University of Twente

Dutch Inflation Rate Forecasting Performance of Econometric and Neural Network Models

Thesis Industrial Engineering & Management

Author:
Joost van der Plas (2628902)

Supervisor:
Prof.dr. L. Spierdijk

June 22, 2023

Management Summary

This thesis explores the importance of precise inflation forecasting in business valuation, recognizing the substantial influence of inflation on a company's expected cash flows and risk, thereby shaping its overall value. The research specifically examines the performance of traditional econometric models alongside neural network models, undertaking a comprehensive comparison of their respective forecasting capabilities using the Dutch Consumer Price Index (CPI) as the key inflation measure. In order to anticipate future fluctuations in the general price level of goods and services, this study incorporates several macroeconomic factors as predictors of inflation, including historical inflation rates, money supply, GDP, interest rates, unemployment rates, and the price of gold. By investigating the performance of these models, this research aims to contribute valuable insights for businesses and decision-makers, shedding light on the most effective methods for accurate inflation forecasting and ultimately enhancing the process of business valuation.

The econometric models used to forecast inflation are ARIMA (Autoregressive Integrated Moving Average) and VAR (Vector Autoregression), which are both time series models. ARIMA models use a combination of autoregressive (AR) and moving average (MA) components to model the relationships between a dependent variable and its past values and error terms. VAR models, on the other hand, are used to model the joint behavior of multiple dependent variables. They assume that each variable is linearly dependent on its own past values, as well as the past values of other variables in the system. VAR models are particularly useful for modeling the dynamics of economic systems, where multiple variables influence each other.

The three Neural Networks (NNs) that are investigated are Feedforward Neural Network (FFNN), Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM). A FFNN is a type of neural network where information flows in one direction, from input to output layer, through one or more hidden layers. RNNs work by using feedback connections that allow the output of a layer to be fed back into the input of the same layer, creating a loop that enables the network to remember previous inputs. Long Short-Term Memory (LSTM) is a type of RNN that can handle long-term dependencies by using a memory cell, which allows the network to selectively remember or forget previous inputs. Training these models involves optimizing the weights of the NNs to minimize the difference between predicted and actual values. The process involves an iterative procedure, where the model is fed with training data, the weights are adjusted based on the errors, and the process is repeated until the model reaches an acceptable level of accuracy. The training of FFNNs, RNNs, and LSTMs involves the use of backpropagation algorithm with gradient descent optimization, where the gradients of the error function with respect to the weights are computed and the weights are adjusted accordingly. The process of training neural networks is computationally intensive and requires careful selection of hyperparameters and optimization methods to avoid overfitting.

The findings of this study are presented in Tabel 1 and indicate that the neural network models outperform the econometric models in the in-sample forecasts. Specifically, the LSTM model shows the best in-sample performance, suggesting that it is the most accurate model for predicting inflation based on historical data. However, when it comes to out-of-sample forecasting, the econometric models perform better, indicating that they can better generalize to future data. The RNN and LSTM model also did not perform better than the naive predictor out of sample. The challenging nature of the dataset, which includes the financial crisis, the COVID-19 pandemic and the invasion of Ukraine by Russia, posed significant difficulties for the RNN and LSTM models in their out-of-sample forecast. These extreme events caused sudden and significant shocks to the economy, resulting in rapid changes in market dynamics that the RNN and LSTM models struggled to adjust to. Furthermore, the RNN and LSTM models may not have been able to capture the complex interactions between economic variables during such unprecedented events. It is worth noting that the study highlights that a more complex model does not necessarily result in better and more accurate performance. This is evident from the performance of the RNN and LSTM models, which, despite their complexity, did not perform as well as the simpler econometric models in out-of-sample forecasting.

	RMSE_in	**RMSE_out**	**MAE_in**	**MAE _out**
Naive predictor	0.2812	1.2286	0.2033	0.7526
ARIMA	0.2798	1.1965	0.2062	0.7239
VAR	0.2765	1.1964	0.2076	0.7107
FFNN	0.2760	1.2203	0.2035	0.7449
RNN	0.2413	1.2672	0.1814	0.8195
LSTM	0.2055	1.7197	0.1459	1.2044

Table 1: Performance measure of the Econometric & Neural Network models

Despite the poor performance of the neural networks, their inherent ability to capture complex patterns and relationships within data suggests that they hold great potential for future advancements, highlighting the need for continued research and development to unlock their full capabilities. Future research should explore the use of other machine learning models, examine the performance of the models on different data frequencies, inflation predictors, and forecast horizons, and test the performance of the models with different architectures.

Keywords: Inflation rate, Forecasting, Artificial Neural Networks, Recurrent Neural Networks, LSTM, VAR, ARIMA

List of abbreviations

Adam	Adaptive Moment Estimation
ADF	Augmented Dickey-Fuller
ANN	Artificial neural network
ARIMA	Autoregressive integrated moving average
ARMA	Autoregressive moving average
BGD	Batch Gradient Descent
BPTT	Backpropagation through time
BVAR	Bayesian vector autoregressive
CBS	Centraal Bureau voor de Statistiek
CPI	Consumer Price Index
DCF	Discounted Cash Flow
DSGE	Dynamic stochastic general equilibrium
ECB	European Central Bank
FFNN	Feedforward Neural Network
GD	Gradient Descent
GDP	Gross Domestic Product
GNP	Gross National Product
HICP	Harmonized Index of Consumer Prices
JNN	Jordan neural network
LSTM	Long Short-term Memory
MAE	Mean Absolute Error
MAPE	Mean absolute percentage error
MBGD	Mini-batch Gradient Descent
MLE	Maximum Likelihood Estimation
MSE	Mean squared error
MSLE	Mean squared logarithmic error
OMO	Open market operation
PPI	Producer Price Index
QE	Quantitative Easing
ReLU	Rectified Linear Unit
RMSE	Root mean squared error
RMSFE	Root mean squared forecast error
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
VAR	Vector Autoregressive
WACC	Weighted Average Cost of Capital

Contents

1 Introduction

1.1 Background

One of the primary macroeconomic indicators that the public and private sectors look at is inflation. Inflation is the rate at which the general level of prices for goods and services is rising and subsequently, purchasing power is falling. It is expressed as a percentage change over a specific period of time, usually a year. Inflation is a crucial indicator of an economy's health and is monitored closely by policymakers and economists to determine monetary policy and economic growth prospects. When inflation is high, it can lead to decreased purchasing power, reduced economic growth, and increased interest rates. Conversely, low inflation can indicate weak demand and economic slowdown. In the Netherlands, the inflation rate in 2022 was reported to be 10.0% based on the Dutch Consumer Price Index (CPI). This is significantly higher than the inflation rate of 2.8% in 2021 and 1.3% in 2020, as can be seen in Figure 1. The COVID-19 pandemic has led to disruptions in global supply chains and changes in consumer behavior, which have affected the prices of goods and services. In addition, many central banks in Europe, including the European Central Bank (ECB), have implemented monetary policy measures such as lowering interest rates and engaging in asset purchases in order to provide support to the economy and financial markets during the pandemic. These measures have led to an expansion of the money supply in the United States. According to data from the Federal Reserve, the change in the M2 supply peaked during the pandemic at a year-over-year percent growth rate of 27% in February 2021 (Federal Reserve, 2022). Similarly, the European Central Bank (ECB), the central bank of the European Union, has also implemented a range of monetary policy measures in response to the COVID-19 pandemic, including lowering interest rates and engaging in asset purchases. These measures have contributed to an expansion of the money supply in Europe. According to data from the ECB, the change in the M2 supply peaked during the pandemic at a year-over-year percent growth rate of more than 11% (European Central Bank, 2022).

Another driver of the recent increase of inflation is Russia's invasion of Ukraine. As Russia is Europe largest energy supplier, many countries in Europe are heavily dependent on oil and gas form Russia. The energy prices in Europe have significantly increased as a result of the the political disagreements with Russia, with direct effects on the inflation rate . The Russian invasion of Ukraine has had a number of economic consequences, including an impact on the inflation rate in Europe. Russia is a major supplier of energy, particularly oil and gas, to many countries in Europe, and the political disagreements between the two sides have resulted in increased energy prices in Europe (Dräger et al., 2022). Higher energy prices can contribute to higher inflation, as they increase the cost of producing and distributing goods and services. For example, if the cost of transportation increases due to higher oil prices, this can lead to higher prices for goods that are shipped long distances. Similarly, if the cost of electricity increases due to higher gas prices, this can lead to higher prices for goods that are produced using electricity.

Especially during times of high inflation, accurate forecasting of future inflation rate has become critical for policymakers, businesses, and individuals as it helps them to plan for

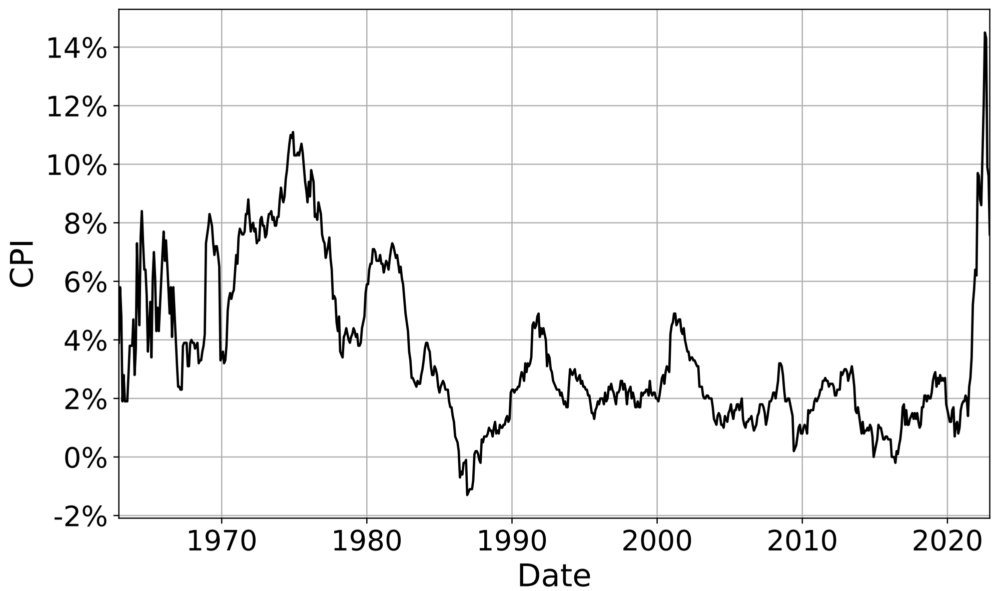

Figure 1: Consumer Price Index (CPI) in the Netherlands (Source: CBS)

the future and make informed decisions. Inflation forecasting is relevant in general as it provides insights into the future economic conditions, the direction of interest rates, and the purchasing power of money. The concept of inflation and the need to measure and forecast it has existed for centuries. The first official consumer price index (CPI) was created in the United States in the late 1800s. It was designed to measure the changes in the cost of living for urban wage earners and clerical workers. Since then, economists began to develop theories and models to explain the causes of inflation and how it could be measured and forecasted. Today, inflation forecasting is a critical tool used by central banks, governments, pension funds, businesses, and investors around the world to guide policy and investment decisions.

1.1.1 Relevance for PwC

The Valuation & Modeling team of PwC Amsterdam is responsible for providing valuation and modeling services to Dutch and international clients. These services may include valuing businesses, intangible assets, financial instruments, and real estate, as well as providing financial modeling support for a variety of purposes, such as forecasting, budgeting, and business planning.

The team works with a variety of clients, including large and small businesses, financial institutions, private equity firms, and other organizations. They use a range of approaches and techniques to value assets and provide financial modeling support, including discounted cash flow (DCF) analysis, comparable company analysis, comparable transaction analysis and real options analysis, among others. In addition to providing valuation and modeling services to clients, the team is also be involved in business development activities, such as building relationships with potential clients and identifying new business opportunities. They are also involved in training and mentoring junior team members and staying up-to-date on the latest developments in valuation and modeling techniques.

Inflation is a key economic indicator that can have significant impact on the valuation and modeling analysis of PwC. Accurate forecasting of inflation rates is therefore crucial for business valuations. While traditional statistical techniques have been widely used for inflation forecasting, recent advances in machine learning, particularly neural networks, offer the potential for improved forecast accuracy.

1.2 Problem context

The inflation rate is an important metric that can significantly impact the value of companies and the financial forecasts of those companies (Ehrhardt and Brigham, 2013; Ross et al., 2003). When the general level of prices is expected to increase, the purchasing power of money is likely to decrease. This is an important consideration when valuing businesses, intangible assets, financial instruments, and real estate, as it can affect the nominal value of these assets. The inflation rate can also affect the required rate of return on investments. When the general level of prices is expected to increase, investors may require a higher nominal return on their investments in order to compensate for the erosion of the purchasing power of money. This is a crucial aspect to keep in mind when offering financial modeling assistance for forecasting, budgeting, and business planning as it can impact the predicted return on investments (Brigham and Ehrhardt, 2013).

