Department of Computer Science

Submitted in part fulfilment for the degree of MEng.

Macroeconomic Forecasting Using Neural Networks

Andrew Skea

Supervisor: Dimitar Kazakov

This project was conducted in accordance with the ethics guidelines set out by the University of York. The work is meant solely as academic research and any use by individuals or groups either commercially or personally, outside academia, should use it at their own risk. There are no legal issues in this project as all data has been downloaded from publicly available sources and no other personal data has been gathered, used or stored during this project.

Acknowledgements

Thank you to my family, friends, the University of York and my project supervisor, Dimitar Kazakov, who have all helped me during the completion of this project.

Contents

List of Figures

List of Tables

Executive Summary

It is impossible to accurately forecast economies or financial markets. Despite this, it still remains one of the most important challenges of today, with huge amounts of money and time spent in order to plan for the future or invest intelligently. The purpose of this paper is to investigate whether a machine learning based approach can improve on current algorithms.

Artificial neural networks have been around for the last few decades but have recently grown in importance and efficiency. Variations of these networks have been produced for certain goals such as categorisation, function approximation and time series forecasting. This project looks at using long-short term memory (LSTM) neural networks to predict certain British macroeconomic indicators. It then compares these results to models that have been used previously, such as auto-regressive integrated moving average (ARIMA) and equation discovery (ED).

The economy is defined by several macroeconomic indicators: gross domestic product, inflation rate and interest rate. These are considered to be lagging parameters as they only fluctuate following a change to industry. These precedent changes are usually recorded within leading indicators such as factory output. Logically, it is therefore beneficial to use these leading indicators to predict the lagging ones. Hence, the other objective of this paper is to model a combination of leading and lagging factors to see if the forecast is more accurate than solely using lagging ones.

According to the statistical tests performed, the LSTM models all performed significantly different. It was also apparent that adding indicators as inputs generated more noise for the network, leading to lower average root mean squared error (RMSE) results. However, these conclusions rely solely on this error function, which has been found to be largely misleading. The graphs showed forecasts that were either similar to moving averages or demonstrated large inaccuracies, despite low RMSE values. It is therefore advised that more research be carried out to improve the architecture, change inputs and most importantly change the evaluation method to one that provides more insight into the strength of the prediction. This may involve an expert opinion or modelling it as investment opportunities. Despite results that have proved the hypotheses incorrect, this project has been successful in achieving its objectives of modelling macroeconomic indicators and comparing them with alternative models.

To create this paper, the data used was downloaded from publicly available sources and the software was created using public, open-source libraries. The results are intended for academic purposes only and any use by individuals or entities, outside academia (such as for trading), should use them at their own risk. There is the ethical stand-point that trading involves people or firms gaining at another person's expense and therefore should not be allowed [1]. However this project aims to predict certain factors within the economy, with the intention of exploring technical models, and not provide any financial gain. This process, if ever used in industry, would primarily be used for business planning rather than trading.

1 Introduction

Predicting the economy has remained an extremely arduous task for businesses and investors alike. Financial firms use it for smarter investments, businesses need it for resource planning and the government uses it for planning the next finance policies. No matter the industry or the scale, it is a challenge of upmost importance.

The difficulty in forecasting comes from a combination of the unknown and the noise within these markets. Unexpected news and the vast amount of variables affecting make it hard to model computationally, even with state-of-the-art technology. However, there has been a recent and drastic increase in processing power. This, along with the development of new machine learning (ML) techniques, has made the task more achievable making it even more of a priority for entities that need this information. An interesting branch within machine learning is artificial neural networks. It has become a promising new way of modelling, categorising and estimating using techniques based on the human brain. Although impossible years ago, the power of processors now enables these networks to achieve impressive results in many fields.

This project aims to use these neural networks to forecast macroeconomic indicators including gross domestic product (GDP), interest rate and inflation. It begins by describing aspects of the economy and how certain factors relate. Secondly, it explains the function and the different types of neural networks that exist. It then moves on to implementing one of these networks with the goal of estimating the indicators' values years into the future. Lastly, there is an in-depth evaluation of the results and multiple ideas on how to extend this project with future work.

2 Literature Review

The literature review will explore the topics required to implement this project. The main areas are financial forecasting, economics, time series data and artificial neural networks.

2.1 Financial Forecasting

Financial forecasting is primarily used in two coinciding areas: planning and investing. A company may require an economic grounding to plan resources, projects and manage finances. Investment firms use it in their core business activities such as trading exchange rates or stocks and the country itself uses their past economic figures in order to publish their new monetary policy.

With the viewpoint of investing, the term 'financial forecasting' is the identification of trends in historical data and projecting those trends in order to provide information about what the financial status of an entity is likely to be at some point in the future [2], [3]. Each of these entities are affected by many variables, some of which are unexpected. A recent example of this was the United Kingdom leaving the European Union, which lead to a very large decrease in the value of the British pound. Most events such as this are impossible to predict making accurate forecasting impossible. Stemming from this, there exists the theory of the 'efficient market hypothesis' where stocks follow a 'random walk' meaning they reflect all current information and predicting values based on technical or fundamental analysis is impossible [4]. It is broken down into three categories. First is the weak-hypothesis, where it is thought that a stock's price can be derived from past prices, trading volume and short interest. Next, the semi-strong hypothesis also takes into account public information such as fundamental data of the stock, balance sheets and earnings forecasts. Lastly, the strong-hypothesis builds on the previous two and says that the price includes insider-only information too. Contrary to what Atiya suggests in his paper [3], there are very few who believe in the original efficient market hypothesis. This is due to investors such as Warren Buffet [5] along with many of the investment firms trading successfully over very long periods of time. Consequently, despite these theories, there remains a serious commitment to forecasting markets. Recently, studies suggest that these forecasts are primarily done through computers without any human interaction. The technology and algorithms powering these decisions are also partially based on neural networks [6], which are discussed in section 2.4.

Financial forecasting is often split into two sections, fundamental analysis and technical analysis. Fundamental analysis is the process of determining an assets value based on its economic and financial aspects, such as the price/earnings-to-growth ratio [7]. This could therefore include macroeconomic and microeconomic factors [8] as well. Technical analysis focuses on price movements, volume and other trading indicators [9]. For forecasting to be successful, a trader must use all the data they can access or else they may get unexpected results. The advantage of bringing in powerful algorithms and modelling techniques is that they can assemble all these variables and make informed decisions that would take a human investor a lot longer to do.

2.2 Economics

Economics, for nearly 100 years, has been split into two overlapping sections: macroeconomics and microeconomics. In essence, microeconomics takes a 'bottom to top' approach to studying a country's economy, focusing largely on businesses and individuals. Macroeconomics uses a 'top to bottom' approach by focusing on industries as a whole. When studying a countries wealth or living standards, it is primarily done by analysing their macroeconomic factors [10].

2.2.1 Macroeconomics

Macroeconomics is the study of economy-wide phenomena, including inflation, unemployment, and economic growth [11]. These factors are set or influenced by monetary and fiscal [12] policies released by the central bank [13] of the that country; the Bank of England (BoE) [14] being the central bank for the UK. When publishing the monetary policy, the bank communicates information about some key drivers in the economy. These drivers aim to control aspects such as inflation, consumption, growth and liquidity. These are achieved by modifying the interest rate, buying or selling sovereign bonds, regulating foreign exchange rates, and changing the amount of money banks are required to maintain as reserves [15]. The multitude of factors that are measured within the economy can be split into two sections: leading and lagging indicators.

2.2.1.1 Leading and lagging indicators

Leading indicators are ones that change direction ahead certain events happening [16]. Investors use this as a signal for entering a trade whilst the government use it to implement financial policies to keep the economy stable.

Lagging factors on the other hand are ones that change only after an event [16]. Although this cannot be used as a signal, it can be used to confirm the start of a new trend, such as the economy beginning to grow again. The table below (Fig 2.1) displays some examples of each type of economic indicator.

Lagging Factors	**Leading Factors**
Unemployment	Manufacturing Activity
Gross Domestic Product	New Business Start-ups
CPI and RPI Inflation	Bond Yields
Interest Rate	Building Permits

Table 2.1: Leading and Lagging macroeconomic factors [17]

The leading factors can indicate the direction an economy might go in. Building permits are only issued when people or businesses have the cash to spend on properties. This along with an increase in the manufacturing activity suggests companies are doing well therefore advancing the economy. Additionally, start-ups are only funded when investors have extra money to invest. These are all indicators that change quickly and ahead of the lagging factors. The lagging factors aggregate the leading factors into high-level figures. A boost in the economy may decrease the unemployment rate by providing extra jobs the following year. GDP will only increase following an increase in manufacturing activity.

2.2.1.2 Inflation

According to the UK Office for National Statistics (ONS), inflation is the rate of increase in prices for goods and services [18]. It is measured by the retail price index (RPI) and the consumer price index (CPI) for each country and plays a crucial role in monetary policy as it is used for setting the interest rate. The inflation rate can be calculated by the change in consumer price index for the period of a year or a month [19]. The following equation calculates the inflation rate for the year, using monthly CPI values:

$$IR_{y,t} = \frac{CPI_t}{CPI_{t-12}} \tag{2.1}$$

Inflation rate is used when calculating the 'Real Interest'. This is the interest rate, taking into account inflation and is defined by the Fisher equation [20]:

$$RealInterest = NominalInterest - Inflation \tag{2.2}$$

In many countries, the central bank has an objective to keep inflation at a certain level. For the UK, that level is 2% and this is to allow individuals and businesses to plan their future and prices appropriately [21]. Inflation rate is analysed mostly on an annual basis due to its 'smoothed' nature, removing the seasonality changes of monthly inflation rates. However, this might lose accuracy and data resulting in a poorer prediction [22]. Hence, papers that have aims to predict macroeconomic factors usually use monthly or quarterly data as they do in this paper [23].

