1.70
SØA	3	-2.30	-5.10	2.80
Norges Bank	4	-2.30	-5.20	2.90
SSB	5	-2.30	-5.50	3.20
Handelsbanken	7	-2.30	-5.80	3.50
DNB	8	-2.30	-5.90	3.60
Swedbank	9	-2.30	-6.00	3.70
Nordea	9	-2.30	-6.00	3.70
NHO	12	-2.30	-6.50	4.20
LO	12	-2.30	-6.50	4.20
EU	14	-0.70	-5.50	4.80
SEB	15	-2.30	-7.40	5.10
AR(1)	11	-2.30	1.48	3.78

Observed Norwegian mainland GDP growth is based on revised data

Norwegian mainland GDP growth predictions of year 2020

Rank is determined by absolute error

Table 26: Deep Learning Models, Revised, 2020

	Rank	Observed	Prediction	Absolute Error
LSTM	1	-2.30	-0.65	1.65
LSTM - LSTM ED	2	-2.30	0.11	2.41
CNN	3	-2.30	1.04	3.34
RNN - RNN ED	5	-2.30	1.65	3.95
RNN	6	-2.30	1.85	4.15
CNN - LSTM ED	7	-2.30	2.17	4.47
AR(1)	4	-2.30	1.48	3.78

Observed Norwegian mainland GDP growth is based on revised data

Norwegian mainland GDP growth predictions of year 2020

Rank is determined by absolute error

B.2 Unrevised

Table 27: Our Model vs Prognoseprisen, Unrevised, 2013

	Rank	Observed	Prediction	Absolute Error
Our Model	1	2.00	1.74	0.26
DNB	2	2.00	2.40	0.40
Finansdepartementet	3	2.00	2.50	0.50
NHO	3	2.00	2.50	0.50
NAM	6	2.00	2.70	0.70
Swedbank	7	2.00	2.80	0.80
Handelsbanken	7	2.00	2.80	0.80
CAMP	9	2.00	2.85	0.85
SSB	10	2.00	2.90	0.90
Norges Bank	11	2.00	3.00	1.00
Nordea	11	2.00	3.00	1.00
SEB	13	2.00	3.10	1.10
Danske Bank	14	2.00	3.30	1.30
OECD	15	2.00	3.60	1.60
IMF	16	0.60	2.35	1.75
AR(1)	5	2.00	2.59	0.59

Observed Norwegian mainland GDP growth is based on unrevised data
Norwegian mainland GDP growth predictions of year 2013
Rank is determined by absolute error

Table 28: Our Model vs Prognoseprisen, Unrevised, 2014

	Rank	Observed	Prediction	Absolute Error
Our Model	2	2.30	2.33	0.03
Swedbank	1	2.30	2.30	0.00
IMF	3	2.20	2.28	0.08
CAMP	3	2.30	2.38	0.08
SEB	5	2.30	2.40	0.10
SSB	7	2.30	2.10	0.20
NHO	7	2.30	2.50	0.20
Danske Bank	7	2.30	2.50	0.20
LO	7	2.30	2.50	0.20
NAM	11	2.30	2.55	0.25
DNB	12	2.30	2.00	0.30
Handelsbanken	12	2.30	2.00	0.30
Norges Bank	12	2.30	2.00	0.30
EU	15	2.20	2.60	0.40
Finansdepartementet	15	2.30	2.70	0.40
Nordea	17	2.30	1.30	1.00
AR(1)	6	2.30	2.45	0.15

Observed Norwegian mainland GDP growth is based on unrevised data
Norwegian mainland GDP growth predictions of year 2014
Rank is determined by absolute error

Table 29: Our Model vs Prognoseprisen, Unrevised, 2015

	Rank	Observed	Prediction	Absolute Error
Our Model	6	1.10	1.43	0.33
DNB	1	1.10	1.20	0.10
SSB	1	1.10	1.00	0.10
NHO	3	1.10	1.25	0.15
IMF	4	1.60	1.86	0.26
Swedbank	5	1.10	1.40	0.30
Norges Bank	7	1.10	1.50	0.40
Nordea	8	1.10	1.60	0.50
Handelsbanken	9	1.10	1.70	0.60
EU	9	1.60	2.20	0.60
NAM	11	1.10	1.80	0.70
Danske Bank	11	1.10	1.80	0.70
Finansdepartementet	13	1.10	2.00	0.90
SEB	14	1.10	2.10	1.00
LO	16	1.10	2.50	1.40
OECD	17	1.10	2.90	1.80
AR(1)	15	1.10	2.30	1.20

Observed Norwegian mainland GDP growth is based on unrevised data
Norwegian mainland GDP growth predictions of year 2015
Rank is determined by absolute error

Table 30: Our Model vs Prognoseprisen, Unrevised, 2016

	Rank	**Observed**	**Prediction**	**Absolute Error**
Our Model	3	1.00	1.07	0.07
Swedbank	1	1.00	1.00	0.00
EU	1	1.10	1.10	0.00
Norges Bank	4	1.00	1.10	0.10
IMF	5	1.10	1.26	0.16
DNB	6	1.00	1.20	0.20
Handelsbanken	7	1.00	0.70	0.30
Nordea	8	1.00	1.40	0.40
CAMP	9	1.00	0.50	0.50
Danske Bank	9	1.00	1.50	0.50
LO	9	1.00	1.50	0.50
Finansdepartementet	12	1.00	1.80	0.80
SEB	12	1.00	1.80	0.80
NAM	15	1.00	1.99	0.99
NHO	16	1.00	2.00	1.00
SSB	16	1.00	2.00	1.00
AR(1)	14	1.00	1.87	0.87