In addition, the inflation rate can impact the forecasted cost of goods and services and the forecasted revenue of companies. When the general level of prices is expected to increase, the cost of goods and services is also likely to increase, which can reduce the forecasted profits of businesses. Similarly, when the general level of prices is expected to increase, the prices that businesses can charge for their products and services may also increase, which can increase the forecasted revenue of businesses. These factors are important considerations when providing financial modeling support for forecasting, budgeting, and business planning, as they can impact the financial performance of businesses (Ross et al., 2003).

There are several different types of inflation indices that can be used to measure inflation, such as the Consumer Price Index (CPI), Producer Price Index (PPI), Harmonized Index of Consumer Prices (HICP) and the Gross Domestic Product Deflator (GDP Deflator). In terms of relevance for business valuation, the CPI and PPI are the most relevant. The CPI captures the changes in prices of goods and services consumed by households and the PPI captures the changes in prices of goods and services at the wholesale level. These are the most relevant as they measure prices of goods and services that are most likely to be consumed or used by a company, and therefore have direct implications on the cost of goods and revenue for the company (Ehrhardt and Brigham, 2013; Ross et al., 2003). All together, different types of inflation indices can be used to estimate the expected rate of inflation when valuing businesses. The CPI and PPI are the most relevant for business valuation as they measure changes in prices of goods and services that are most likely to be consumed or used by a company and have direct implications on the cost of goods and revenue for the company.

Inflation has a direct impact on the value of a company as it affects the estimated future cash flows. The discounted cash flow (DCF) method, which is widely used to value investment, calculates the value of the company by discounting the estimated future cash flows. The formula for DCF is the sum of the cash flow in each period divided by one plus the discount rate (WACC) raised to the power of the period number. Inflation reduces the purchasing power of money, meaning that money in the future is worth less than money today. As a result, the present value of each cash flow is reduced by the discount rate, which incorporates the impact of inflation. Additionally, the inflation rate can influence the WACC and the discount rate, which are important factors in the calculation of the company's value. When the inflation rate is expected to increase, the required rate of return on investments is also likely to increase, which can increase the cost of capital for a company and, in turn, increase the WACC (Ehrhardt and Brigham, 2013).

In conclusion, the inflation rate is an important metric that can significantly impact the value of companies and the financial forecasts of those companies. It can affect the purchasing power of money, the required rate of return on investments, the forecasted cost of goods and services, the forecasted revenue of companies, and the WACC and discount rate. Understanding the impact of the inflation rate on these factors is important for accurately valuing companies and making informed business decisions. Accurate forecasting of the inflation rate is therefore important for the Valuation & Modeling team of PwC, as it can impact the accuracy of the valuations and financial forecasts produced by the team. To address the uncertainty surrounding the inflation rate, the team may use a range of approaches and techniques to estimate the impact of the inflation rate on the value of businesses and assets, and may also be involved in conducting research to better understand the factors that drive the inflation rate. This can help to improve the accuracy of the valuations and financial forecasts produced by the team and contribute to better decision-making by their clients (Ehrhardt and Brigham, 2013; Ross et al., 2003).

1.3 The Core Problem

Inflation is a complex phenomenon that is influenced by a wide variety of factors, including economic growth, changes in interest rates, and shifts in the supply and demand for goods and services (Phillips, 1958). As a result, it can be difficult to predict using traditional statistical models. One reason why inflation is hard to forecast is that it is influenced by both short-term and long-term factors. Short-term factors, such as changes in oil prices or weather-related disruptions to agricultural production, can have a significant impact on inflation in the near term (Cecchetti and Moessner, 2008). However, these effects may not persist over the long run. Long-term factors, such as trends in productivity or changes in the demographic makeup of the population, can have a more persistent impact on inflation, but they may be more difficult to measure or predict (Blanchard and Gali, 2007). Another reason why inflation is hard to forecast is that it is affected by structural changes in the economy. For example, technological innovation and globalization have led to increased competition and disinflation in many sectors of the economy (Baldwin, 2018). These structural changes can make it difficult for traditional statistical models, which are based on historical data, to accurately predict future inflation.

Additionally, there are multiple different factors influencing inflation and they are interrelated, so it can be hard to predict how each of these factors will evolve and how they will influence inflation. Also, there might be unexpected events that can affect inflation, such as a pandemic or a geopolitical event, that can influence both the supply and demand and the expectations of inflation, making it hard to forecast. In light of these challenges, many central banks and other organizations use a combination of statistical models and expert judgment to forecast inflation (Fulton and Hubrich, 2021). This can help to incorporate the latest economic data and account for the impact of unexpected events and structural changes on the economy.

Traditional econometric models such as Vector Autoregression (VAR) and Autoregressive Integrated Moving Average (ARIMA) are widely used for inflation forecasting. VAR models take into account the interdependence between multiple macroeconomic variables such as inflation, GDP, and interest rates to make predictions about future inflation. On the other hand, ARIMA models use past inflation data to make predictions about future inflation, considering trends and seasonality in the data. Both VAR and ARIMA models are relatively simple and easy to implement, but their performance can be affected by the choice of variables and the length of the time series data used for analysis. Despite these limitations, traditional econometric models continue to play a significant role in inflation forecasting, especially for medium-term and long-term horizons.

Neural networks, also known as artificial neural networks (ANNs), are a type of machine learning model that can be used to improve inflation forecasting. Throughout the past decades, machine learning models have become more and more popular. Due to the increasing availability of databases and computer power, machine learning models have become a popular method to forecast inflation (Rodríguez-Vargas, 2020). They are particularly well-suited for this task because they are able to learn complex nonlinear relationships between input variables and the output variable such as inflation. One way neural networks can be used to improve inflation forecasting is by incorporating a wide variety of relevant data, including both economic and non-economic data. Machine learning models enable the exploration of the inflation dynamics using a wide variety of macroeconomic data and other financial measurements (Yadav et al., 2019). For example, a neural network model could be trained on a dataset that includes information on interest rates, GDP, oil prices, and even weather patterns (Hochreiter and Schmidhuber, 1997). The ability of neural networks to handle large and complex datasets can be useful for capturing the various short-term and long-term factors that influence inflation.

Another way that neural networks can be used to improve inflation forecasting is by modeling the nonlinear and non-stationary relationships between variables. Traditional statistical models such as ARIMA and VAR are based on linear and stationary assumptions, which are not always met in the real-world (Tsay, 2010; Enders, 2014). Neural networks, however, can model nonlinear and non-stationary relationships which are more likely to capture the true underlying structure of the data. Additionally, Neural network models have been able to incorporate high dimensional time-series data, which enables the capturing of complex dynamics (Vapnik, 1999; Goodfellow et al., 2016), this is especially relevant for forecasting inflation because many factors like oil prices, GDP, and money supply which can be incorporated together in a neural network model.

In summary, inflation is a complex phenomenon that is influenced by multiple factors and is difficult to predict. Central banks and organizations use a combination of statistical models and expert judgment to make inflation forecasts. Traditional econometric models such as VAR and ARIMA are widely used and have been successful, but they have limitations and are based on linear and stationary assumptions. Neural networks, or artificial neural networks (ANNs), have become increasingly popular for inflation forecasting due to the ability to handle large and complex datasets, model nonlinear and non-stationary relationships, and incorporate high dimensional time-series data.

1.4 Research Questions

Inflation forecasting is a critical aspect of economic analysis and decision-making, as it can have significant impacts on a wide range of areas including monetary policy, investment, and financial planning. Accurate inflation forecasting can help policymakers, businesses, and individuals make informed decisions about how to allocate resources and plan for the future.

Traditionally, inflation forecasting has been performed using econometric models, which are statistical models that are used to analyze the relationships between economic variables. These models are based on assumptions about the underlying economic relationships and can be used to make predictions about future values of variables such as inflation. In recent years, there has been growing interest in the use of neural networks as a tool for inflation forecasting. Neural networks are a type of machine learning algorithm that are inspired by the structure and function of the human brain. They consist of multiple layers of interconnected ”neurons” that can process and transmit information. Neural networks have the ability to learn and adapt to new data, and have been widely used in a variety of applications including image and speech recognition, natural language processing, and finance.

The aim of this research is to investigate the inflation rate forecasting performance of traditional and neural network models. The specific objectives of the research are to first review the existing literature on inflation forecasting, including traditional statistical techniques and machine learning approaches. Secondly, explore the use of different neural network architectures and training methods for inflation forecasting. Thereafter, compare the performance of traditional inflation forecasting models with neural network-based models using a range of evaluation metrics. Finally, To identify any factors that may affect the performance of traditional and neural network-based inflation forecasting models, and to suggest strategies for improving their accuracy.

Given the importance of inflation forecasting and the emergence of neural networks as a tool for this purpose, the main research question can be formulated as follows:

How does the forecasting performance of econometric and neural network models compare in predicting Dutch inflation rates?

To address this main research question, the following sub questions will be explored:

- What are the key factors that influence Dutch inflation rate, and how are they captured by econometric and neural network models?
- What are the practical considerations for using econometric and neural network models for inflation rate forecasting in the Dutch context?
- How do the assumptions and limitations of econometric models, as compared to neural network models, impact their forecasting performance?

By answering these sub questions, this thesis will provide a comprehensive analysis of the forecasting performance of econometric and neural network models for Dutch inflation rate forecasting. This will be of value for the PwC valuations & modeling team, as well as, for policymakers, businesses, and individuals seeking to make informed decisions about how to allocate resources and plan for the future.

Furthermore, it is expected that this research will contribute to a better understanding of the potential of neural networks for forecasting inflation rates, and will provide insights into the factors that may affect the performance of such models. The research will also provide a comparison of the performance of neural network-based inflation forecasting models with traditional statistical methods and other machine learning approaches, which will be useful for practitioners seeking to choose the most appropriate forecasting method. Accurate inflation forecasting is important for a range of stakeholders, including individuals, businesses, and governments. The use of neural networks for inflation forecasting has the potential to improve forecast accuracy, which could have significant implications for decision-making and policy-making. This research will therefore contribute to a better understanding of the capabilities and limitations of neural networks for inflation forecasting, and will provide valuable insights for practitioners seeking to use these methods in practice. Regarding the contribution to the existing literature, this research will enrich the existing academic literature by offering a comprehensive comparison between various econometric models and neural networks for inflation rate forecasting. This research will contribute to a better understanding of the strengths and weaknesses of econometric and neural network models in the Dutch context. By exploring the key factors that influence Dutch inflation rates and how they are captured by econometric and neural network models, this research will provide valuable insights into the factors that influence inflation. Finally, the thesis can shed light on the assumptions and limitations of econometric and neural network models and their impact on forecasting performance.

1.5 The Problem Approach

A comprehensive approach to answer the sub questions related to the influence of key factors on Dutch inflation rates, the comparison of econometric and neural network models for inflation rate forecasting, and their performance over different time horizons, can be achieved through the following steps. First, a review of the relevant literature will be conducted to gather information about the key factors that influence Dutch inflation rates and how they are captured by econometric and neural network models. The literature review also includes an overview of various prior studies on inflation forecasting using econometric and neural network models. Next, data on Dutch inflation rates, relevant macroeconomic indicators, and other relevant variables will be collected for a relevant time period. Based on the literature review, econometric and neural network models will then be selected and described. Thereafter the models will be designed and developed to forecast Dutch inflation rates based on the collected data, and their performance will be evaluated based on their ability to capture the key factors that influence inflation rates. The design process of the models is depicted in Figure 2. The performance of the two models will be compared based on various performance metrics such as the mean squared error, mean absolute error, and the impact of their assumptions and limitations will also be analyzed. Finally, the forecasting performance of the models will be analyzed over different time horizons to determine their strengths and weaknesses for different forecasting scenarios. This information will be synthesized to arrive at a conclusion regarding the suitability of econometric and neural network models for inflation rate forecasting in the Dutch context.