2.2.1.3 Gross Domestic Product

Gross domestic product (GDP) measures the value of goods and services produced in a country. It estimates the size of and growth in the economy and is seen to be an estimate of a nation's overall economic activity. In many countries, their GDP is released on a quarterly basis and is published for anyone to see. According the National Office of Statistics, GDP can theoretically be calculated in three different ways:

- money spent on goods/services + value of exports - value of imports
- money earned through wages and profits
- value of goods and services produced

GDP especially is one of the most important variables for defining an economy. It is therefore equally important to forecast which is what many research papers have tried to do. However as mentioned above, this is a lagging factor and is dependant on other variables within the economy [24].

2.2.1.4 Unemployment Rate

Unemployment rate is the percentage of the labour force that does not have a job, and is measured by the Labour Force Survey (LFS) in the UK. Many factors affect and get effected by unemployment rate and therefore predicting it becomes an issue of having a few inputs but not enough data to model [25]. Usually, in a healthier economy, there is a lower percentage of unemployment. A high unemployment rate causes a decrease in general sales, the housing market and even company share prices [17]. Hence, this could be an ideal indicator to use for prediction or to be predicted as it may confirm trends within an economy.

2.2.1.5 Business and manufacturing activity

Manufacturing activity is based largely on the demand from the consumer. Businesses will only produce goods from raw materials if they can then sell it. This process means that when there is a higher demand, the public have more spare money, suggesting a more positive outlook for the economy and ergo, an increase in GDP. Several indicators can be categorised under this heading, most notably 'Factory output' or 'Industrial production' as well as 'Purchasing Managers' Index' (PMI) which can be seen below in the chart for the UK:

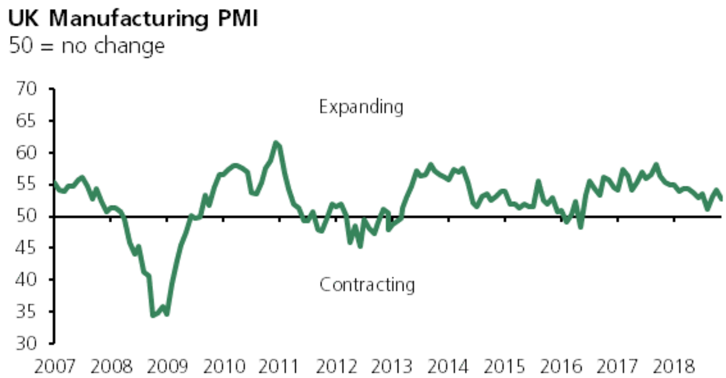


Figure 2.1: Purchasing Managers' Index (PMI) UK [26]

Another key factor indicating how healthy the economy is number of start-up businesses that get funded throughout the UK. A paper by David Mills and Laurence Schumann [27] explains that small firms account for a greater share of economic activity during economic expansion and a reduced share during economic contraction. This suggests that it is a good indication that people are willing to invest in the economy and have the extra cash in order to do so.

For innovation and businesses to keep booming, the government inject money into the economy and offer business grants. This portion of the budget funds scientific research, technological advancements and university studies, keeping the growth of industry and the consumer spending ever growing. According to Oxford Economics, they estimate that this area has contributed nearly 2.6% towards the total GDP of the UK in 2013 meaning it is a vital indicator for forecasting the economy [28].

2.2.1.6 Borrowing and Debt

Each individual country has two types of debt at a national level, the Gross Government Debt and the General Government Debt. The gross debt, according to the Maastricht treaty compromises currency, bills, bonds and loans and are presented at nominal value [29] however the general government debt deducts any liquid assets [30].

Debt is accumulated usually during harder times or times without any economic growth. When they have passed, and the economy can advance again, the debt must then start to be paid off. This is the second tenant of Keynesian economics and can be used for forecasting how an economy might look like. If the debt in only increasing, it could be a sign that the economy is stagnant however if more is being paid off then it suggests that the economy has started growing again.

2.3 Time Series

Time series data is the periodic measurement over time of a certain variable. Some examples include stock market prices, economic output and most governmental data [31]. Time series can be used for analytics, planning and prediction. Its advantage is that a lot of research has been done in attempt to model this sort of data. It is also very easy to understand as a person whereas some data-sets may be more complex. Its drawback is that without a lot of data and processing power, predicting and modelling is usually very hard to achieve.

2.3.1 ARIMA models

Auto-Regressive Integrated Moving Average models are widely used for predicting time series data [32]. These models have three parameters ARIMA(p, d, q):

- p: number of autoregressive terms (number of time lags)
- d: number of differences needed for stationarity
- q: order of the moving-average model

The process of using ARIMA for time series analysis has four steps [32]. The time series must first be stabilised or 'de-trended' by the difference method that is shown below. The process of de-trending could remove an upward trend for example and lets us predict only the change in that data, not exactly where the next point will be.

$$\nabla^d y_t = z_t \tag{2.3}$$

where d is the number of differences at period t. For example, the table below shows how an example data-set would look after d=1, d=2 and d=3:

Y	d=0 (Y_t)	d=1 ($Y_t - Y_{t-1}$)	d=2 ($Y_{t-1} - Y_{t-2}$)	d=3 ($Y_{t-2} - Y_{t-3}$)
1	5	N/A	N/A	N/A
2	7	2 (7-5 = 2)	N/A	N/A
3	13	6 (13-7 = 6)	4	N/A
4	17	4 (17-13 = 4)	-2	-6
5	29	12 (29-17 = 12)	8	10

Table 2.2: Example of differencing a data-set by d

Once the rows that have 'N/A' are removed, the ARIMA(p,d,q) model is then established for the result of the difference:

$$\varphi(B)(1-B)^d y_t = \psi(B)\varepsilon_t \tag{2.4}$$

$$\varphi(B) = 1 - \sum_{i=1}^{p} \alpha_i B^i \tag{2.5}$$

$$\psi(B) = 1 - \sum_{i=1}^{q} \beta_i B^i \tag{2.6}$$

where B is the lagged operator [32].

After the model has been trained and the error value reduced to a minimum, the model can then be used for prediction. However, ARIMA assumes a linearity within the data which we know for macroeconomic data, not to be the case. Hence, a lot of research has been done comparing other machine learning methods to the success of ARIMA models, usually providing mixed results. The disadvantage or ARIMA models is that it uses only one time series for prediction. Therefore, if other variables could be used to signal a change of direction, it cannot make use of them.

2.3.2 Stationarity

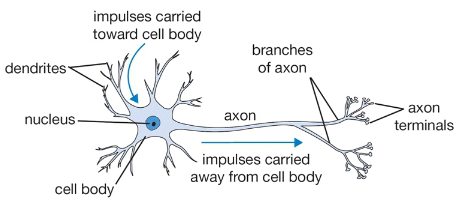
Statistical tests can be utilised to examine the stationary assumptions by decomposing the time series into three elements: trend, random walk, and stationary error. Two popular tests are the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [33] and the Augmented Dickey-Fuller (ADF) test [34]. The ADF test determines whether an autoregressive model contains a unit root, which is a feature that determines how strongly a time series is defined by a trend. The KPSS is similar, however the null hypothesis for KPSS is when it does not contain a unit root. The process of making data stationary can be seen in table 2.2. The advantage of stationarity is that the data will then have a constant location and variance allowing for easier forecasting.

2.4 Artificial Neural Networks

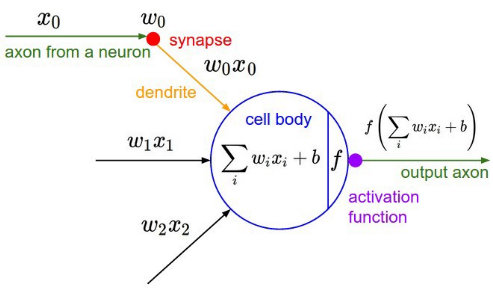
Artificial Neural Networks (ANN) were introduced in the 1940s [35] but only in recent decades have they become useful in so many fields. They are essentially a data processing system that models how the human brain works in order to solve more complex problems [36]. The inventor of one of the first neuro-computers, Dr Rober Hecht-Nielson, defined it as:

'...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.'

When modelling the brain, you must first create the most fundamental unit, called a neuron. These are the equivalent of nodes in a neural network which use mathematical equations instead of chemistry within cells. The diagram below shows how a node was inspired from a neuron within the brain:



(a) Neuron in Brain



(b) Node in ANN

Figure 2.2: Comparing neuron in brain to node in ANN [37].

The function of a node is to receive inputs from other nodes or external sources and compute an output. Each input has an associated weight which is assigned depending on the importance of that input in comparison to others. These weights are learnable through a specialised algorithm. If the function within the node outputs a value that is above a certain threshold, it sends a spike on its output axon. All output axons eventually accumulate in some fashion to provide an output or a series of outputs. The threshold within the node is calculated by the activation function, examples of which are the sigmoid and tanh functions [38].

Having defined a node, we can now create a collection of them. This forms a neural network, made of three parts:

- Input layer: This is the data you send into the neural network. One piece of data per input neuron (no calculation is performed at this point).
- Hidden layer(s): These are layers of neurons, all fully inter-connected.
- Output layer: This layer maps the results of the hidden layers to a final output(s) on one or more output neurons.