Observed Norwegian mainland GDP growth is based on unrevised data
Norwegian mainland GDP growth predictions of year 2016
Rank is determined by absolute error

Table 31: Our Model vs Prognoseprisen, Unrevised, 2017

	Rank	**Observed**	**Prediction**	**Absolute Error**
Our Model	15	2.00	1.24	0.76
NHO	1	2.00	2.10	0.10
Handelsbanken	1	2.00	1.90	0.10
Danske Bank	4	2.00	1.80	0.20
OECD	4	2.00	2.20	0.20
Finansdepartementet	6	2.00	1.70	0.30
SSB	6	2.00	1.70	0.30
Nordea	6	2.00	1.70	0.30
SEB	6	2.00	1.70	0.30
EU	10	2.00	1.60	0.40
Swedbank	11	2.00	1.50	0.50
CAMP	11	2.00	1.50	0.50
Norges Bank	11	2.00	1.50	0.50
DNB	14	2.00	1.30	0.70
IMF	16	2.00	1.20	0.80
LO	17	2.00	1.00	1.00
AR(1)	3	2.00	2.15	0.15

Observed Norwegian mainland GDP growth is based on unrevised data
Norwegian mainland GDP growth predictions of year 2017
Rank is determined by absolute error

Table 32: Our Model vs Prognoseprisen, Unrevised, 2018

	Rank	**Observed**	**Prediction**	**Absolute Error**
Our Model	9	2.20	2.09	0.11
LO	1	2.20	2.25	0.05
OECD	3	2.20	2.30	0.10
Norges Bank	3	2.20	2.30	0.10
Danske Bank	3	2.20	2.30	0.10
NHO	3	2.20	2.10	0.10
Swedbank	3	2.20	2.10	0.10
Handelsbanken	3	2.20	2.10	0.10
CAMP	10	2.20	2.40	0.20
SEB	10	2.20	2.40	0.20
DNB	10	2.20	2.00	0.20
Finansdepartementet	13	2.20	2.50	0.30
SSB	13	2.20	2.50	0.30
Nordea	13	2.20	2.50	0.30
EU	13	1.30	1.60	0.30
IMF	17	1.30	1.62	0.32
AR(1)	2	2.20	2.18	0.02

Observed Norwegian mainland GDP growth is based on unrevised data

Norwegian mainland GDP growth predictions of year 2018

Rank is determined by absolute error

Table 33: Our Model vs Prognoseprisen, Unrevised, 2019

	Rank	Observed	Prediction	Absolute Error
Our Model	6	2.30	2.10	0.20
Norges Bank	1	2.30	2.30	0.00
OECD	3	2.30	2.20	0.10
LO	3	2.30	2.20	0.10
CAMP	3	2.30	2.40	0.10
Handelsbanken	6	2.30	2.10	0.20
NHO	6	2.30	2.50	0.20
Swedbank	6	2.30	2.50	0.20
SEB	6	2.30	2.50	0.20
DNB	11	2.30	2.00	0.30
Danske Bank	11	2.30	2.60	0.30
Finansdepartementet	13	2.30	2.70	0.40
SSB	13	2.30	2.70	0.40
Nordea	13	2.30	2.70	0.40
EU	16	0.90	1.90	1.00
IMF	17	0.90	2.06	1.16
AR(1)	2	2.30	2.25	0.05

Observed Norwegian mainland GDP growth is based on unrevised data

Norwegian mainland GDP growth predictions of year 2019

Rank is determined by absolute error

Table 34: Our Model vs Prognoseprisen, Unrevised, 2020

	Rank	**Observed**	**Prediction**	**Absolute Error**
Our Model	6	-2.30	1.04	3.34
Danske Bank	1	-2.30	-2.50	0.20
Finansdepartementet	2	-2.30	-4.00	1.70
SØA	3	-2.30	-5.10	2.80
Norges Bank	4	-2.30	-5.20	2.90
SSB	5	-2.30	-5.50	3.20
Handelsbanken	7	-2.30	-5.80	3.50
DNB	8	-2.30	-5.90	3.60
Swedbank	9	-2.30	-6.00	3.70
Nordea	9	-2.30	-6.00	3.70
NHO	12	-2.30	-6.50	4.20
LO	12	-2.30	-6.50	4.20
EU	14	-0.70	-5.50	4.80
SEB	15	-2.30	-7.40	5.10
AR(1)	11	-2.30	1.48	3.78

Observed Norwegian mainland GDP growth is based on unrevised data
Norwegian mainland GDP growth predictions of year 2020
Rank is determined by absolute error

C Appendix - Samfunnsøkonomenes Prognosepris

Table 35: Samfunnsøkonomens Prognosepris, Participants

Participant	Link
CAMP	https://www.tinyurl.com/5n8v5m8a
Danske Bank	https://www.danskebank.no/
DNB	https://www.dnb.no/en
Finansdepartementet	https://www.regjeringen.no/en/dep/fin/id216/
Handelsbanken	https://www.handelsbanken.no/en/
IMF	https://www.imf.org/en/Home
LO	https://www.lo.no/language/english/
NAM	https://www.normetrics.no/
NHO	https://www.nho.no/en/
Nordea	https://www.nordea.no/
Norges Bank	https://www.norges-bank.no/en/
OECD	https://www.oecd.org/
SEB	https://www.seb.no/
SSB	https://www.ssb.no/en
Swedbank	https://www.swedbank.com/