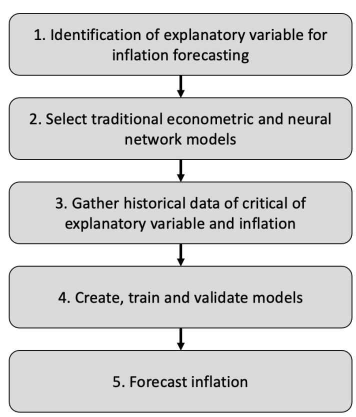

Figure 2: Design process of the inflation rate forecasting models

1.6 Outline of the Report

The structure of this report is organized into six main sections, each of which focuses on a specific aspect of inflation forecasting. The introductory section provides an overview of the report and introduces the problem and the research questions that are addressed. The subsequent section, which is the literature review, delves into the existing research on inflation, discussing measures of inflation, inflation targeting, and the impact of inflation on business valuation. This section also highlights gaps in the literature and reviews previous studies on inflation forecasting. The third section, the methodology, outlines the various methods used for inflation forecasting. This includes traditional econometric models and artificial neural networks. In addition, the section covers the training of (recurrent) neural networks. The fourth section is the empirical approach, which provides specific details on the data used in the study, the performance measures, and the hyperparameters selection process for the neural network models. The fifth section presents the results and discussions of the study, analyzing the findings obtained from both the econometric and neural network models and comparing their respective performances. Finally, the sixth and final section is the conclusion and future research, which summarizes the key findings of the study and offers suggestions for future research on inflation forecasting.

2 Literature Review

2.1 Introduction

The literature review aimed at exploring the impact of inflation on business valuation begins by examining the effects of inflation on the valuation of businesses. This is followed by an overview of the various measures of inflation that are used to track and analyze this economic phenomenon. The next section of the review focuses on identifying the key predictors of inflation and exploring their relationship with inflation. Finally, the literature review provides an overview of prior research on inflation forecasting, including studies that have explored the effectiveness of econometric models and neural network approaches for forecasting inflation. The aim of this literature review is to provide a comprehensive understanding of inflation and its impact on business valuations. By synthesizing the existing research and highlighting the key findings, the literature review will inform the selection of the measure of inflation for this research, the predictors of inflation to be investigated, and the most appropriate econometric and neural network models for forecasting inflation. The findings of this literature review will contribute to the current body of knowledge on inflation and its impact on business valuations and inform future research in this field.

2.2 Measures of Inflation

There are several measures of inflation that are commonly used to track changes in the general level of prices for goods and services. Some of the most widely used measures of inflation include:

1. Consumer Price Index (CPI): The CPI is a measure of the average change in prices of a basket of goods and services consumed by households. It is typically used to track changes in the cost of living and is often used as a benchmark for adjusting wages and other payments (Ehrhardt and Brigham, 2013).

2. Producer Price Index (PPI): The PPI measures the average change in prices received by domestic producers for their output. It is often used to track changes in the cost of raw materials and other inputs used in the production process Ross et al. (2003).

3. Harmonized Index of Consumer Prices (HICP): This index measures the change in the price of a basket of goods and services consumed by households. It is similar to the CPI but it is calculated according to EU standards and it is used for comparing the inflation rate between countries in the European Union (Diewert, 2002).

4. Gross Domestic Product Deflator (GDP Deflator): This index measures the change in the overall level of prices in the economy. It is considered to be a comprehensive measure of inflation as it covers all goods and services produced within a country (Brigham and Ehrhardt, 2013).

Inflation is typically measured as the percentage change in one of these measures over a certain period of time, such as a month, a quarter, or a year. The CPI and PPI are the most relevant for business valuation as they measure changes in prices of goods and services that are most likely to be consumed or used by a company and have direct implications on the cost of goods and revenue for the company (Ehrhardt and Brigham, 2013; Ross et al., 2003). One of the main downsides of using the PPI are that it is an indicator with little public impact, and also that it excludes services, which is one of the largest sectors of the economy in most countries (Auray et al., 2009). The Valuations team of PwC, in general, also uses the CPI as most relevant measure of inflation for their valuation and modelling projects. The CPI is considered by many as the best measure of inflation for investors due to several reasons. First and foremost, CPI reflects consumer spending, which is crucial for investors to know how inflation affects the prices of goods and services that they invest in or consume (Öner, 2012). CPI is a relevant metric for investors since it reflects the prices paid by consumers for everyday items such as groceries, healthcare, transportation, and housing. Another reason CPI is popular among investors is its wide usage and transparency. The government, policymakers, and market participants widely use CPI as a benchmark for measuring inflation (Heenan et al., 2006). This transparency means that the index is easily accessible and regularly updated, making it an important reference point for investors to assess the overall trend of inflation. Lastly, CPI helps inform investment decisions Rashid and Saeed (2017). For investors, understanding inflation is critical to making informed investment decisions. By tracking CPI, investors can better assess the real returns of their investments and adjust their investment strategies accordingly. Investors can also use CPI to assess the relative attractiveness of different asset classes and investment opportunities.

To conclude, the Consumer Price Index is a widely used and transparent measure of inflation that reflects consumer spending, adjusts for changes in the quality and quantity of goods and services purchased over time, and helps inform investment decisions. Therefore, it is a valuable tool for investors to assess the overall trend of inflation and make informed investment decisions. For these reasons the Dutch CPI will be used as measure of inflation for this research.

2.3 Inflation targeting

Central banks around the world have a primary mandate to ensure price stability in their respective economies. To achieve this objective, central banks adopt different monetary policy tools and strategies to manage inflation. This section will explore how central banks target inflation and the various strategies they use to achieve their goals.

Central banks use a variety of tools and strategies to manage inflation. One of the most common approaches is to set an inflation target. An inflation target is the desired level of inflation that a central bank aims to achieve within a specified period. The target can be set as a range or a specific number. For instance, the US Federal Reserve has a long-term inflation target of 2%, while the European Central Bank aims to keep inflation below, but close to 2%. Central banks use different tools to achieve their inflation targets. One such tool is the policy interest rate, which is the rate at which central banks lend money to commercial banks. By adjusting the policy interest rate, central banks can influence the borrowing and lending rates in the economy. When the central bank raises the policy rate, commercial banks tend to raise their lending rates, which reduces borrowing and spending in the economy. This, in turn, lowers demand for goods and services, leading to lower inflation (Cornand and M'baye, 2018).

Another tool that central banks use to manage inflation is open market operations (OMOs). OMOs involve buying or selling government bonds in the open market to adjust the money supply. When the central bank buys bonds, it injects money into the economy, which increases the money supply and lowers interest rates. This encourages borrowing and spending, which can stimulate economic growth and increase inflation. Conversely, when the central bank sells bonds, it reduces the money supply, which increases interest rates and reduces spending, leading to lower inflation. Central banks also use forward guidance to manage inflation expectations. Forward guidance is a communication strategy where central banks provide guidance on the future path of monetary policy. By providing clear and consistent guidance on future policy actions, central banks can influence market expectations and prevent any surprises that could destabilize the economy. For example, if a central bank signals that it will keep interest rates low for an extended period, it can encourage borrowing and spending, leading to higher inflation (Rocheteau et al., 2018).

In addition to these tools, central banks also use unconventional monetary policy measures to manage inflation. Unconventional measures include quantitative easing (QE), where central banks buy long-term government bonds or other assets to lower long-term interest rates. QE can stimulate borrowing and spending and increase inflation by increasing the money supply and encouraging investment Reis (2016). Another unconventional measure is negative interest rates, where central banks set policy rates below zero, effectively charging commercial banks to hold excess reserves. Negative interest rates can encourage banks to lend more, leading to higher borrowing and spending, and increased inflation.

Central banks use different strategies to achieve their inflation targets. One strategy is the Taylor rule, which is a mathematical formula that links interest rates to inflation and output gaps. The Taylor rule suggests that central banks should raise interest rates when inflation is above the target level and lower them when inflation is below the target level. This strategy aims to stabilize inflation and promote economic growth. Another strategy is inflation targeting, where central banks set a specific inflation target and use policy tools to achieve it. Inflation targeting has been widely adopted by central banks worldwide, including the European Central Bank, Reserve Bank of Australia, the Bank of England, and the Bank of Canada. Under an inflation targeting regime, central banks communicate their inflation targets to the public and use various monetary policy tools to achieve them. This approach has been successful in achieving price stability in many countries and has become a popular approach to monetary policy (Mishkin, 2001).

Before central banks started targeting inflation, inflation was generally more volatile and less predictable. Inflation was often driven by supply-side shocks such as fluctuations in oil prices, agricultural output, or geopolitical tensions. These shocks could lead to sudden changes in the price level, making it difficult for businesses and households to plan for the future. However, when central banks began targeting inflation, they became more proactive in managing inflation by adjusting monetary policy to achieve their inflation targets. By using a combination of tools such as interest rates, open market operations, and forward guidance, central banks could influence the level of aggregate demand in the economy and stabilize inflation.

Eickmeier and Hofmann (2022) analyze the behavior of inflation in the United States and the euro area over the past 50 years by estimating indicators of aggregate demand and supply conditions based on a structural factor model. The results suggest that during the Great Inflation of the 1970s, a combination of strong demand and tight supply were at work, while the Volcker disinflation of the early-1980s was driven by the elimination of strong demand. The Global Financial Crisis was characterized by a collapse of demand and a marked tightening in supply, which explains the missing disinflation during the crisis. The most recent observations indicate that the inflation surge since mid-2021 has been driven by a combination of extraordinarily expansionary demand conditions and tight supply in both regions, with a greater role of tight supply conditions in the euro area due to adverse energy supply developments in the wake of the Russia-Ukraine war. The analysis further suggests that tighter monetary policy primarily dampens demand, while financial shocks adversely impact demand and supply in a similar fashion, implying that central banks could bring inflation back down through an appropriate tightening of the monetary policy stance.

In conclusion, central banks have a crucial role in managing inflation in their respective economies. They use various monetary policy tools and strategies to achieve their inflation targets, such as policy interest rates, open market operations, forward guidance, quantitative easing, negative interest rates, and inflation targeting. These tools and strategies aim to stabilize inflation and promote economic growth while avoiding the negative effects of high or unstable inflation. Central banks' ability to manage inflation is critical to ensuring economic stability, promoting sustainable growth, and improving the welfare of their citizens.

2.4 Impact of Inflation on Business Valuation

When assessing the impact of inflation on individual company values, it is important to consider the effect inflation has on expected cash flows/growth and risk. Companies that are less exposed to high and rising inflation are those that have pricing power on their products and services, with low input costs, and operate in a business where investments are short term and reversible. On the risk front, these companies should have a large and stable earnings stream and a light debt load. Historical data has shown that small-cap stocks tend to outperform in decades where inflation is higher than expected, while the value effect, measured as the difference between low price to book and high price to book stocks, was highest in the 1970s when both actual and unexpected inflation were high. The recent experience in 2022 has shown similar patterns, with small-cap and value stocks outperforming in the context of higher than expected inflation (Damodaran, 2022).

Cornell and Gerger (2017) discusses the impact of inflation on business valuation using discounted cash flow models. It explains that the most common way to estimate the continuing value is to assume a steady state growth rate, but that handling inflation correctly is crucial and often done wrong. The paper simplifies the analysis by isolating the two key issues and providing example calculations to show that proper treatment of inflation has a significant impact on valuation, even at low levels. It also highlights that if inflation were to accelerate in the future, the significance of this issue will increase.

The Weighted Average Cost of Capital (WACC) is a widely-used approach in the valuation of companies. The WACC is the average cost of all the capital a company has raised, including both debt and equity. It is typically used to evaluate a company's ability to generate cash flow, and is often used as a discount rate when determining the present value of a company's future cash flows in a discounted cash flow (DCF) analysis. In practice, PwC's valuation and modeling team would use the WACC in a DCF analysis as a discount rate to determine the present value of a company's future cash flows. The WACC is a measure of the overall cost of capital for a company. It is calculated by weighting the cost of each type of capital (e.g. debt, equity, etc.) in proportion to its relative importance in the company's capital structure. The WACC is used to determine the required rate of return that a company needs to earn on its investments in order to provide an acceptable level of return to its shareholders. Inflation is an important factor that can affect the WACC of a company. When the general level of prices is expected to increase, the cost of capital is also likely to increase. This is because the cost of capital is typically expressed in nominal terms, which means that it is denominated in a specific currency and is not adjusted for changes in the purchasing power of that currency. As a result, when the general level of prices increases, the cost of capital will also increase in nominal terms, even if the real cost of capital remains unchanged (Ehrhardt and Brigham, 2013).

There are a few different ways in which the inflation rate can be incorporated into the calculation of the WACC. One way is to use an inflation-adjusted discount rate when calculating the cost of capital. This involves adjusting the discount rate for the expected impact of inflation in order to reflect the real cost of capital. This can be done by using an inflation rate forecast or by making assumptions about the expected increase in the general level of prices. Another way to incorporate the inflation rate into the calculation of the WACC is to use inflation-adjusted cash flows when estimating the value of a company's assets. This involves adjusting the forecasted cash flows for the expected impact of inflation in order to reflect the real value of the cash flows. This can be done by using an inflation rate forecast or by making assumptions about the expected increase in the general level of prices. It is worth noting that the impact of inflation on the WACC will depend on the specific characteristics of the company and its capital structure. For example, companies with a higher proportion of debt in their capital structure may be more sensitive to changes in the cost of debt as a result of inflation, while companies with a higher proportion of equity may be less affected. Additionally, the impact of inflation on the WACC will depend on the expected rate of inflation and the duration of the investments being considered (Ehrhardt and Brigham, 2013).