There are several variables that are required to initialise a NN. This includes the number of inputs, number of outputs, number of hidden layers and their neurons. Then each layer must have a set activation function. Different activation functions are used for a variety of reasons such as 'tanh' for categorisation. In addition to this, the NN has to have a loss function and an optimizer. The loss function evaluates a set of weights within the network and the optimizer is used to minimise this loss to create the optimal version of the network. A common combination of loss and optimizer functions are 'crossentropy' and 'adam' [39] but many more are discussed in this article [40].

After the setup, the NN will use backpropagation to enhance its training. The backpropagation algorithm was originally introduced in the 1970s, but its importance was not fully appreciated until this 1986 paper [41] when the algorithm allowed neural networks to be trained much faster than before. Backpropagation trains a neural network by changing its internal representations to improve results. In the book "Artificial Intelligence: A Modern Approach" by Stuart Russel [42], he describes the backpropagation process in more mathematical detail but in essence, when the training data goes through the network, the error is calculated between the output and the correct value, using the loss function. Therefore, the problem is really an optimisation problem to find a function that maps inputs to outputs. For neural networks, this is based on the weights between all the nodes. By plotting all the weights against the error, you get a parabolic bowl-shaped graph which in order to find a minima, a gradient descent approach is used, utilising the optimisation function to find it. The algorithm then updates the weights of the network accordingly, iteratively getting closer to the optimal set of weights.

ANNs have already proved themselves to be a leading contender for predicting time series data according to many sources, many of which are compared in the article by Khashei and Mehdi in 2010 [43]. For time series analysis, studies have shown that certain networks are more accurate than others. The standard NN is called a Feed Forward NN and is simple to implement. However, more complex models such as the LSTM NN, described below, has been modified for the purpose of time series prediction.

2.4.1 Inputs and Outputs

When creating a network, it is important to decide how many variables you want to process and how many data points need outputted. For example, you can use several macroeconomic variables to output several time steps ahead for one of those variables, or, output the next point for each variable. The table below shows a quick representation of what type of setups you can create.

	Multivariate	**Univariate**
Multi-step	Several input variables mapping to several forecasts (t+1...t+n) of one variable	One input variable mapping to several forecasts (t+1...t+n) of one variable
One-step	Several input variables mapping to one forecast (t+1)	One input variable mapping to one forecast (t+1)

Table 2.3: Variables and Forecast definitions [44]

2.4.2 Training & Testing

Neural networks are trained, validated and tested before deployment. The data you have must therefore be split into 3 different categories. The standard split would be 70% for training, 15% for validation and 15% for testing. The network trains on the training data and then tests on the validation data. It alters the weights to minimise the error on the validation data and then repeats the process with each iteration called an epoch. Once the error is minimised, it tests once on the out-of-sample test data where it provides a final result.

The other method commonly used is k-Fold validation. This is where you split your data up into K sections. You use k-1 sections for training and validation, then tests on the remaining section. You complete this k times, so each section acts as a testing section at some point, then compute and average score [45]. This is preferable for smaller data-sets, however, is not recommended for time series data as the data would not be continuous.

2.4.3 Feed-Forward Neural Networks (FFN)

Feed Forward networks input data at one end of the network and outputs are calculated at the other end. The data travels only one way and the weights and biases are adjusted when training to provide the most optimum output. It can have as many neurons and layers as required (shown in Fig 2.3) to find features in any dimension of the data. The advantage of this network is it is simple, efficient and can approximate most functions.

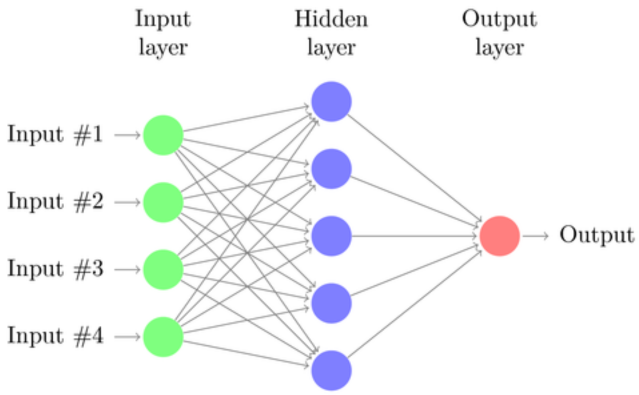


Figure 2.3: Feed-Forward Neural Network

2.4.4 Deep Learning Neural networks (DLNN)

Deep learning neural networks are a more recent development within the neural network field and have been discussed, and partly introduced, in this paper [46] following an in-depth discussion of the difficulties faced in this paper [47]. They are a branch of feed-forward networks but contain many layers and many neurons. Utilised mostly for image classification and feature extraction, they perform well but require very large data sets to train on. For time-series data, as they do not perform sequential modelling, they perform worse than recurrent neural networks (described below), however there is ongoing work into merging both these networks [48] which will make for interesting findings in the future.

2.4.5 Recurrent Neural Networks

LSTM networks are a subsection of recurrent neural networks (RNN). RNNs have the ability to re-input values outputted from a neuron, in the next cycle, keeping information persistent. However, as discussed by Hochreiter in 1991 [49] and Bengio in 1994 [50], they have one large downfall. For example, if you are predicting the next word in a sentence, that is independent of other sentences, then you can easily use a RNN. However, if this sentence is based on another paragraph, then RNNs cannot remember that far back and will therefore not be able to predict it. Stemming from this idea, Hochreiter and Schmidhuber proposed long-short term memory (LSTM) [51] networks, which are designed to remember long and short-term information.

2.4.6 Long Short Term Memory Networks (LSTM)

In essence, an LSTM has many unit blocks. Each unit has one or more memory cells and three gating units, shared by all cells in the block. It uses linear units called Constant Error Carousels (CECs) to overcome gradient vanishing or explosion problems in normal RNNs [52]. Each CEC has a fixed self-connection and is encompassed by three gating units that control the flow of information for the CEC. Fig. 2.4 shows the standard LSTM memory cell.

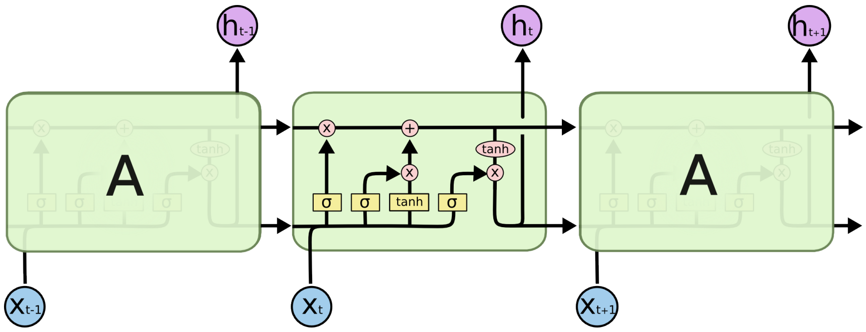


Figure 2.4: LSTM node

LSTM networks, due to their ability to remember long term and short-term trends, have been used largely for modelling time-series data, based on the theory that 'history will always repeat itself [53]. Further details of LSTM networks are explained in detail in this article [54] by Christopher Olah.

2.4.7 Implementing Neural Networks - Software

The principle of Neural Networks has been around for several decades but recently, popular software libraries and programming languages have implemented and open-sourced them for anyone to use. The two popular libraries are Keras [55] and MATLAB [56]. Keras, based on Google's Tensorflow AI library [57], is a python-based package and enables the user to create complex networks with a few lines of code and to perform in-depth debugging. MATLAB provides a visual approach however is not as versatile and limits the available functions provided.

2.5 Previous Work

Due to the rise in popularity of LSTM networks to predict time series data, several pieces of work have been published over the last few years within different application areas. The table below summarises what has been concluded up to now within time series forecasting based on reviewed papers.

Author	**Application**	**Method**	**Type of Model**
D. Kazakov & Z. Georgiev [58]	Macroeconomics	Differential Equations	Multivariate Multi-step
E. Nakamura [59]	Macroeconomics	Neural Networks	Univariate One-step
G. Zhang & D. M. Kline [60]	Macroeconomics	Neural Networks	Multivariate One-step
G. Dong [61]	Stock Markets	Neural Networks	Multivariate Multi-step

Table 2.4: Summary of previous work

As seen in the table above, much work has been done around predicting time series data using a variety of different methods (not all shown in the table). However, an obvious combination to develop now is a multi-variate and multi-step LSTM network for macroeconomics. Specifically using the idea of leading and lagging macroeconomic indicators. In light of this, the rest of the paper will explore the creation and the results of the network.

3 Problem Analysis

As discussed in the literature review, modeling time-series data using neural networks is best done by an LSTM network. As the aim of the project is to forecast macroeconomic indicators and obtain a model versatile enough to be used with a varying number of inputs and outputs, LSTMs will be used going forward. This section will explore the details of what needs to be created, tested, compared and evaluated to achieve the goals of this paper.

3.1 Hypotheses

In order to quantify what this goal is, the following hypotheses have been set out:

1. Using a leading indicator as well as lagging indicators will obtain a lower root mean squared error (RMSE) value than using just lagging indicators as inputs.
2. Based on an RMSE result with variables used in [58], an LSTM network should perform at least as well as differential equations.
3. Based on an RMSE result, a univariate LSTM network should perform at least as well as an ARIMA model.