In discounted cash flow (DCF) valuations, estimating the continuing value is crucial. One common method is to assume that the company reaches a steady state and grows at a constant rate by the terminal horizon. However, incorporating inflation into this calculation can be complex and often done incorrectly, leading to a significant impact on the valuation. Cornell et al. (2021) aim to simplify the analysis by identifying and addressing the two key issues related to inflation and providing example calculations to demonstrate the importance of proper treatment of inflation, even at low levels. As inflation may increase due to current monetary and fiscal policies, the significance of this issue will become even more important.

In conclusion, inflation has a significant impact on business valuation, particularly for estimating continuing value. To mitigate the effects of inflation, companies should focus on increasing prices, improving operations, and reviewing finances. Incorporating inflation into WACC calculations using inflation-adjusted discount rates or cash flows is crucial. Accurately estimating continuing value is also essential for precise DCF valuations.

2.5 Predictors of Inflation

Inflation forecasting involves the use of various economic indicators to predict future changes in the general price level of goods and services. Finding the variables with the highest predictive power is the main challenge in inflation forecasting. According to Rodríguez-Vargas (2020) some of the factors that drive inflation are money supply, interest rate and Gross National Product (GNP). These drivers have predictive power which can be used to forecast future inflation rates. Historical inflation can also provide insight into long-term trends in the rate of inflation and can help to identify patterns that may influence future inflation. Throughout this section various predictions of inflation and their relation to inflation are introduced.

2.5.1 Historical inflation

Historical inflation data can be used to forecast future inflation rates in a number of ways. Some of the main approaches to using historical inflation data for forecasting purposes include:

1. Statistical analysis: Statistical techniques such as regression analysis and time series analysis can be used to identify patterns in historical inflation data and to forecast future inflation rates. These techniques can be useful for identifying trends, seasonal patterns, and other factors that may influence inflation.

2. Econometric models: Econometric models are statistical models that are used to analyze economic data and make forecasts. These models can incorporate a range of economic variables, including historical inflation data, to forecast future inflation rates.

3. Machine learning: Machine learning algorithms, such as neural networks, can be trained on historical inflation data to learn patterns and relationships that can be used to forecast future inflation rates. These approaches can be particularly useful for handling large datasets and for identifying complex, non-linear relationships in the data.

Overall, the accuracy of forecasts made using historical inflation data will depend on the quality of the data, the techniques used for analysis and modeling, and the underlying economic conditions. Figure 3 provides a graph of the behavior of the Dutch CPI over time giving a clear depiction of the rise of inflation in 2022.

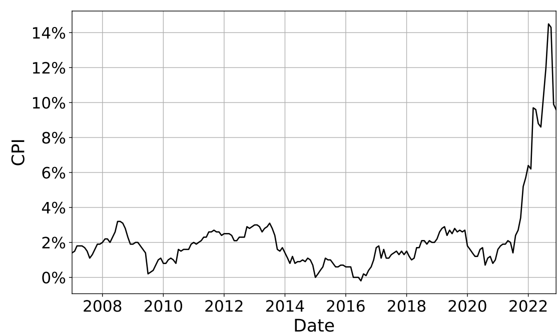

Figure 3: Historical Dutch Consumer Price Index (CPI)

2.5.2 Interest rate

Inflation and interest rates are two important economic indicators that are closely related. Inflation is a measure of the rate at which the general level of prices for goods and services is rising, and is typically measured as the percentage change in a price index over a certain period of time. Interest rates, on the other hand, refer to the cost of borrowing money, and are typically expressed as a percentage of the amount borrowed. Figure 4 depicts the yield curve of the 30 year Dutch Government Bond over time which is a relevant measure of the historical interest rate. In general, a higher interest rate will lead to higher borrowing costs, which can in turn help to reduce inflation by slowing down the demand for goods and services. As can be seen in the figure, the interest rate has increased the past year to slow down inflation. When interest rates are high, consumers and businesses are less likely to borrow money to make purchases, which can help to reduce the overall level of demand in an economy. This can put downward pressure on prices, helping to keep inflation in check. Conversely, a lower interest rate may encourage borrowing and lead to higher demand, which can contribute to higher inflation. When interest rates are low, consumers and businesses may be more willing to borrow money to make purchases, which can increase the overall level of demand in an economy. This can put upward pressure on prices, potentially leading to higher inflation. As a result, central banks and other monetary authorities often use changes in interest rates as a tool to help control inflation and maintain price stability.

By raising or lowering interest rates, they can help to influence the level of demand in an economy and maintain a healthy level of inflation. The Fisher effect, an economic theory that states that the nominal interest rate equals the real interest rate plus the expected inflation rate and named after economist Irving Fisher (1930), highlights the correlation between inflation and interest rates. According to the Fisher effect, changes in inflation expectations will be reflected in changes in nominal interest rates, and monetary policy can be used to control inflation by manipulating nominal interest rates. The Fisher Effect is now considered outdated due to its assumptions of rational expectations, stable relationship between nominal interest rates and inflation, and inability to account for other factors that influence interest rates. Angelina and Nugraha (2020) analyzes the impact of various factors on inflation in Indonesia including interest rate. Their findings indicate that interest rates have a significant and negative effect on inflation.

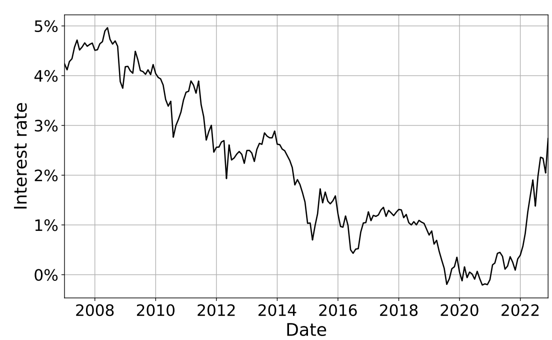

Figure 4: Historical yield curve of the 30 year Dutch Government Bond

2.5.3 Money Supply

The money supply refers to the total amount of money in circulation in an economy, and it is typically measured by central banks and other financial authorities. There are several ways to measure the money supply, and M1, M2, and M3 are three commonly used measures that include different types of money and financial assets.

- M1 is the narrowest measure of the money supply and includes only the most liquid forms of money, such as cash and checking deposits.
- M2 is a broader measure of the money supply that includes M1 plus other assets that are relatively liquid and can be easily converted into cash, such as savings deposits, money market securities, and certificates of deposit.
- M3 is the broadest measure of the money supply and includes M2 plus other less liquid assets, such as institutional money market funds and large-denomination time deposits.

There is generally a positive relationship between the money supply and inflation. When the money supply increases faster than the supply of goods and services in the economy, it can lead to an excess of money chasing a limited number of goods and services, which can put upward pressure on prices (inflation). Conversely, when the money supply grows more slowly than the supply of goods and services, it can help to keep prices in check. Figure 5 provides a graph of the historical M3 money supply of the Euro where 2015 equal to 100. The figure clearly depicts the increase in money supply due to the monetary policy during the COVID-19 pandemic. Ofori et al. (2017) investigates the correlation between money supply and inflation in Ghana. Data from 1967 to 2015 was analyzed using an Ordinary Least Squares model. The study focuses solely on the effect of money supply on inflation, and the Bank of Ghana's role in controlling money supply. The findings indicate a long-term positive relationship between money supply and inflation in Ghana. In the research of Angelina and Nugraha (2020), their findings indicate that money supply and previous money supply have a significant and positive effect on inflation in Indonesia.

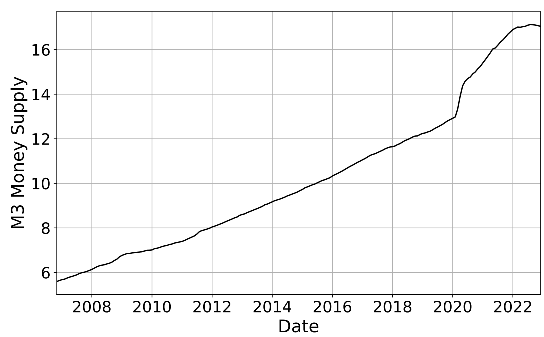

Figure 5: Historical European M3 money supply (in trillions)

The M3 money supply is a crucial measure of the money supply in an economy, and it is particularly relevant for inflation forecasting. M3 includes not only the most liquid forms of money but also less liquid assets such as time deposits and repurchase agreements. Time deposits represent funds held in bank accounts that are not available for immediate withdrawal, while repurchase agreements are agreements between banks and other financial institutions to sell and repurchase securities. M3 is relevant for inflation forecasting because it reflects the total amount of money in circulation in an economy, including both the most liquid forms of money and less liquid assets that are still part of the money supply. Increases in the M3 money supply can indicate a greater availability of credit and spending power, which can lead to inflationary pressures. Conversely, decreases in the M3 money supply can suggest a contraction in credit availability and spending power, which can lead to deflationary pressures. Moreover, M3 is also relevant for assessing the overall health of an economy. An increase in M3 can indicate strong economic growth and investment activity, while a decrease in M3 can suggest a weakening of the economy. Therefore, M3 is a key indicator for policymakers and analysts when it comes to making decisions about monetary policy and predicting future economic conditions.

2.5.4 Gross Domestic Product

GDP and inflation are both key indicators of the health and performance of an economy, and they are often closely watched by policymakers, businesses, and investors. Understanding the relationship between these two variables can help policymakers make informed decisions about monetary and fiscal policy and can also provide insight into the direction and strength of the economy. Figure 6 shows the historical GDP of the Netherlands with a clear dip in 2021 caused by the COVID-19 pandemic. In general, there is a positive relationship between GDP and inflation, which means that when GDP is growing, inflation is likely to rise as well. This is because GDP reflects the overall level of economic activity in an economy, and when economic activity is increasing, it can lead to increased demand for goods and services, labor, and other resources. As firms compete to attract and retain workers and other resources, they may be willing to pay higher wages and prices, which can lead to higher costs and ultimately higher prices for goods and services (inflation). Mallik and Chowdhury (2001) examine whether there is correlation between economic growth and inflation and find that inflation and economic growth are positively related.

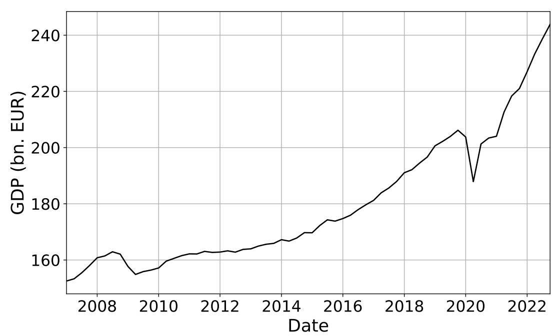

Figure 6: Historical GDP

2.5.5 Unemployment rate

The relationship between inflation and the unemployment rate is a topic of much debate in macroeconomics. The Phillips curve, first introduced by economist Phillips in 1958, suggested that unemployment and inflation are negatively correlated. According to the Phillips curve, when unemployment is low, inflation tends to be high, and vice versa. This relationship was thought to be driven by the fact that when unemployment is low, labor markets are tight, and employers have to compete for workers by offering higher wages. This increase in wages then leads to higher prices and inflation. Conversely, when unemployment is high, there is less competition for workers, and wages tend to be lower, which puts downward pressure on prices and inflation. Figure 7 depicts the historical unemployment rate in the Netherlands with a small peak during the COVID-19 pandemic and a slight increase over the past months. The highest unemployment rate in 2014 was caused by the financial crisis.

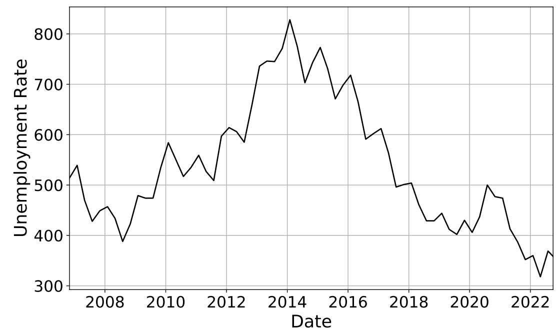

Figure 7: Historical unemployment rate (in thousands)

2.5.6 Gold price

Gold has long been seen as a hedge against inflation. The relationship between gold and inflation is complex and multifaceted, and there are several ways in which gold can be used as a tool to protect against inflation. First, gold is a tangible asset with intrinsic value, meaning that it has value in and of itself, independent of any other factors. This makes it a popular choice for investors who are looking for a way to protect their wealth in the face of inflation. As inflation erodes the purchasing power of paper currency, the value of gold can increase, making it a valuable store of wealth. Second, gold is often seen as a safe haven asset, meaning that it tends to perform well during times of economic and political turmoil. Inflation often occurs during times of economic uncertainty, and as a result, gold can be an effective way to protect against the negative effects of inflation on investment portfolios. Finally, gold can be used as a way to diversify investment portfolios. As inflation can have a negative impact on a wide range of asset classes, including stocks, bonds, and real estate, holding gold can help to balance out the risks in a portfolio and provide a hedge against inflation-related losses (Ghosh et al., 2004). Figure 8 represents the historical XAU/EUR exchange rate which is the spot price between gold and the euro. As can be seen in the figure, during times of uncertainty or high inflation the gold price increases.