3.2 Objectives

In order to answer the above hypotheses, the following objectives have been set out:

1. Evaluate how accurately an LSTM network can be used to forecast lagging macroeconomic parameter using itself only. **(One-to-one model)**
2. Evaluate how accurately an LSTM network can be used to forecast a lagging macroeconomic parameter using three lagging macroeconomic parameters, including itself. **(Lagging-to-one model)**
3. Evaluate how accurately an LSTM network can be used to forecast

a lagging macroeconomic parameter using three lagging macroeconomic parameters and a leading indicator. **(All-to-one model)**

4. Complete the above using one-step and multi-step prediction models.
5. Use the best of the above multi-step models and compare the results to equation discovery method detailed in this paper [58].
6. Use the best of the above multi-step models and compare the results to an ARIMA model.

3.3 Requirements

These requirements have been collated to achieve the above objectives. In addition, this paper [62] concludes that the most optimal network parameters are similar for stocks in the same industry which is a theory that will be used within this paper.

1. Gather data that will be used for prediction. This must include the same columns from [58] for comparison as well as a leading macroeconomic factor.
2. Pre-process the data so it can used by an LSTM network.
3. Create a network for each indicator using one-step and multi-step prediction. Perform this with univariate and multivariate data.
4. Obtain a root mean squared error (RMSE) value for each time step in each forecast.
5. Compare the results to the ARIMA model and the equation discovery model from [58].
6. Analyse the results of all models to determine whether using a leading indicator has improved performance.

3.4 Problems and Resolutions

The most obvious problem that will be faced is the regularity of the data. For example, some indicators will be measured monthly from 1987 but some are measured yearly only starting in 1998. This means there is a different amount of data for each indicator. The way neural networks will process data relies on having an input for each time interval. Consequently, in order to bypass this problem, either data has to be chosen that is published at the same time intervals, or interpolation must be used. The disadvantage of interpolation is that it can make the network approximate the wrong function which will lead to bad results. Hence, for this paper, we will use data that is recorded at the same intervals.

The second issue is that all the indicators have been recorded starting in different years. GDP was started 1955 but Unemployment rate started in 1971. This means that we cannot use the full data set for GDP in order to train the network, but only use data from 1971 onward. Unfortunately, some indicators have only been recorded since the 1990s. This results in a lot less data for training the network, affecting the end result. The solution is to just use data that has been recorded for at least 60 years which provides a maximum amount of data.

Ethically, there are some that believe that trading involves individuals or firms gaining at someone's expense and it should therefore not be allowed. Another ethical issue with using algorithms to trade is their ability to cause drastic changes following a chain of events that could lead to financial crisis for a stock, country or even globally, such as the 'flash crash' of 2010 [63]. This projects looks at predicting economic factors within the UK. It uses professional methods, some which are already used world-wide, and all the data has come from public sources. The results obtained from this experiment are not detailed enough to make professional business decisions and could not be used within trading as the main purpose is research.

3.5 Evaluating the results

The calculation of errors and evaluation of the results will be crucial in coming to a conclusion about whether an LSTM can be used for prediction. The models are trained on the same data and used to predict the out of sample testing data.

3.5.1 Calculating Errors

Mean absolute error measures the average magnitude of the errors in a set of predictions, without considering their direction.

$$\frac{1}{n}\sum_{j=1}^{n}|\hat{y}_i - y_i| \tag{3.1}$$

Root mean-square error is a quadratic scoring rule that also measures the average magnitude of the error.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}(\hat{y}_i - y_i)^2} \tag{3.2}$$

The similarity between MAE and RMSE is that they both output an error with equal proportions, no matter the size of the data. However, the use of the squared function in RMSE penalises larger errors more so than smaller ones. This is useful when larger errors are more detrimental and need to be avoided. Both these errors can be used in back-propagation of a neural network and by comparison, better results have been obtained using RMSE, hence this paper will continue with it as an error function. The error value will be calculated for each model on the same, out of sample testing data, to provide the most accurate results.

When forecasting using the multi-step model, 2 years of data (8 time steps) will be predicted. Getting an average RMSE value for the prediction would remove all information about how well it performed throughout the 2 years. Therefore, after creating all the predictions in the test data, an average for each time step will be calculated. This will form a table of average RMSE values for time-steps t+1 to t+8. This allows us to see whether the forecast got more deviated or stayed on course.

3.5.2 Statistical tests for independent measures

When comparing models, you need a test to check whether the models actually create statistically different results. For accuracy, each LSTM network will be run 100 times and then this test test is performed.

To compare 3 or more sets of results, the Kruskal-Wallis test is used [64]. It is a non-parametric test to compare independent samples of any size. The null hypothesis assumes that all samples are from identical populations, in which case there would be no difference between the models. If this is rejected, then at least one sample originates from a different population. This means at least one of the models performed differently. The null-hypothesis is rejected if the p-value is below 0.05. However, if rejected, this test can't tell which sample is statistically different. Hence, we use the Mann-Whitney U-test [65]. It compares 2 independent samples and checks if it is equally likely that a randomly selected value from one sample will be less than or greater than a randomly selected value from a second sample. If the null hypothesis is rejected, then it confirms that these two samples are different. This is then performed on each combination of samples to check which models provide significantly different results.

4 Design and Implementation

4.1 Choosing the data

The data will be sourced from the National Office of Statistics for the UK [66]. The following columns of data have been used:

1. Inflation Rate (RPI) (%) [67]
2. Interest Rate (%) [68]
3. Change in GDP (%) [69]
4. Change Production Output (%) [70]

This data will then have the columns that are mentioned in the [58] paper and a leading indicator (Production Output). Each indicator has been recorded from different dates and at different time intervals. In order to solve this, the data has been limited to range from 1959 (Q1) to 2018 (Q4) creating 60 years worth of data.

4.2 Pre-processing the data

The most important step ahead of training a network is the data pre-processing. The first step is to ensure that all columns are real-value numbers. As they are all prices or percentages, this step is already complete. Next, no data should be missing otherwise we will encounter errors. Fortunately, the data downloaded is complete. Following this, outliers are often removed from the data set so the network does not learn them. This may be seen as removing important aspects of the data however it can make for better results on average. In this case, an easy test of finding values is using a windowed mean and removing all values that are further away than 3 standard deviations. This data had no significant outliers, so no data was removed.

Next, all redundant factors must be removed. This means that if two columns are very highly correlated, then there is no reason to have them both in, as it would only increase the network size and not provide any new knowledge. The plot matrix below (Fig. 4.1) shows a comparison between each variable to check for correlation. As you can see, no columns will have to be removed.

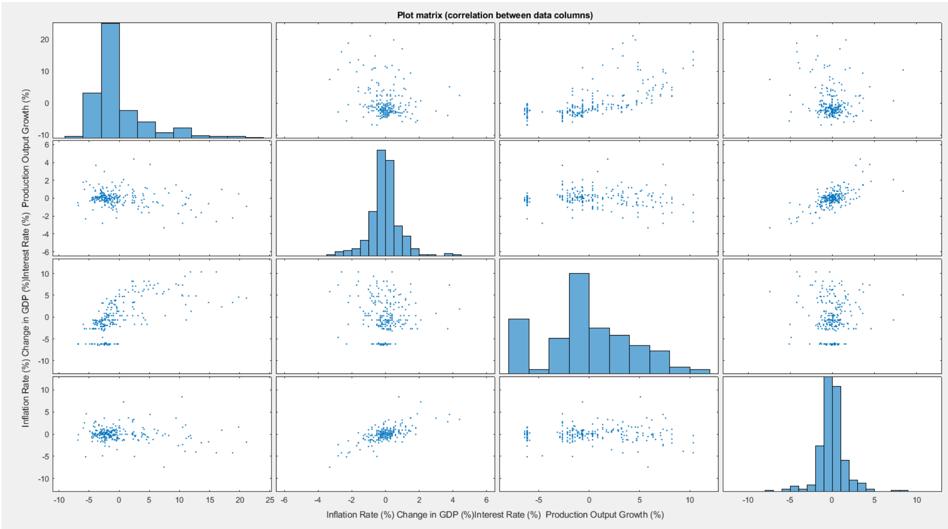


Figure 4.1: Macroeconomic indicator correlation matrix

As found by testing and in the literature review, it is highly recommended to make the data stationary and therefore predict the change in value. Hence, we make the inputs the differenced data-set. Note the actual data point can be calculated by reversing the process. This only has to be performed on the columns that are not in this format already. When doing this, we can also use the Dickey-Fuller test to ensure that the data is stationary.

Also visible on this figure is that the data has been normalized around the mean. It therefore is centred around 0 and has unit variance. This removes the skew within the data and scales the data into a certain range. No neuron can therefore get saturated by extremely large inputs allowing the network to learn an outcome without bias. The equation to normalise the data, for each data point, v, in column, x, is:

$$v' = \frac{v - mean(x)}{stddev(x)} \tag{4.1}$$

Now that the data has been modified numerically, we must modify its structure.

4.2.1 Prepping data for Neural Networks

Currently, all the data is in a 2D format (indicator, dates) however for an LSTM to process data, it must have the correct dimensions. The final data should be a 3D matrix of N number of rows, by M number of indicators, by

T number of lags. The next step is to add lags in. This allows the network to see the last few data points in order to complement the decision process. With lags=1 (t=1), the input data for each step will be a vector that looks like the following (table 4.1):

RPI	Interest	GDP	Output	RPI-1	Interest-1	GDP-1	Output-1

Table 4.1: Input Vector for LSTM Network with t=1

There exists two ways to split the data up into training and testing data now that we have the final data-set. Either K-Fold validation or into successive batches. As discussed in the literature review, we cannot use K-Fold validation so we successive partitioning instead. There is 60 years worth of data with 4 data points per year. The split will be as per the table 4.2. This will maximise the amount of data to train on but will also give an accurate result when testing or validating the model.

	Percentage of total points	**Number of points**
Train	70%	168 (42 years)
Validation	15%	36 (9 years)
Test	15%	36 (9 years)
Total	100%	240 (60 years)

Table 4.2: Data split for training, validation and testing

The trained model will be used to forecast the next 8 time steps (2 years), at each point in the test data. Therefore:

$$NumberMultistepForecasts = NumberTestValues - 8 = 28 \quad (4.2)$$

The multistep forecasts will be compared to the equivalent data in the test set to work out a RMSE value for each step in the forecast. A similar process will be true for the one-step forecasts:

$$NumberOneStepForecasts = NumberTestDataPoints - 1 = 35 \quad (4.3)$$

Using the RMSE values, it is then easy to compare how well the models performed against equation discovery or ARIMA models.

4.3 Creating the Neural Network

Using Keras, many LSTM networks were tried and tested in order to find the best performing one. This involved going through many combinations of setup configurations and using suggested models from online sources [71]–[73] as a starting point. Final results were calculated by following these steps:

1. Start with a simple architecture with the required number of inputs, 1 hidden neuron and 1 output
2. Iterate through all combinations of network sizes and number of neurons to find the lowest RMSE value (based on 100 iterations for an accurate average)
3. Complete the step above using all appropriate loss functions, optimization functions and number of lags
4. Review all the results and apply theorem of Occam's Razor [74] (choose simplicity over complexity for networks with equal results)
5. Choose the best architecture and setup for each indicator

The other advantage of Occam's razor is that large architectures can overfit easily and provide bad results during testing. All the best architectures found are summarised in this table 4.3.

Variable	**Inflation**	**GDP**	**Interest**	**Output**
Number of Epochs	15	15	15	15
LSTM units (layer 1)	32	36	32	32
Number of Lags	12	6	5	20
Optimizer function	SGD	SGD	SGD	SGD
Loss function	MSE	MSE	MSE	MSE
Dropout rate	0	0	0	0
Activation function	linear	linear	linear	linear

Table 4.3: LSTM Network configuration summary

5 Results and Evaluation

5.1 Results

This chapter displays the results of each of the neural networks per indicator. Each sub-section will have a table stating the RMSE for the prediction at each time interval in the future and then a summary of the observations.

The graphs in this section are interpreted in the following way:

1. Blue line is the true values for that indicator.
2. The graph can present the entire history or just the test data.
3. The red lines start at a point on the blue line and represent a prediction from that point in time. They can be 1 or 8 time steps.
4. Only every 4 red lines are displayed (for visibility).
5. The green line connects the final point of each prediction together. It forms the t+1 (one-step) or the t+8 (multi-step) line.

5.1.1 One Step Prediction

This section contains all results for the LSTM models that predicts one point into the future. Table 5.2 shows, for each indicator, the model's RMSE results:

Model	**Inflation RMSE**	**GDP RMSE**	**Interest RMSE**
One to One	0.69606	0.362951	0.098734
All to one	1.046394	0.464202	0.178685
Lagging to one	0.851892	0.366887	0.140822

Table 5.1: RMSE for LSTM network one step prediction

These results vary significantly for each indicator. The inflation indicator has a very high RMSE suggesting that the model found it very difficult to predict even the general direction in which the next point may be. We can see from the graph 5.1 that the data within the training set doesn't represent the patterns found in the test set. This is most likely due to the effects of the financial crisis in 2008 however this cannot remain an excuse as the economy must be forecast nonetheless.

We also note from this table the one-to-one model is the best performing and the all-to-one the worst, contrary to hypothesis 1. As this is only a one-step prediction, it is best to confirm this with the multi-step models' results however it is an indication the hypothesis may be wrong. This may be down to the extra inputs and noise generated by other indicators.

The graph 5.1 shows inflation being predicted using a one-step, all-to-one model, and then graph 5.2 shows a close-up on the test section:

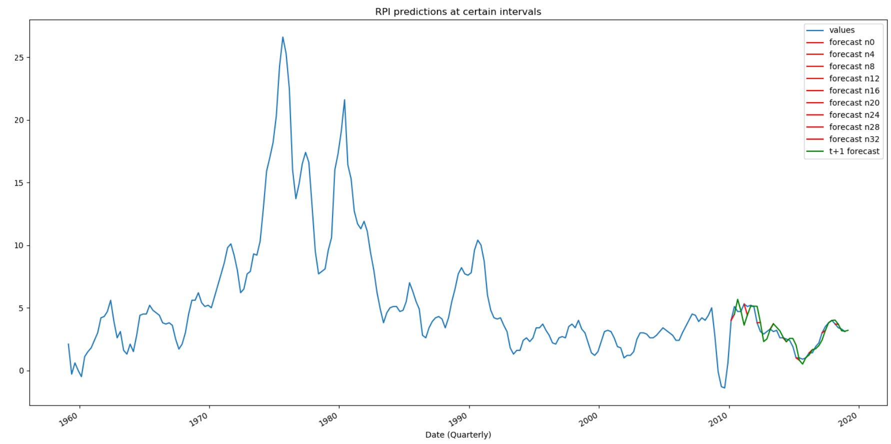


Figure 5.1: Inflation One step ahead prediction - all data

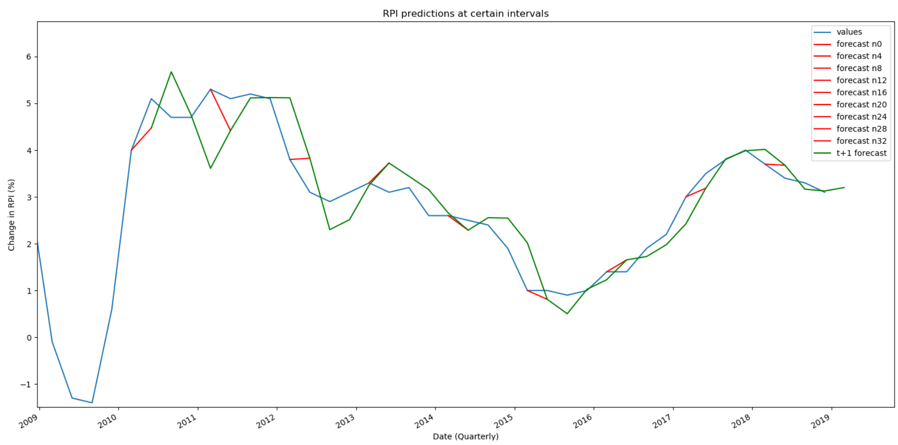


Figure 5.2: Inflation One step ahead prediction - zoomed in on test section

You can see that although the green line fits quite closely to the original blue one, at each step, the green line does not do a good job at predicting where it will be next. It could be argued that the line represents more of a lagging moving average rather than a prediction line. It is evident that this cannot be used for prediction.

5.1.2 Multi-step

5.1.2.1 One to One Model

Table A.1 shows the RMSE results for an indicator being predicted by historic values of that same indicator. Usually, the more into the future your prediction is, the more likely the prediction is further away from the truth. However, the table above shows the opposite, for all indicators. You can see that the RMSE value actually decreases when it approaches t+8 and is very high when it is close to t. This suggests that the model is not predicting well and the line is taking a completely different route to what it should be. The graph 5.3 displays inflation along with the various forecasts of 8

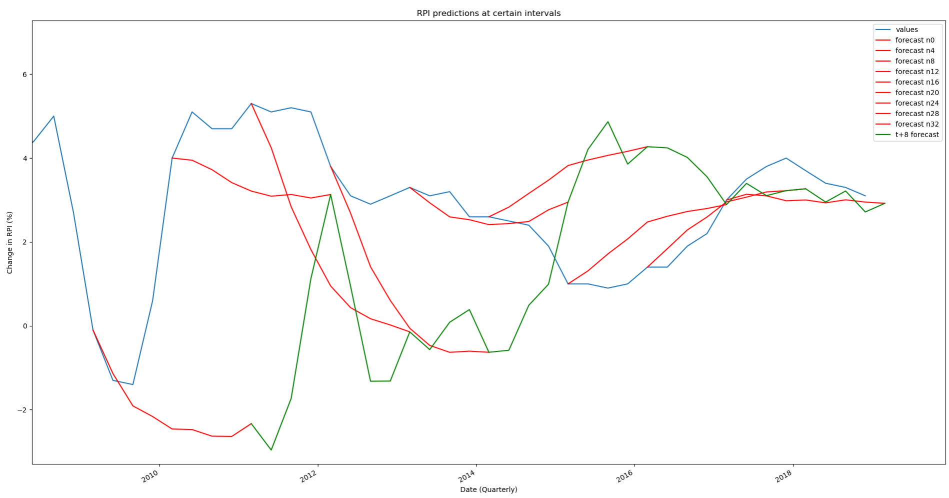


Figure 5.3: Inflation Multi step ahead prediction - zoomed in on test section

time steps. It shows there is very little correlation between the prediction and the true values. This could be down to the aforementioned notion that the financial crisis has changed the seasonal patterns. It also might be affecting the earlier forecasts more as they are closer to 2008. This is especially similar with the RPI indicator as from 2008 onwards, there has been a target for the interest rate, set by the central bank. As the network has trained on data before 2008 and is tested on after 2008 it has not been able to model this change (Fig 5.4). Despite this, the RMSE results are very low. This alludes to the fact that RMSE as an evaluation method is not suitable for this type of problem and a human element is required instead.