Figure 8: Historical gold price

2.5.7 Conclusion

In conclusion, inflation forecasting is crucial in predicting future changes in the general price level of goods and services, and identifying the variables with the highest predictive power is a significant challenge. This section focused on several factors that drive inflation, including historical inflation, interest rate, money supply, GDP, the unemployment rate and gold price. Historical inflation data can be used to forecast future inflation rates using statistical analysis, econometric models, and machine learning. Interest rates and inflation are closely related, and monetary authorities often use changes in interest rates to help control inflation and maintain price stability. The money supply is another crucial factor that influences inflation, and there is generally a positive relationship between the money supply and inflation. GDP and inflation have a negative relationship, meaning that as GDP increases, inflation tends to decrease. Regarding the relation between inflation and the unemployment rate, there is a widely recognized inverse relationship between the two, known as the Phillips curve. Finally, the gold spot price is a predictor of inflation, as it has long been seen as a hedge against inflation. As such, these factors are closely monitored to accurately forecast inflation as is depicted in Figure 9.

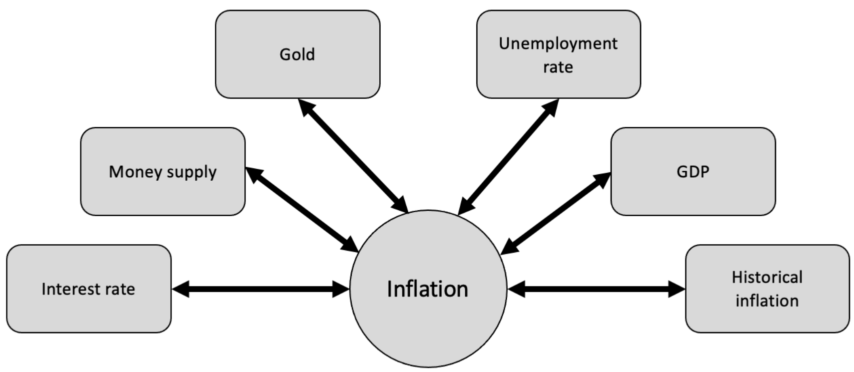

Figure 9: Predictors of inflation

2.6 Prior research and studies on inflation forecasting

The literature on inflation forecasting highlights the use of a wide range of modeling techniques, including univariate autoregressive integrated moving average (ARIMA) and multivariate vector autoregressive (VAR) models. Both methods have their own advantages and disadvantages. On the other hand, VAR models incorporate multiple independent variables, providing a more comprehensive understanding of the economic factors influencing inflation. Univariate forecasting models use a single variable to predict future outcomes while multivariate forecasting models use multiple variables to make predictions. A univariate model only considers the historical data of the variable of interest, while a multivariate model takes into account multiple variables that might affect the outcome. The main advantage of univariate models is its simplicity, while multivariate models are able to consider multiple factors that might affect the outcome. In general, multivariate models tend to provide more accurate predictions than univariate models, but they can be more complex to develop and interpret (Montgomery et al., 2015).

The traditional methodology for forecasting inflation were the Box-Jenkins with moving average linear models (Liu et al., 1992). The Box–Jenkins method applies ARMA or ARIMA models to find the best fit of a time-series model to past values of a time series. Another traditional inflation forecasting model is Vector Auto-regression (VAR) which is a multivariate time series model that incorporates the relationship between multiple economic variables to predict future inflation trends. Central Banks typically use a variety of forecasting models to back their inflation predictions, such as dynamic stochastic general equilibrium (DSGE) models, factor models, vector autoregressive models, and the Phillips curve (Iversen et al., 2016).

The Phillips Curve, introduced by British economist Phillips in 1958, was originally used to forecast inflation by assuming that there is a stable, inverse relationship between unemployment and inflation. In other words, as unemployment decreases, inflation is expected to increase, and vice versa. Based on this relationship, economists could use historical data to estimate the future inflation rate based on current unemployment levels. For example, if the unemployment rate was low, inflation was expected to be high, and vice versa. However, the relationship between unemployment and inflation is not always stable, and it can be influenced by other factors such as changes in technological progress, globalization, and monetary policy. As a result, the use of the Phillips Curve for inflation forecasting has become less popular in recent decades.

Several studies have compared the performance of univariate and multivariate models in forecasting inflation. Fritzer et al. (2002) give a comparison of VAR and ARIMA models for forecasting Austrian HICP inflation over the short term. The evaluation of forecasting performance reveals that VAR models predict the HICP inflation more accurately than ARIMA specifications over a longer forecasting horizon (8 to 12 months ahead). Meyler et al. (1998) also found that ARIMA models are effective in forecasting inflation during periods of relative stability, but may not be as accurate in volatile and high-frequency data. However, it is important to note that univariate models should not be seen as a replacement for multivariate techniques. Papavangjeli (2019) estimated several univariate models to forecast short-term inflation in Albania and found that a Bayesian vector autoregressive (BVAR) model, which incorporates more economic information, outperforms univariate and unrestricted VAR models at different time horizons. (Ogunc et al., 2018) also used a Bayesian VAR model with five variables to examine inflation dynamics in Turkey. Kasuya et al. (2000) similarly found that BVAR models performed better than ordinary VAR models in forecasting Japanese inflation.

Ramakrishnan and Vamvakidis (2002) conducted a study on the transmission effects of domestic and international factors on inflation in Indonesia and found that the exchange rate and foreign inflation had a strong predictive power on domestic inflation, while the impact of base money growth on the headline Consumer Price Index (CPI) was relatively small. Ulke and Ergun (2011) examined the relationship between inflation and import volume in Turkey, using monthly data from 1995 to 2010. They applied various econometric techniques and found that there was a long-term and short-term co-integration relationship between inflation and import volume. Similarly, Muktadir-Al-Mukit et al. (2013) investigated the relationship between inflation and imports in Bangladesh over the period of 2000-2011 using different econometric frameworks. Their results showed that there is a stable, positive, and significant relationship between inflation and imports. Moreover, Khalid (2005) studied the leading factors that influence inflation in Pakistan and found that imported inflation, deficit-GDP ratio, seigniorage, money depth, exchange rate depreciation, and domestic credit are important factors that contribute to inflation in Pakistan. This is consistent with the experience of many emerging economies in their early stages. In addition to these studies, others have also found that the monetary aggregate M3 and bank loans are important variables for forecasting Swiss inflation (Lack et al., 2006). Arsene and Guy-Paulin (2013) examined the relationship between credit to the private sector, inflation, and monetary policy in West Africa and found that the relationship between credit and inflation is non-linear.

Overall, the literature suggests that different factors and models may be more useful in predicting inflation in different countries and contexts. It is important for forecasting to take into account the specific economic conditions and the availability of data for a given country or region.

2.6.1 Comparison between Traditional models and Neural Networks

There are several different types of neural networks that can be used for inflation forecasting, including feedforward neural networks, recurrent neural networks, and convolutional neural networks. Each type of neural network has its own unique characteristics and is well-suited to different types of data and forecasting tasks. One of the key advantages of using neural networks for inflation forecasting is their ability to handle large amounts of data and identify patterns and trends that may not be apparent with traditional methods. Similar to multivariate models, NNs can also be trained to consider a wide range of factors that may influence inflation, such as economic growth, unemployment, and monetary policy. In addition, neural networks are able to capture non-linear relationships between variables, which can be a major limitation of traditional econometric models.

Zhang et al. (1998) claim that Artificial Neural Network (ANN) models in forecasting economic indicators, such as inflation, are more effective than traditional statistical methods. This paper is one of the first to investigate the potential of ANNs for inflation forecasting, and it provides important insights into the advantages of using these models in economic analysis. The paper identifies several characteristics of ANNs that make them particularly useful for forecasting tasks. Firstly, unlike traditional model-based methods, ANNs are data-driven self-adaptive methods. Secondly, ANNs have the ability to generalize, meaning they can make predictions based on patterns in the data, even if the data is not seen before. Thirdly, ANNs are universal functional approximators, meaning they can approximate any function, and finally, ANNs are nonlinear, making them able to handle complex relationships and patterns in the data. Another older paper compared the performance of NNs with traditional inflation forecasting model, Moshiri and Cameron (2000) found that feedforward neural networks (FNNs) were able to predict inflation with a similar level of accuracy to traditional multivariate autoregression models, such as vector autoregression (VAR) and Bayesian vector autoregression (BVAR), over both short-term (3 months) and long-term (12 months) horizons. They found that FNNs were more accurate than traditional models in the particularly short prediction horizon of one month. Šestanović and Arnerić (2021) investigate the ability of a specific type of recurrent neural network, Jordan neural network (JNN), to capture expected inflation compared to commonly used feedforward neural networks and traditional time-series models. They also compare predictions of expected inflation made by survey respondents to predictions made by forecasting models. The paper suggests a strategy for modeling non-stationary time-series data using JNN and finds that it accurately predicts inflation within a 2-year horizon. The results also suggest that JNN predictions of inflation are consistent with predictions made by survey of professional forecasters, making it a useful tool for monetary policy makers.

Furthermore, a study by Binner et al. (2005) compares the forecasting performance of linear models (ARIMA and VAR) and nonlinear models (NN) for Euro inflation. Results suggest that nonlinear models provide better forecasts than linear models, and that the Divisia index, which is an index of money supply, performs better when evaluated in a nonlinear framework. A recent study by Almosova and Andresen (2023) explores the use of nonlinear machine learning techniques, specifically a long short-term memory recurrent neural network (LSTM), in forecasting U.S. consumer price index (CPI) inflation. Results show that LSTM outperforms linear models like AR, RW, SARIMA, MS-AR and simple NN, in terms of the root mean squared forecast error (RMSFE) at all horizons. The study also found that LSTM rolling-window real-time forecasts are more accurate than those of AR and NN. The paper also examines the sensitivity of the model to hyper-parameters and provides an interpretation of what the network learns. The conclusion is that LSTM's good performance is due to its ability to capture nonlinearities in the data and its flexible architecture. Işığıçok et al. (2020) discuss the importance of forecasting inflation rates in the economy, and compares the performance of two different techniques for predicting future inflation rates. The study uses consumer price index (CPI) data from January 2002 to March 2019 to forecast 9-month inflation rates in April-December 2019 using both Box-Jenkins (ARIMA) models and Artificial Neural Networks (ANN). The study aims to determine which technique performs better in predicting future inflation rates based on statistical and econometric criteria. The results show that both techniques provided similar results and were relatively close to each other.

2.7 Conclusion

In conclusion, the literature research has shown that the impact of inflation on business valuation is significant and should be handled correctly, making accurate inflation forecasting extremely relevant. Inflation forecasting involves the use of various economic indicators to predict future changes in the general price level of goods and services. Finding the variables with the highest predictive power is the main challenge in inflation forecasting. The predictors used to forecast the inflation are the historical inflation, interest rate, money supply (M3), GDP, unemployment rate and gold price. There are several different types of models that can be used for inflation forecasting, such as the more traditional econometric models, as well as, Neural Networks. Both traditional econometric models and neural networks have their own strengths and limitations when it comes to inflation forecasting. Econometric models are well-established and have a long history of successful use, but can be limited by their assumption of linear relationships and their reliance on large amounts of data. Neural networks, on the other hand, have the ability to handle large amounts of data and capture complex and non linear relationships, but can be sensitive to the initial conditions and hyperparameters used. The econometric models that are investigated in this research are the VAR and ARIMA model and for the Neural Networks the Feedforward Neural Network, Recurrent Neural Network and LSTM. The performance of these models will be analyzed and compared with each other. The measure of inflation used to train and validate the models is the Dutch Consumer Price Index (CPI).

3 Methodology

3.1 Traditional inflation forecasting models

This section explores traditional econometric models for inflation forecasting. There are several traditional econometric models that are widely used for inflation forecasting, including the Philips curve, VAR (Vector Autoregression), ARMA (Autoregressive Moving Average), and ARIMA (Autoregressive Integrated Moving Average). These models have been widely applied and studied for decades and are considered the backbone of inflation forecasting. In this section, we will explore the different traditional econometric models and how they can be applied to inflation forecasting.