5.1.2.2 All to One Model

Table A.2 presents the RMSE values for the all-to-one model. Primarily we can see that using all the indicators makes for slightly worse results

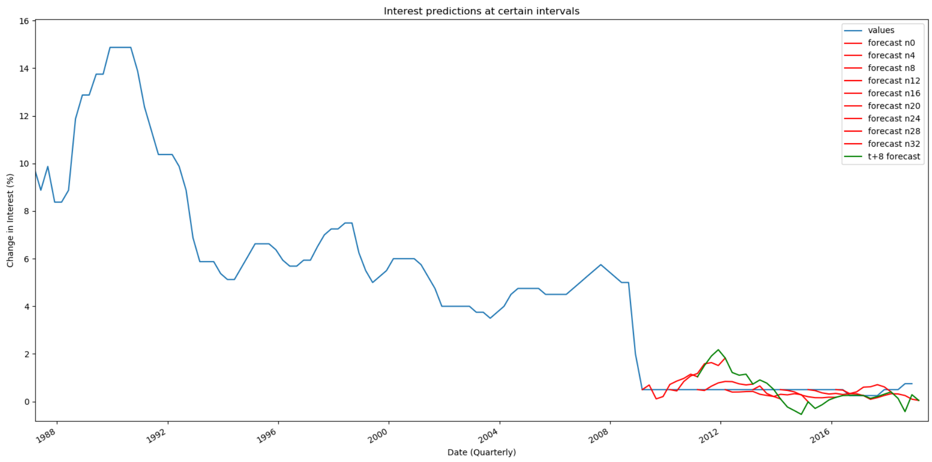


Figure 5.4: Interest prediction (zoomed in on last half of data)

than using just the same column as input. However, apart from inflation, the pattern of decreasing RMSE values is not as evident here, especially for inflation and GDP. Worse results suggest the network had difficulty modelling the extra data and/or found bad links between each data series. This might have also been enhanced by all the indicators being affected differently due to the financial crisis making the model predict even more incorrect values.

5.1.2.3 Lagging to One Model

Table A.3 shows the results of a lagging-to-one LSTM. Using the lagging columns provides a similar output to using all the columns with a slight decrease in error for GDP and Interest indicators. This confirms the theory that the more data there is, the more noise is modelled. This is probably down to limited data for the network to train on but also the fact that using only lagging parameters should in theory not help, so adding them in just creates noise.

5.1.2.4 Summary

The graph A.2 plots the RMSE values for each model, for each indicator. The most striking observation, and a response to hypothesis 1, is that there is not clear win for any of the models. A reason for this may be down to lack of data as the models could not find proper correlations. It could also be that we are attempting to model the wrong data and that the leading indicator does not actually provide useful information for forecasting. Either way, adding in more data has caused either a larger error or very limited effect on the prediction.

The largest problem faced by all models was that of predicting with no fundamental understanding. This was evident when, following the financial crisis, some patterns looked slightly different making it hard work for the LSTM to predict. In addition, the bank interest target, introduced following the crisis, was not known to the network. It could be that adding these events in would help, however, if we were predicting that many years ago, these events would be unknown so it could be argued that this was fair. However, as the majority of the time there is not an 'international crisis', completing the test on other data may be more indicative of how well these networks perform.

5.2 Comparison to previous work

5.2.1 ARIMA models

This table depicts the RMSE for the one-step forecast using the ARIMA model.

Model	**Inflation RMSE**	**GDP RMSE**	**Interest RMSE**
One to One	0.486	0.312	0.177

Table 5.2: RMSE for ARIMA one step prediction

The results are similar to those of the one-step, one-to-one LSTM model however the ARIMA model performed significantly better in the Inflation prediction compared to the LSTM network. Unfortunately, this is another case where the RMSE does not show the full picture. Even though the RMSE is lower, the ARIMA model just predicted a straight line making it not very useful for users, as you can see per the graph A.1.

The table A.4 are the results from the ARIMA model predicting 8 values into the future based on the same data used for the LSTM one-to-one. In this case, the inflation is a lot higher than the one-to-one LSTM model. Interestingly, it has the opposite trend to that of the LSTM whereby the further into the future the prediction is, the higher the RMSE (which was expected initially). Similar results for GDP as well where the RMSE value increases to a much higher value. As noted above, the ARIMA model predicts roughly a straight line and in the case that the true values deviate from this over the test period, the RMSE value will increase.

5.2.2 Comparison to LSTM results

Here are the results (Fig. 5.3, graph 5.5) comparing ARIMA (one-to-one), Differential Equations (lagging-to-one) and each LSTM network. Note that the bold values indicate the best RMSE for that indicator and the graph has a cap at 1.0 for RMSE for visibility reasons. The ordinary differential equation (ODE) with step uses a thresholding model described in (Fig.1 in D. Kazakov [58]).

Model	Inflation RMSE	GDP RMSE	Interest RMSE
ARIMA	1.209	**0.373**	0.766
ODE (step)	**0.215**	0.588	0.299
LSTM (lagging-to-one)	0.737	0.480	0.136
LSTM (one-to-one)	0.716	0.485	**0.125**
LSTM (all-to-one)	0.710	0.504	0.168

Table 5.3: RMSE for each model

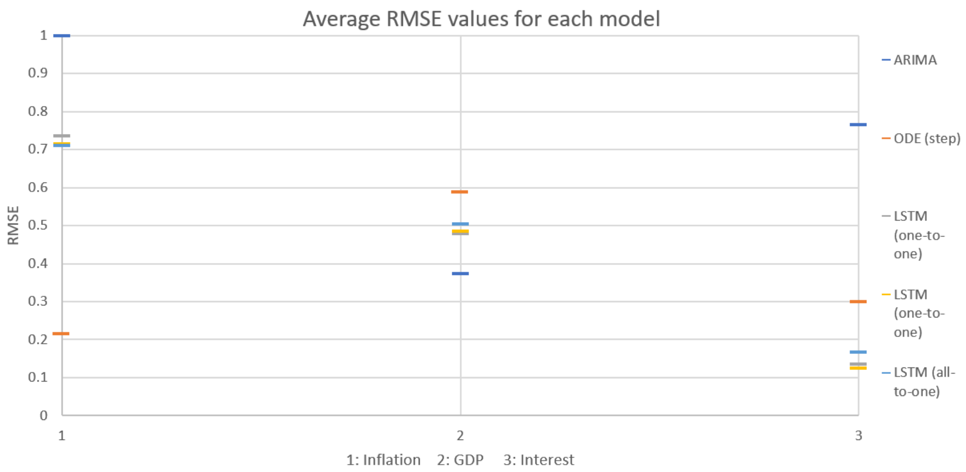


Figure 5.5: RMSE values for each model (lower down the better)

We can quite clearly see that there is no top performer. It is interesting to view the range of the RMSE values. For inflation, the range is very high, suggesting it is quite hard to predict with only the ODE being able to predict it with any accuracy. The GDP on the other hand, has a small variance and all models found it equally difficult to predict. When performing a Kruskal-Wallis between the LSTM networks using 100 iterations worth of multi-step predictions, the result is significant. The p-value was 0.00001 for each indicator and therefore smaller than 0.05. This means the null-hypothesis is rejected and there is a statistical difference in accuracy between networks. In order to test which one, the Wilcoxon signed-rank test was performed on the three combinations of networks, all of which rejected the null hypothesis. This suggests that each model did performed differently, for each indicator, despite the very similar results according to the RMSE averages.

6 Conclusions and Future Work

6.1 Conclusions

The importance of predicting macroeconomic parameters is more vital than ever with the vast amount of businesses having to plan their future. This coupled with investment companies using it for trading makes it on-going research. A range of objectives were set out in this paper and their results are discussed below.

1. **One to one model** - According to the RMSE results, this model on average proved to be the most accurate for one-step and multi-step predictions. Contrary to my hypothesis, using uni-variate data minimised the difference between the outputs and true values. However, when viewing the predictions on a graph, it can be clearly seen that the results are not useful for predicting.

2. **Lagging to one model** - This model was the second most successful however showed a larger decrease in performance than expected. As all the inputs were lagging, no indicator should signal a change for any another indicator, therefore it would make sense to obtain similar results to the one-to-one model. As this is not the case, it must be that the noise from the indicators was getting incorporated into the prediction and negatively affecting it, suggesting over-fitting or lack of training data. This would have to be tested more thoroughly by using different combinations of indicators to work out if one is affecting it.

3. **All to one model** - This model performed the worst on average but the best for inflation, disproving the initial hypothesis. The leading factor was meant to indicate that a change was to happen however results proved otherwise. Possible reasons for this might be lack of training and test data as the network was not able to pick up on the correlation, or, the wrong leading indicator was used. Research is needed to find the best leading indicator for this problem.

4. **Comparison to other models** - The LSTM networks were compared to Equation discovery and ARIMA models. No conclusive results were gathered due to very different performances by each model on each indicator. LSTM models performed better than the ODE on GDP and interest however were vastly outperformed on the inflation. The

ARIMA model, although performing best on the GDP predictions, performed a lot worse on the others as it predicted a straight line. The LSTM models do have statistically different results; however, they have been based on the RMSE values, which have been found to be inadequate for quarterly forecasting. They can still be used to perform long term investments however do not provide enough accuracy to say where an indicator will be next quarter. In light of this, the tests should be redone using a different evaluation method, possibly following the method in [58] and consulting an economist to look at the predictions.

6.2 Future Work

The impossible but necessary task of getting accurate predictions will always exist and a lot more work is required to make it more effective. After completing the tests described in this paper there are several points that can be looked at for future work:

1. **More data** - There was not enough training data for a model to be created. The indicators used had an early date for when records began but many indicators do not. To increase data size, experimenting with interpolation to create monthly data or using a vast amount more columns may be viable options.

2. **Different columns** - It would be interesting to see whether different economic indicators have a more positive effect on the result. This could extend to stock market data or factors within foreign economies like the EU.

3. **Evaluation** - A different evaluation method must be used as accuracy cannot be observed using RMSE. An expert opinion from an economist will provide an undesirable element of bias however might be more insightful. Or, the project could be modelled as an investment fund with a starting capital. Trades would be performed using expected results and a final net income could be calculated. This would involve exploring ethical and professional issues related to trading.

4. **ANN architectures** - Testing architectures other than LSTMs and using an array of optimisation and loss functions to improve on the network. This could also be simplified into a categorisation algorithm allowing the use of radial basis networks.

5. **Extra inputs** - A downside to the LSTM network was the inability to add any constraints (evident when predicting the interest rate after the financial crisis). Having an element of the wider economic picture could provide better results.

Appendices

A Results

Time step	Inflation RMSE	GDP RMSE	Interest RMSE
t+1	0.952385	0.468770	0.243960
t+2	1.007448	0.545778	0.092440
t+3	0.973395	0.482591	0.138746
t+4	0.848485	0.476743	0.072573
t+5	0.752399	0.469579	0.094856
t+6	0.445727	0.475356	0.099176
t+7	0.385566	0.477019	0.122305
t+8	0.366222	0.481450	0.134224
Average	0.716453	0.484661	0.124785

Table A.1: RMSE for LSTM network using one data source

Time step	Inflation RMSE	GDP RMSE	Interest RMSE
t+1	0.906180	0.463218	0.310158
t+2	0.975964	0.532797	0.172879
t+3	0.936834	0.533609	0.113031
t+4	0.860537	0.478739	0.533609
t+5	0.714439	0.477185	0.114180
t+6	0.431733	0.533114	0.116213
t+7	0.417566	0.518333	0.179383
t+8	0.440253	0.492380	0.197158
Average	0.710438	0.503672	0.167680

Table A.2: RMSE for LSTM network using all data sources

Time step	Inflation RMSE	GDP RMSE	Interest RMSE
t+1	0.976317	0.417553	0.288028
t+2	1.020452	0.531921	0.078415
t+3	1.044172	0.499936	0.105350
t+4	0.870822	0.476867	0.101655
t+5	0.723801	0.470908	0.093215
t+6	0.434508	0.482722	0.091162
t+7	0.401588	0.475593	0.192577
t+8	0.422230	0.482406	0.140782
Average	0.736736	0.479739	0.136398

Table A.3: RMSE for LSTM network using only lagging data sources

Time step	Inflation RMSE	GDP RMSE	Interest RMSE
t+1	0.442371	0.397247	0.162488
t+2	0.765382	0.393246	0.335058
t+3	1.006711	0.392219	0.510265
t+4	1.252105	0.376740	0.675591
t+5	1.425356	0.347047	0.862061
t+6	1.511432	0.360561	1.027822
t+7	1.586562	0.360119	1.191127
t+8	1.684867	0.353473	1.362422
Average	1.209348	0.372581	0.765854

Table A.4: RMSE for ARIMA network using one data source

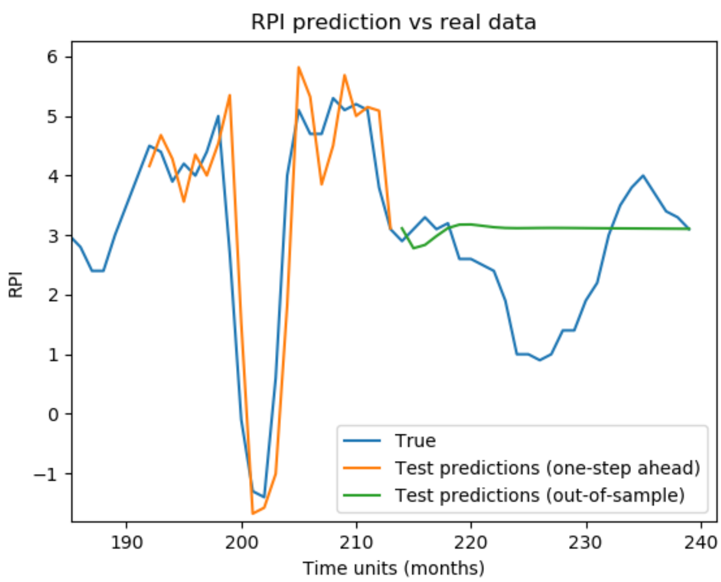


Figure A.1: RPI ARIMA predictions

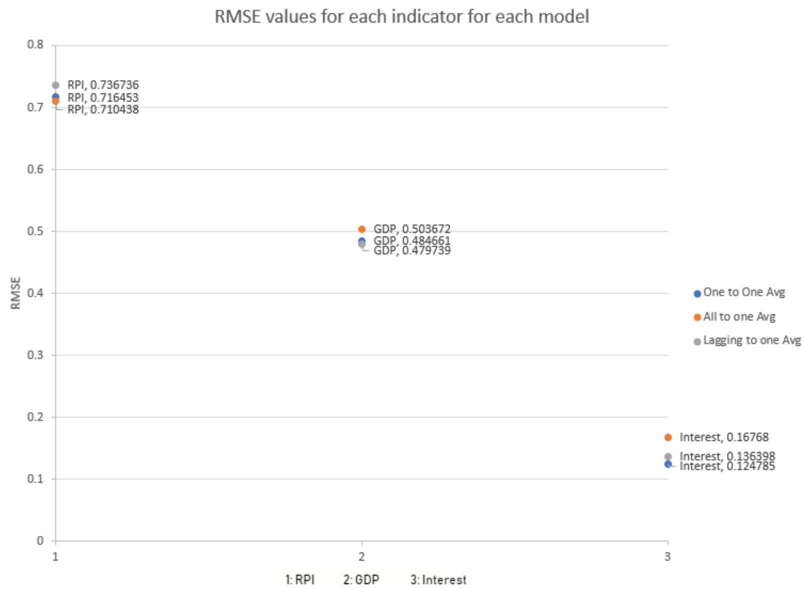


Figure A.2: RMSE values for each model, for each indicator

Bibliography

[1] J. J. Angel and D. McCabe, ‘Fairness in financial markets: The case of high frequency trading’, *Journal of Business Ethics*, vol. 112, no. 4, pp. 585–595, 2013.

[2] L. Cao and F. E. Tay, ‘Financial forecasting using support vector machines’, *Neural Computing & Applications*, vol. 10, no. 2, pp. 184–192, 2001.

[3] Y. S. Abu-Mostafa and A. F. Atiya, ‘Introduction to financial forecasting’, *Applied Intelligence*, vol. 6, no. 3, pp. 205–213, 1996.

[4] Z. Bodie, A. Kane and A. J. Marcus, *Essentials of investments*. McGraw-Hill/Irwin Taipei, 2013.

[5] (Mar. 2019). Warren buffet, Forbes, [Online]. Available: https://www.forbes.com/profile/warren-buffett/. (accessed: 17.02.2019).

[6] E. Cheng, ‘Just 10% of trading is regular stock picking, jpmorgan estimates’, [Online]. Available: https://uk.finance.yahoo.com/news/death-human-investor-just-10-204900514.html, (accessed: 22.01.2019).

[7] P. D. Easton, ‘Pe ratios, peg ratios, and estimating the implied expected rate of return on equity capital’, *The accounting review*, vol. 79, no. 1, pp. 73–95, 2004.

[8] B. Lev and S. R. Thiagarajan, ‘Fundamental information analysis’, *Journal of Accounting research*, vol. 31, no. 2, pp. 190–215, 1993.

[9] M. P. Taylor and H. Allen, ‘The use of technical analysis in the foreign exchange market’, *Journal of international Money and Finance*, vol. 11, no. 3, pp. 304–314, 1992.

[10] C. C. Hardy, *The Investor’s Guide to Technical Analysis*. Tata McGraw-Hill Education, 1978.

[11] N. G. Mankiw, *Principles of Macroeconomics*, (accessed: 22.01.2019).

[12] W. Easterly and S. Rebelo, ‘Fiscal policy and economic growth’, *Journal of monetary economics*, vol. 32, no. 3, pp. 417–458, 1993.

[13] T. E. o. E. Britannica, *Bank of england*, Jan. 2019. [Online]. Available: https://www.britannica.com/topic/Bank-of-England.

[14] (Jan. 2019). Bank of england, Bank of England, [Online]. Available: https://www.bankofengland.co.uk/. (accessed: 22.01.2019).

[15] M. Friedman, 'The role of monetary policy', in *Essential Readings in Economics*, Springer, 1995, pp. 215–231.

[16] M. K. Kim and C. Wu, 'Macro-economic factors and stock returns', *Journal of Financial Research*, vol. 10, no. 2, pp. 87–98, 1987.

[17] K. Smith, 'List of 16 major leading lagging economic indicators', [Online]. Available: https://www.moneycrashers.com/leading-lagging-economic-indicators/, (accessed: 17.02.2019).

[18] (Jan. 2019). Inflation and price indices, Office for National Statistics, [Online]. Available: https://www.ons.gov.uk/economy/inflationandpriceindices. (accessed: 22.01.2019).