3.1.1 The Autoregressive Moving Average (ARMA) model

The autoregressive moving average (ARMA) model was first introduced in the 1950s by the statistician Whittle (1951), who developed it as a method for analyzing and forecasting time series data. The ARMA model is a combination of an autoregressive (AR) model, which uses past values of a time series to predict future values, and a moving average (MA) model, which uses the residual errors from a prediction model to generate a forecast. While Whittle is credited with the original development of the ARMA model, Box and Jenkins (1970) made significant contributions to the development and application of ARMA models in the 1960s and 1970s. They published several influential papers on the topic and developed a systematic approach to modeling and forecasting time series data using ARMA models and related techniques, which is known as the "Box-Jenkins methodology". This approach has become widely used in the field of statistical analysis and is still used today as a standard method for modeling and forecasting time series data.

To create an ARMA model, you would first need to determine the appropriate values for the p (the number of autoregressive terms) and q (the number of moving average terms) parameters, which determine the order of the model. Once you have determined the appropriate values for p and q, you can use the following equation to create an ARMA(p,q) model:

$$y_t = \underbrace{\sum_{i=1}^{p} \phi_i y_{t-i}}_{\text{AR}} + \underbrace{\sum_{i=1}^{q} \theta_i \epsilon_{t-i}}_{\text{MA}} + \epsilon_t \tag{1}$$

where:

y_t : the inflation rate at time t

ϕ_i : the autoregression coefficients for the AR component

θ_i : the moving average coefficients for the MA component

ϵ_t : the residual error at time t

p : the order of the autoregression component

q : the order of the moving average component

3.1.2 Autoregressive Integrated Moving Average (ARIMA) model

An Autoregressive Integrated Moving Average (ARIMA) model is a statistical model that is used to analyze and forecast time series data. It is an extension of the Autoregressive Moving Average (ARMA) model that can also handle non-stationary data. The model is composed of three components: the autoregression (AR) component, the differencing (I) component, and the moving average (MA) component. The autoregression component of an ARIMA model captures the dependence between an observation and a number of lagged observations. The mathematical representation of an autoregression model of order p (AR(p)) is given by the following equation:

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t \tag{2}$$

The moving average component captures the dependence between the residual errors of the time series at different times. The mathematical representation of a moving average model of order q (MA(q)) is given by the following equation:

$$y_t = \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \tag{3}$$

The differencing component, also known as the integration component, is used to remove non-stationarity from the data. This is done by taking the difference between consecutive observations. The mathematical representation of a differencing component of order d (I(d)) is given by the following equation:

$$\Delta^d y_t = y_t - y_{t-d} \tag{4}$$

The ARIMA(p,d,q) model is a combination of the Autoregression (AR), Difference (I), and Moving Average (MA) components. The general form of the ARIMA model can be rewritten as:

$$y_t = \sum_{i=1}^{p} \phi_i \Delta^d y_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \tag{5}$$

where:

$$\begin{aligned}
y_t &: \text{the inflation rate at time } t \\
\phi_i &: \text{the autoregression coefficients for the AR component} \\
\theta_i &: \text{the moving average coefficients for the MA component} \\
\epsilon_t &: \text{the residual error at time } t \\
\Delta^d &: \text{the difference operator applied } d \text{ times to the time series data} \\
p &: \text{the order of the autoregression component} \\
d &: \text{the order of the differencing component} \\
q &: \text{the order of the moving average component}
\end{aligned}$$

In summary, an ARIMA model is a time series forecasting method that can handle both stationary and non-stationary data by using a combination of autoregression, differencing, and moving average components. The model is represented by the notation ARIMA(p,d,q), where p is the order of the autoregression component, d is the order of the differencing component, and q is the order of the moving average component.

3.1.3 Vector Autoregressive (VAR) model

Vector autoregressive (VAR) models are a class of statistical models that are often used to forecast time series data, including inflation rates. A VAR model consists of a set of variables that are assumed to be related to one another through a set of linear equations. To forecast inflation using a VAR model, we would first need to identify the variables that are likely to influence inflation and include them in the model. These could include economic indicators such as GDP growth, unemployment rate, and interest rates, as well as other factors such as commodity prices and exchange rates (Lütkepohl, 2005). Once the variables have been selected, we would estimate the VAR model by fitting the model to a dataset of historical data for these variables. This typically involves estimating the coefficients of the linear equations that describe the relationships between the variables, as well as any error terms (Enders, 2014; Hamilton, 1994).

Let y_t be a K x T matrix of K variables observed over T time periods. Each row of y represents the time series of a particular variable, and each column represents a particular time period. The VAR model can be represented as:

$$y_t = A_1 y_{t-1} + A_2 y_{t-2} + ... + A_p y_{t-p} + \epsilon_t \tag{6}$$

where $A_1, A_2, ..., A_p$ are K x K matrices of coefficients, $y_{t-1}, y_{t-2}, ..., y_{t-p}$ are the lagged values of y_t, and ϵ_t is a K x T matrix of error terms. The coefficients in the matrices $A_1, A_2, ..., A_p$ are estimated using regression analysis, typically using a method called maximum likelihood estimation (MLE). This allows us to estimate the relationship between the variables and use this relationship to make predictions about future values of the variables based on their past values.

3.2 Artificial Neural Networks

This section gives an introduction to Artificial Neural Networks (ANNs), also known as Neural Networks (NNs), and their ability to forecast time series such as inflation. Neural networks are a type of machine learning algorithm inspired by the structure and function of the human brain. They consist of interconnected artificial neurons that are organized into layers and are trained to perform a specific task by adjusting the strengths of the connections between the neurons, also known as the weights. In general, a neural network consists out of an input layer, a number of hidden layers and an output layer. Neural networks are particularly powerful at identifying patterns and relationships in complex and high-dimensional data. One application of neural networks is in financial time series forecasting, where the goal is to predict future values of financial variables such as stock prices, exchange rates, and inflation rates. Neural networks have been shown to be effective at financial time series forecasting because they are able to capture non-linear and dynamic dependencies in the data, and can handle large amounts of historical data.

3.2.1 Feedforward Neural Networks

The most basic neural network is the Feedforward Neural Network (FFNN) and is also known as a multi-layer perceptron (MLP). This is a type of neural network that consists of an input layer, one or more hidden layers, and an output layer. The network processes the input data by passing it through the different layers in a feedforward manner, without any loops or cycles. This means that the information flows in one direction, from the input layer to the output layer, without any feedback or recursion. Figure 10 depicts a feedforward neural network with a single hidden layer. However, it's worth noting that it is also possible to incorporate multiple hidden layers within the structure. When a neural network comprises of more than one hidden layer, it is referred to as a "deep" neural network. The incorporation of multiple hidden layers allows the network to learn more complex patterns and dependencies in the data, thereby potentially enhancing its performance.

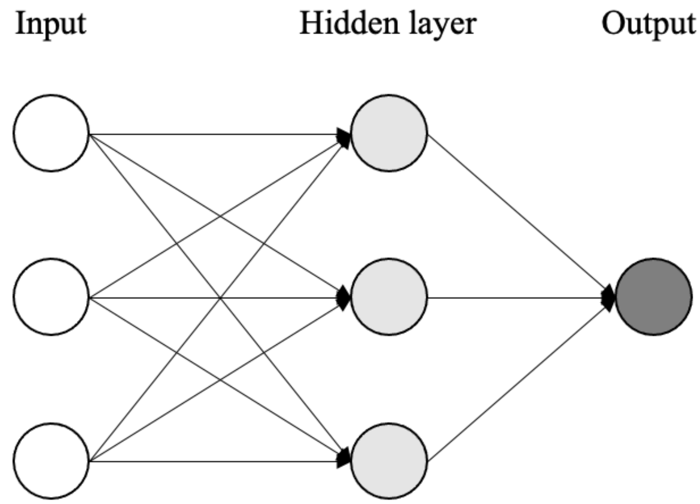

Figure 10: Feedforward Neural Network

A zoomed in node of the neural network is depicted in Figure 11 giving a clear visualization of the structure of a neuron. Let $x \in \mathbb{R}^n$ be the input vector with n the number of input features, denoted as follows $x = [x_1, x_2, \ldots, x_n]^\top$. For example, the two input features could be the inflation rate and the money supply of the previous time period. The bias b_q for node q acts in a similar fashion to the intercept as in a regression model. $W \in \mathbb{R}^{q\text{x}n}$ is the weight matrix associated with the inputs, q refers to the number of neurons in the next layer. The output y_q is produced by passing the weighted sum of the inputs together with the bias through the activation function (f). The output y_q, as in Equation 7, is the input for the next layer until it reaches the output layer which produces the final output $\hat{y}$.

$$y_q = f(W_q x + b_q) \tag{7}$$

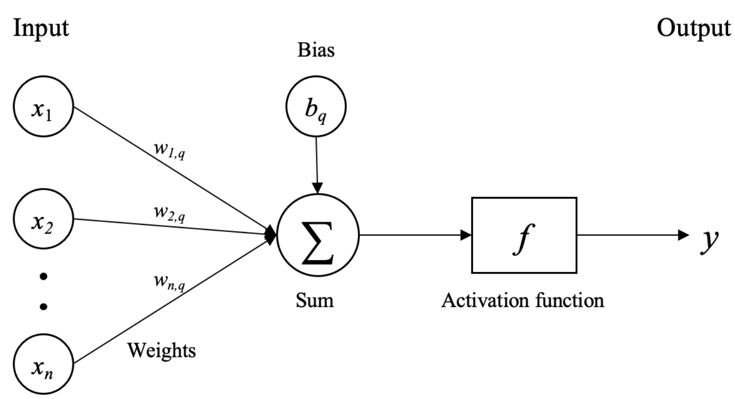

Figure 11: Structure of a artificial neuron

This example only considers one of the neurons in the layer. To compute y_q, row q of the weights matrix (W), corresponding to the weights associated with neuron q, is multiplied with the input vector (x). Then the bias is added and the activation function is applied. However, when an entire network is taken into account the computation becomes very large and complex. To simplify the function above, bias b can be considered as a weight that is always multiplied with a 'dummy' input value of 1. The simplified version for an entire layer can be written as follows:

$$y = f(Wx) \quad \text{where} \quad W = \begin{bmatrix} b_1, w_{1,1}, w_{2,1}, \ldots, w_{n,1} \\ b_2, w_{1,2}, w_{2,2} \ldots, w_{n,2} \\ \vdots \quad \vdots \qquad\qquad\quad \vdots \\ b_q, w_{1,q}, w_{2,q} \ldots, w_{n,q} \end{bmatrix} \quad \text{and} \quad x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{8}$$

An activation function, represented mathematically as f, plays a crucial role in determining the output of a node or multiple nodes in a layer of a neural network. The activation function transforms the weighted sum of the inputs into the final output. There are several commonly used activation functions, including the linear function, the sigmoid (or logistic) function, the hyperbolic tangent (tanh) function, and the Rectified Linear Unit (ReLU) function. These functions are expressed mathematically as follows:

$$\text{Linear}: \quad f(x) = x$$

$$\text{Sigmoid}: \quad f(x) = \frac{1}{1+e^{-x}}$$

$$\text{tanh}: \quad f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\text{ReLU}: \quad f(x) = \max(0, x)$$

The activation function can be either a linear function, known as the 'identity function' or 'no activation', where the input is equal to the output, or a non-linear function such as the sigmoid, tanh, or ReLU functions. The sigmoid function has a range of 0 to 1 and is often used in the hidden layer to mimic the behavior of biological neurons. The tanh function has a range of -1 to 1 and is also used in the hidden layer. For classification problems, the ReLU function is more commonly used in the output layer, while a linear function is preferred for regression problems. The choice of activation function is a crucial factor in the design of an Artificial Neural Network (ANN). If all the activation functions were linear, the ANN would be reduced to a linear regression model, and the non-linear functions such as the sigmoid and tanh enable the ANN to discover complex relationships between targets and features. With linear activation function, also known as 'no activation' or identity function', the input is equal to the output. The sigmoid and tanh function are mostly used in the hidden layer mimicking the behavior of biological neurons. The sigmoid function has a range of 0 to 1, and the tanh function has a range of -1 to 1. For classification problems, the ReLU function is more commonly used in the output layer, whereas a linear function is preferred for regression problems. The selection of activation functions is critical in the architecture of an ANN. It is worth noting that if all of the activation functions were linear, the ANN would be reduced to a linear regression model, whereas non-linear functions such as the sigmoid and tanh functions enable ANNs to discover complex functional relationships that may exist between targets and features (Sharma et al., 2017).