[19] J. Arlt and M. Arltová, 'Forecasting of the annual inflation rate in the unstable economic conditions', in *Mathematics and Computers in Sciences and in Industry (MCSI), 2015 Second International Conference on*, IEEE, 2015, pp. 231–234.

[20] B. K. Marcus, *Investments*. McGraw-Hill Education, 2014, p. 121.

[21] (Aug. 2018). Inflation and the 2% target, Bank of England, [Online]. Available: https://www.bankofengland.co.uk/monetary-policy/inflation. (accessed: 09.02.2019).

[22] J. Arlt and M. Arltová, 'The yearly inflation rate and its forecasting', *International Proceedings of Economics Development and Research*, vol. 59, p. 21, 2013.

[23] D. Kazakov and T. Tsenova, 'Equation discovery for macroeconomic modelling', 2008.

[24] G. Camba-Mendez, G. Kapetanios, R. J. Smith and M. R. Weale, 'An automatic leading indicator of economic activity: Forecasting gdp growth for european countries', *The Econometrics Journal*, vol. 4, no. 1, S56–S90, 2001.

[25] G. Wang and X. Zheng, 'The unemployment rate forecast model based on neural network', in *Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on*, IEEE, 2009, pp. 1–4.

[26] G. Tyler, *Manufacturing: Key economic indicators*, Feb. 2019. [Online]. Available: https://researchbriefings.parliament.uk/ResearchBriefing/Summary/SN05206.

[27] D. Mills and L. Schumann, 'Industry structure with fluctuating demand', *American Economic Review*, vol. 75, no. 4, pp. 758–67, 1985. [Online]. Available: https://EconPapers.repec.org/RePEc:aea:aecrev:v:75:y:1985:i:4:p:758-67.

[28] *The impact of the innovation, research and technology sector on the uk economy*. [Online]. Available: https://www.oxfordeconomics.com/recent-releases/the-impact-of-the-innovation-research-and-technology-sector-on-the-uk-economy.

[29] (Jan. 2019). Uk government debt and deficit, Office for National Statistics, [Online]. Available: https://www.ons.gov.uk/economy/%20governmentpublicsectorandtaxes/%20publicspending/bulletins/%20ukgovernmentdebtanddeficitforeurostatmaast/june2018. (accessed: 14.02.2019).

[30] (Mar. 2014). General government gross debt (maastricht definition), OECD, [Online]. Available: https://stats.oecd.org/glossary/detail.asp?ID=1161. (accessed: 14.02.2019).

[31] D. Agnieszka and L. Magdalena, 'Detection of outliers in the financial time series using arima models', in *2018 Applications of Electromagnetics in Modern Techniques and Medicine (PTZE)*, IEEE, 2018, pp. 49–52.

[32] Y. Du, 'Application and analysis of forecasting stock price index based on combination of arima model and bp neural network', in *2018 Chinese Control And Decision Conference (CCDC)*, IEEE, 2018, pp. 2854–2857.

[33] D. Kwiatkowski, P. C. Phillips, P. Schmidt and Y. Shin, 'Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?', *Journal of econometrics*, vol. 54, no. 1-3, pp. 159–178, 1992.

[34] D. A. Dickey and W. A. Fuller, 'Distribution of the estimators for autoregressive time series with a unit root', *Journal of the American statistical association*, vol. 74, no. 366a, pp. 427–431, 1979.

[35] W. S. McCulloch and W. Pitts, 'A logical calculus of the ideas immanent in nervous activity', *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[36] H. M. Igor Aleksander, *An introduction to neural computing*, 2nd ed. 1990.

[37] (Aug. 2017). A gentle introduction to neural networks series, towardsdatascience, [Online]. Available: https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc. (accessed: 14.02.2019).

[38] A. S. V. (2017). Understanding activation functions in neural networks, [Online]. Available: https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0.

[39] D. P. Kingma and J. Ba, 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980*, 2014.

[40] J. Brownlee. (Jul. 2017). Adam optimization algorithm for deep learning, [Online]. Available: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

[41] D. E. Rumelhart, G. E. Hinton and R. J. Williams, 'Learning representations by back-propagating errors', *nature*, vol. 323, no. 6088, p. 533, 1986.

[42] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, ser. Always learning. Pearson, 2016, ISBN: 9781292153964.

[43] M. Khashei and M. Bijari, 'An artificial neural network (p, d, q) model for timeseries forecasting', *Expert Systems with applications*, vol. 37, no. 1, pp. 479–489, 2010.

[44] J. Brownlee, 'Multivariate, multi-step, and multi-site time series forecasting', Aug. 2019. [Online]. Available: https://machinelearningmastery.com/standard-multivariate-multi-step-multi-site-time-series-forecasting-problem/.

[45] R. Kohavi *et al.*, 'A study of cross-validation and bootstrap for accuracy estimation and model selection', in *Ijcai*, Montreal, Canada, vol. 14, 1995, pp. 1137–1145.

[46] Y. LeCun, Y. Bengio and G. Hinton, 'Deep learning', *nature*, vol. 521, no. 7553, p. 436, 2015.

[47] X. Glorot and Y. Bengio, 'Understanding the difficulty of training deep feedforward neural networks', in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[48] Z. Xu, S. Li and W. Deng, 'Learning temporal features using lstm-cnn architecture for face anti-spoofing', in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, IEEE, 2015, pp. 141–145.

[49] S. Hochreiter, 'Investigations on dynamic neural networks', *Diploma, Technical University*, vol. 91, no. 1, 1991.

[50] Y. Bengio, P. Simard and P. Frasconi, 'Learning long-term dependencies with gradient descent is difficult', *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[51] S. Hochreiter and J. Schmidhuber, 'Long short-term memory', *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[52] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, *Gradient flow in recurrent nets: The difficulty of learning long-term dependencies*, 2001.

[53] G. S. Atsalakis and K. P. Valavanis, 'Forecasting stock market short-term trends using a neuro-fuzzy based methodology', *Expert systems with Applications*, vol. 36, no. 7, pp. 10 696–10 707, 2009.

[54] C. Olah. (Sep. 2015). Understanding lstm networks, [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. (accessed: 22.01.2019).

[55] K. Team, *Keras*, https://github.com/keras-team/keras/, 2013.

[56] (2019). Matlab, [Online]. Available: https://uk.mathworks.com/products/matlab.html.

[57] (2019). Tensorflow, [Online]. Available: https://www.tensorflow.org/.

[58] Z. Georgiev and D. Kazakov, 'Learning ordinary differential equations for macroeconomic modelling', in *2015 IEEE Symposium Series on Computational Intelligence*, IEEE, 2015, pp. 905–909.

[59] E. Nakamura, 'Inflation forecasting using a neural network', *Economics Letters*, vol. 86, no. 3, pp. 373–378, 2005.

[60] G. P. Zhang and D. M. Kline, 'Quarterly time-series forecasting with neural networks', *IEEE transactions on neural networks*, vol. 18, no. 6, pp. 1800–1814, 2007.

[61] G. Dong, K. Fataliyev and L. Wang, 'One-step and multi-step ahead stock prediction using backpropagation neural networks', in *2013 9th International Conference on Information, Communications & Signal Processing*, IEEE, 2013, pp. 1–5.

[62] 'Parameter optimization of multilayer perceptron network for stock financial forecasting', [Online]. Available: https://www.cs.york.ac.uk/library/proj_files/2014/3rdyr/gc649/Dissertation.pdf, (accessed: 22.01.2019).

[63] A. Kirilenko, A. S. Kyle, M. Samadi and T. Tuzun, 'The flash crash: High-frequency trading in an electronic market', *The Journal of Finance*, vol. 72, no. 3, pp. 967–998, 2017.

[64] W. H. Kruskal and W. A. Wallis, 'Use of ranks in one-criterion variance analysis', *Journal of the American statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[65] H. B. Mann and D. R. Whitney, 'On a test of whether one of two random variables is stochastically larger than the other', *The annals of mathematical statistics*, pp. 50–60, 1947.

[66] (Jan. 2019). Office for national statistics, Office for National Statistics, [Online]. Available: https://www.ons.gov.uk/. (accessed: 22.01.2019).

[67] (Jan. 2019). Rpi all items, [Online]. Available: https://www.ons.gov.uk/economy/inflationandpriceindices/timeseries/czbh/mm23. (accessed: 01.03.2019).

[68] (Apr. 2019). Interest rates, discount rate for united kingdom, [Online]. Available: https://fred.stlouisfed.org/series/INTDSRGBM193N. (accessed: 21.04.2019).

[69] (Jan. 2019). Sections: B, c, d and e (period on period growth) %:cvm, [Online]. Available: https://www.ons.gov.uk/economy/grossdomesticproductgdp/timeseries/l3bg/pn2. (accessed: 01.03.2019).

[70] (Jan. 2019). Iop: B-e: Production: Cvmsa, [Online]. Available: https://www.ons.gov.uk/economy/economicoutputandproductivity/output/timeseries/k222/diop. (accessed: 01.03.2019).

[71] J. Brownlee, 'How to develop lstm models for time series forecasting', 2018. [Online]. Available: https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/.

[72] J. Brownlee, 'How to develop lstm models for multi-step time series forecasting of household power consumption', Sep. 2018. [Online]. Available: https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/.

[73] J. Brownlee, 'Multivariate time series forecasting with lstms in keras', Oct. 2017. [Online]. Available: https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/.

[74] A. Blumer, A. Ehrenfeucht, D. Haussler and M. K. Warmuth, 'Occam's razor', *Information processing letters*, vol. 24, no. 6, pp. 377–380, 1987.