3.3 Training Artificial Neural Networks

Artificial Neural Networks (ANNs) were inspired by the human brain and both learn from experience and adjust themselves in response to mistakes. This section aims to provide an intuitive understanding of how ANNs are trained. The goal of the training algorithm is to optimize the performance of the neural network by adjusting the weights to minimize the observed error. To enable the model to learn from its mistakes, a loss function is introduced. This loss function calculates the difference or error between the actual value and the predicted value at a given moment (Gurney, 2018). The Mean Squared Error (MSE) function is commonly used for regression problems such as inflation forecasting. The MSE formula takes the squared difference between the predicted and actual value 9.

$$L(y, \hat{y}) = \frac{1}{2} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{9}$$

Where n is the number of observations in the sample, y_i is the observed value and $\hat{y}_i$ the predicted value. The neural network is trained using a sample set, which is a subset of the entire dataset. The goal of the training process is to start with a poorly performing network and to improve the accuracy of the predictions by adjusting the weights and bias. The training process begins by initializing the weights and biases of the network with random values, resulting in a poorly performing network. As the training process progresses, the accuracy of the predicted output produced by the model improves by adjusting the weights and bias (Alpaydin, 2020).

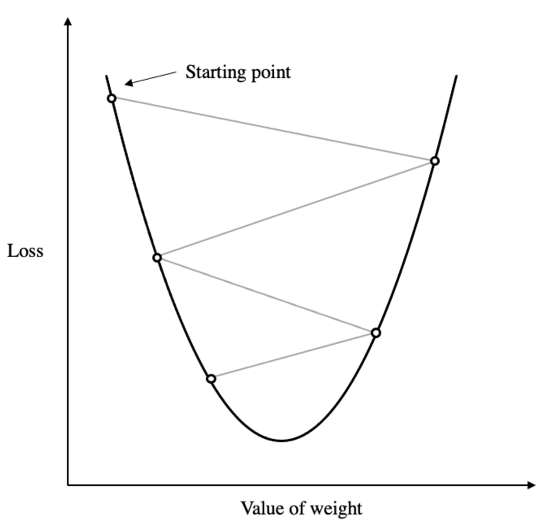

Figure 12: Gradient descent with large learning rate

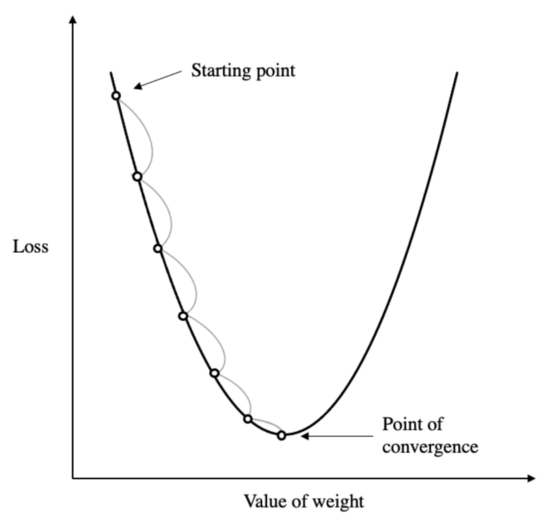

Figure 13: Gradient descent with small learning rate

The optimization of NNs is a crucial step in achieving a high-performing model, and one of the most widely used methods for this purpose is gradient descent. Gradient descent is an optimization algorithm that is used to find the local minimum of a differentiable function. The algorithm works by determining the derivative or slope of a function at a certain point, and then adjusting the input value in order to move closer to the local minimum. For instance consider Figure 12 & 13, if the derivative (g'(x)) at a certain point is greater than 0, the input value (x) is reduced to move closer to the local minimum. Conversely, if the derivative is less than 0, the input value is increased to move closer to the local minimum. The learning rate (η), which is a value between 0 and 1, represents the size of the steps taken to reach the local minimum. In general, a small learning rate is used, as a high learning rate results in larger steps and faster convergence but less accurate results (Figure 12), while a smaller learning rate results in small step sizes, slower convergence but more accurate results (Figure 13).

However, determining the gradient for nodes in hidden layers of the network, which are one or more steps away from the output layer, can be challenging. As the neural networks becomes larger and more complex, computing all partial derivatives is a very slow operation. To address this issue, an efficient algorithm called backpropagation is used in combination with gradient descent. Backpropagation was first introduced in the early 1970s and gained widespread recognition after the publication of a paper by Rumelhart et al. (1985). Backpropagation is a supervised learning algorithm used to calculate the gradients of the loss function with respect to the weights of the network. The gradients are calculated by working backwards through the layers of the network, starting at the output and moving towards the input. The algorithm uses the chain rule to calculate the gradients, and it stores the partial derivatives as it moves back through the layers. Once the gradients have been calculated, they are used by the gradient descent algorithm to adjust the weights of the network. The gradient descent algorithm updates the weights of the network in the opposite direction of the gradients, which moves the network towards a lower value of the loss function (Goodfellow et al., 2016).

The process of backpropagation and gradient descent is repeated multiple times, with each iteration referred to as an epoch. During each epoch, the backpropagation algorithm is used to calculate the gradients and the gradient descent algorithm is used to update the weights. This process is repeated until the loss function is minimized, and the network has learned to produce accurate predictions. The combination of backpropagation and gradient descent allows for efficient training of neural networks by providing a way to calculate the gradients and adjust the weights in a way that leads to a lower value of the loss function (Nielsen, 2015).

The above example just considers one independent variable, x. Now take the network depicted in Figure 10 as example, there is a weight matrix between the input and the hidden layer and one between the hidden and output layer, W_x and W_y respectively. These matrices are updated using gradient descent with backpropagation to optimize the model. After each iteration the weights of the NN are updated as follows:

$$W_j := W_j + \Delta W_j \tag{10}$$

The magnitude and direction of the weight update is computed by taking a step in the opposite direction of the cost gradient. Backpropagation calculates the loss with respect to each individual weight separately, $\frac{\partial L}{\partial w_{nq}}$ using the chain rule. However, it is very inefficient to do this for every weight separately. Therefore, the loss with respect to the weight matrix can be written as in Function 11, where the gradient is calculated with respect to each weight in the matrix.

$$\Delta W_j = -\eta \frac{\partial L}{\partial W_j} \tag{11}$$

In Equation 11, L is the loss function, η is the learning rate and W is the weight matrix corresponding to layer j. In each iteration, the weight vector between every layer is updated. For the network depicted in Figure 10 the the gradient with respect to the weight matrix $W_x \in \mathbb{R}^{q\text{x}n}$ and $W_y \in \mathbb{R}^{k\text{x}q}$ needs to be calculated to improve the performance of the model. n is the number of features or inputs, q is the number of neurons in the hidden layer and k is the number of outputs. The equations in the neural network are:

$$h = f(W_x x) \tag{12}$$
$$\hat{y} = f(W_y h) \tag{13}$$

The state of the hidden layer (h) is calculated by multuplying the imput vector $x \in \mathbb{R}^n$ with the weight matrix W_x. To compute output $\hat{y}$, the hidden layer is multiplied with weight matrix W_y. Now the gradients can be calculated as follows:

$$\frac{\partial L}{\partial W_x} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h} \frac{\partial h}{\partial W_x} \tag{14}$$

$$\frac{\partial L}{\partial W_y} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W_y} \tag{15}$$

These equations represent the backpropagation algorithm used in artificial neural networks to calculate the gradient of the loss function (L) with respect to the weights (W) in the network. The first equation calculates the gradient of the loss with respect to the input layer weights (W_x), while the second equation calculates the gradient of the loss with respect to the output layer weights (W_y). The equations use the chain rule to calculate the gradients by breaking down the computation into smaller parts involving the partial derivatives of the output layer (y), the hidden layer (h), and the weights in the network.

Even though the gradient descent method is the most common for the optimization of neural networks, there are some drawbacks. Gradient descent aims to find the global minimum which is where the performance of the model is optimal. For a convex function, such as in Figure 12 & 13, the global minimum is easily found. However, with nonconvex functions the gradient descent can struggle to find the global minimum as there is a possibility to end up in a local minimum or saddle point. This happens when the gradient of the loss function is at or close to zero.

The are several types of gradient descent algorithms, each having their own advantages and disadvantages. Firstly, Gradient Descent (GD) is a widely used optimization algorithm that calculates the gradients using the entire dataset. While it is considered the most standard and traditional optimization algorithm, it can be computationally expensive and slow for large datasets. As an alternative, Batch Gradient Descent (BGD) can be used, which is similar to GD but uses a fixed batch size for each iteration. While it is also computationally expensive for large datasets, it is less noisy than GD and can converge more quickly to the optimal solution. Another optimization algorithm is Stochastic Gradient Descent (SGD) which uses a single sample from the dataset to calculate the gradients, making it faster than GD and BGD. However, the gradients are more noisy and it may converge to a local minimum instead of global minimum. Lastly, Mini-batch Gradient Descent (MBGD) is an optimization algorithm that uses a small subset of the dataset (a mini-batch) to calculate the gradients. It is a trade-off between SGD and BGD, it is faster than BGD but less noisy than SGD, it can converge more quickly to the optimal solution than SGD and it is computationally less expensive than BGD.

In summary, neural networks are trained using the backpropagation algorithm, a supervised learning algorithm that adjusts the weights of the network in order to minimize the error between the predicted output and the actual output. The backpropagation algorithm calculates the gradients of the error with respect to the weights using the chain rule of calculus and updates the weights in the opposite direction of the gradient. The choice of which optimization algorithm to use depends on the size of the dataset and the desired trade-off between accuracy and computational efficiency.

3.3.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) were first introduced by psychologist Elman in the 1990s as a solution to the limitations of Feedforward Neural Networks (FFNNs) in processing sequential data, such as time series, text, and speech. Unlike FFNNs, which analyze each set of input variables in isolation with no knowledge of prior inputs and ignore the sequential order of features within each sample, RNNs have a "memory" that allows them to retain information from previous time steps. This creates a loop where hidden layers receive data from both previous time steps and current inputs, enabling the network to learn patterns and dependencies across the entire input sequence Dematos et al. (1996). The basic building block of an RNN is the recurrent neuron, which has a hidden state that is updated at each time step and encodes the information learned from previous time steps. The input to the network at each time step is a vector of real numbers and the output is also a vector of real numbers. The term "recurrent" refers to the fact that the hidden layer receives information from previous time steps, making RNNs particularly useful for forecasting financial prices, as this data is often time dependent. By being able to learn the time dependency of the data as well as the data itself, RNNs are a suitable solution for addressing the shortcomings of FFNNs in processing sequential data.

Recurrent Neural Networks (RNNs) are a type of neural network specifically designed to process sequential data, such as time series, text, and speech. Unlike feedforward neural networks (FFNNs), which analyze each set of input variables in isolation with no knowledge of prior inputs and ignore the sequential order of features within each sample, RNNs have

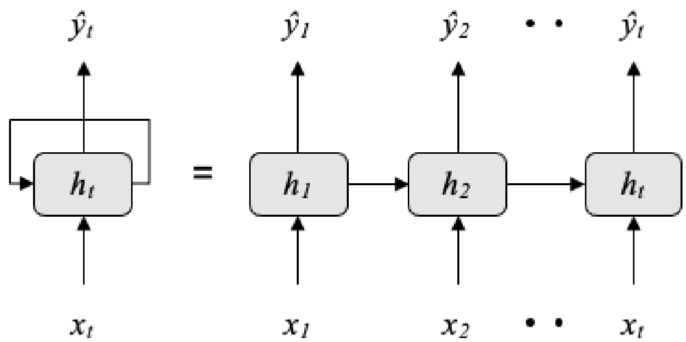

Figure 14: Recurrent Neural Network

a "memory" that allows them to retain information from previous time steps. Figure 14 depicts an unfolded RNN where the input at time t, x_t, represents the number of features (n) and lags (s) and is a vector of real numbers. The output, y_t, also a vector of real numbers, has the size of $y_t \in \mathbb{R}^k$, where $k = 1$ for a single output value. The hidden state h_t in between consists of q number of neurons, often referred to as units, and the first time step, $t = 1$, is initialized with h_0. The weight matrix between the input and hidden layer is $W_x \in \mathbb{R}^{qxn}$, the weight matrix between the previous and current hidden layer is $W_h \in \mathbb{R}^{qxq}$, and the weight matrix between the hidden and the output layer is $W_y \in \mathbb{R}^{kxq}$. The activation function, f, and the bias associated with each neuron q in the hidden layer, b_q, are also included in the calculation of the output, which is given by the following equation:

$$\hat{y}_t = f(W_y h_t + b_q) \tag{16}$$

Where,

$$h_t = f(W_h h_{t-1} + W_x x_t + b_h) \tag{17}$$

The Recurrent Neural Network (RNN) depicted in Figure 14 operates in a many-to-one manner, aggregating multiple inputs from prior time steps to produce a single output. This structure is commonly employed in sequence-related applications, such as predicting inflation. Other popular variants of RNNs include one-to-many and many-to-many architectures.

3.3.2 Long Short-term Memory

Long Short-Term Memory networks, commonly referred to as LSTMs, are a variant of Recurrent Neural Networks (RNNs) that are specifically designed to learn long-term dependencies. They were first introduced by Hochreiter and Schmidhuber (1997) and have undergone significant advancements and improvements over the years. LSTMs are able to overcome the problem of vanishing gradients, which is a common issue in traditional RNNs, that makes it difficult for them to learn long-term dependencies. LSTMs have a structure similar to that of RNNs, but with a crucial difference in their internal cell. Instead of a single neural layer, an LSTM cell comprises of four layers that interact with each other in a unique way. These layers are responsible for maintaining and updating the cell's internal state, which enables the network to remember information over longer periods of time. The remainder of this section provides a step-by-step explanation of how an LSTM network works, highlighting the role of each of these layers and how they interact to enable the network to learn long-term dependencies.

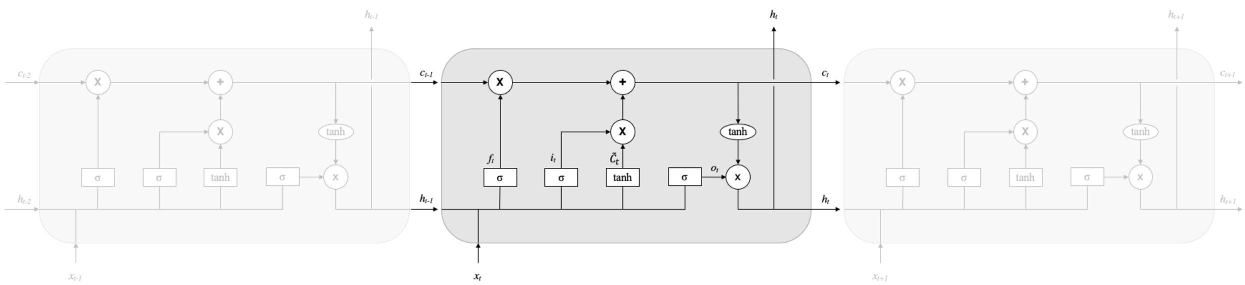

Figure 15: LSTM cell

Figure 15 illustrates a single LSTM cell, where a hidden layer consists out of multiple LSTM cells. The square boxes within the cell represent the four trained neural network layers and the circles indicate pointwise operations such as vector multiplication or addition. The LSTM cell consists out of three gates: the forget gate, the input gate and the output gate. The combination of these three gates and the cell state, which is the horizontal line at the top of the cell in figure 16, gives this neural network the ability to learn long-term dependencies. The square boxes within the cell are the four trained neural network layers and the circles are pointwise operations such as a vector multiplication or addition.

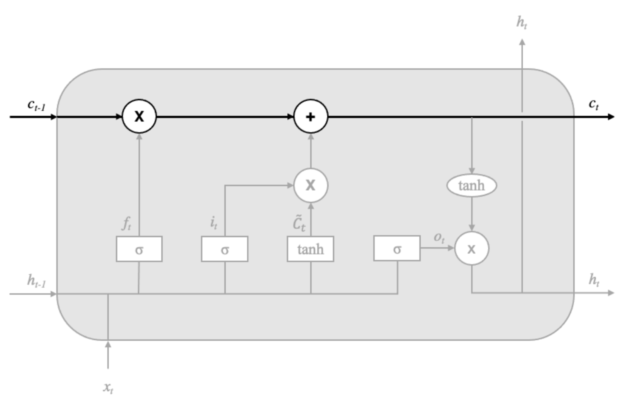

Figure 16: LSTM cell state c_t

The horizontal line running through the top of Figure 16 is the cell state (c_t) which can be seen as a transport line through the linked LSTM cells. The presence of the cell state, or long term memory state, is the reason that LSTMs do not suffer form exploding or vanishing gradients (Graves, 2012). On the other hand, the hidden state (h_t) at the bottom of the cell, takes into account the most recent information and is known as the short term memory state. Through the three regulated gates, the LSTM cell has the ability to remember or forget (ir)relevant information. These three gates are known as the forget, input and output gate. The inputs and outputs of the gates are the following:

$$\begin{aligned} n &: \text{number of input features} \\ q &: \text{number of neurons in the LSTM cell} \\ x_t \in \mathbb{R}^n &: \text{input current cell} \\ h_t \in \mathbb{R}^q &: \text{hidden state input and output} \\ c_t \in \mathbb{R}^q &: \text{cell state input and output} \\ f_t, i_t, o_i \in \mathbb{R}^q &: \text{forget, input and output gate} \end{aligned}$$

The computations in each gate are explained in the remainder of this section. First the equations corresponding to each layer are introduced and at the end of the section the size shape of the variables are explained in detail. As well as the input and output shape of the LSTM.

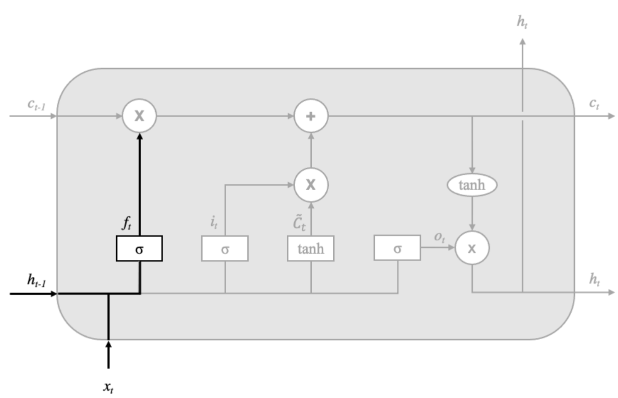

Figure 17: LSTM forget gate

The forget gate is the first gate in the LSTM cell, it is responsible for deciding which information from the previous cell state can be forgotten. The forget gate is highlighted in Figure 17. The output of the forget gate is passed on to the current cell state. Before the output is passed on, it is processed by the sigmoid activation function, which bounds the output between 0 and 1. A value of 0 means that all information from the previous cell should be forgotten, while a value of 1 means that all information should be remembered. The equation for the forget gate is given by:

$$f_t = \sigma \left(W_{hf} h_{t-1} + W_{xf} x_t + b_f \right) \tag{18}$$

Where

$$W_{xf} \in \mathbb{R}^{qxn} : \text{input weight matrix for the forget gate}$$
$$W_{hf} \in \mathbb{R}^{qxq} : \text{hidden state weight matrix for the forget gate}$$
$$b_f \in \mathbb{R}^{q} : \text{bias forget gate}$$
$$\sigma : \text{sigmoid activation function}$$

The LSTM cell has a layer called the "input gate" which decides which new information should be passed on to the cell state (Figure 18). This gate is composed of two layers. The first layer is a sigmoid layer which uses the sigmoid function to calculate a value between 0 and 1 for the weighted sum of the input (x_t) and previous hidden layer (h_{t-1}) in order to determine which information should be updated. The second layer is a tanh layer which creates a new vector $\tilde{C}_t$ using the same inputs used in the sigmoid layer, which consists of the information that could be added to the cell state. The tanh function outputs a value between -1 and 1, which is useful in cases where the impact of a component in the cell state needs to be reduced. The output of both layers is multiplied to determine which information is added to the new cell state.

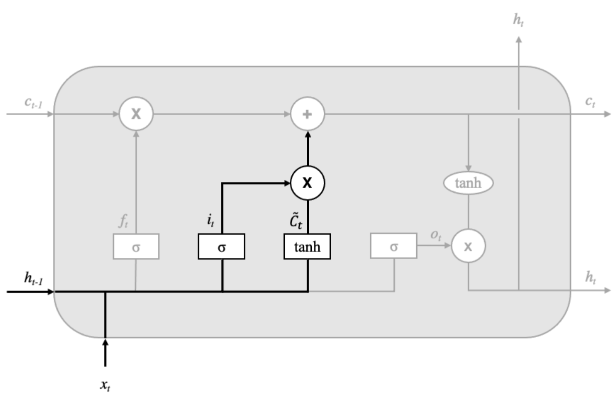

Figure 18: LSTM input gate

The input gate is computed using the equations represented by Equation 19 and Equation 20. i_t represents the output of the sigmoid layer and $\tilde{C}_t$ represents the output of the tanh layer.

$$i_t = \sigma\left(W_{hi}h_{t-1} + W_{xi}x_t + b_i\right) \tag{19}$$

$$\tilde{C}_t = \tanh\left(W_{hC}h_{t-1} + W_{xC}x_t + b_C\right) \tag{20}$$

Where

$$
\begin{aligned}
\tilde{C}_t \in \mathbb{R}^q &: \text{output tanh layer} \\
W_{xi}, W_{xC} \in \mathbb{R}^{q\text{x}n} &: \text{input weight matrix of the input sigmoid and tanh gate} \\
W_{hi}, W_{hC} \in \mathbb{R}^{q\text{x}q} &: \text{hidden state weight matrix for the input sigmoid and tanh gate} \\
b_i, b_C \in \mathbb{R}^q &: \text{bias for the input sigmoid and tanh gate} \\
\sigma &: \text{sigmoid activation function} \\
tanh &: \text{tanh activation function}
\end{aligned}
$$

The long-term memory of the network can be updated by combining the output of the forget gate f_t and input gate i_t with the previous cell state C_{t-1}. This is done by taking the element-wise product of the output of the forget gate f_t and the previous cell state C_{t-1}, and adding it to the element-wise product of the output of the sigmoid layer i_t and the output of the tanh layer $\tilde{C}_t$. This operation is also known as the Hadamard product and is denoted by the symbol $\circ$. Equation 21 describes the update of the previous cell state c_{t-1} into the new cell state c_t.

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{C}_t \tag{21}$$

The final gate in the LSTM cell is the "output gate", which is responsible for determining the new hidden state (h_t). This can be seen in Figure 19. In order to compute the new hidden state, three inputs are required: the previous hidden state (h_{t-1}), the updated cell state (c_t), and the current input (x_t). Similar to the forget and input gates, a sigmoid layer is applied to the input (x_t) and the previous hidden state (h_{t-1}) using Equation 22. To compute the updated hidden state, the updated cell state is passed through the tanh function to scale the output between -1 and 1, and then multiplied by the output of the sigmoid layer, as shown in Equation 23.

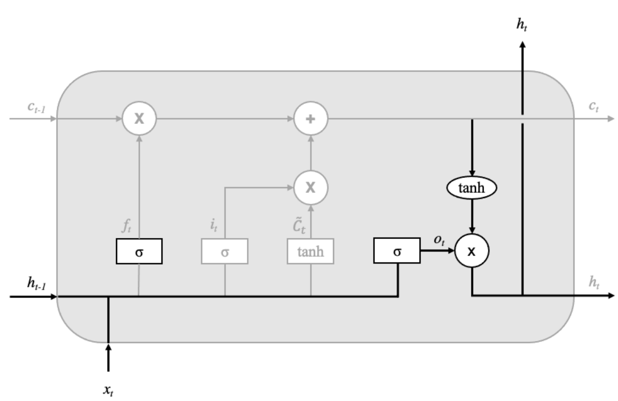

Figure 19: LSTM output gate

$$o_t = \sigma \left(W_{ho} h_{t-1} + W_{xo} x_t + b_o \right) \tag{22}$$

$$h_t = o_t \circ \tanh(c_t) \tag{23}$$

Where

$$\begin{aligned} W_{xo} \in \mathbb{R}^{q\text{x}n} &: \text{input weight matrix for the output gate} \\ W_{ho} \in \mathbb{R}^{q\text{x}q} &: \text{hidden state weight matrix for the output gate} \\ b_o \in \mathbb{R}^{q} &: \text{bias output gate} \\ \sigma &: \text{sigmoid activation function} \\ tanh &: \text{tanh activation function} \end{aligned}$$

The dimensions of all the variables involved in the LSTM cell computations, along with an overview of all the equations used to compute the output, are provided below.

$$\begin{aligned} f_t &= \sigma(W_{hf}h_{t-1} + W_{xf}x_t + b_f) \\ i_t &= \sigma(W_{hi}h_{t-1} + W_{xi}x_t + b_i) \\ \tilde{C}_t &= \tanh(W_{hC}h_{t-1} + W_{xC}x_t + b_C) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{C}_t \\ o_t &= \sigma(W_{ho}h_{t-1} + W_{xo}x_t + b_o) \\ h_t &= o_t \circ \tanh(c_t) \end{aligned}$$

The size and shape of the variables involved in the LSTM cell computations depend on the number of neurons $q \in \mathbb{Z}$ in the LSTM cell and the number of input features $n \in \mathbb{Z}$. The size and shape of the inputs, weights, bias and output can be expressed as:

$$\begin{aligned} &x_t \in \mathbb{R}^n \\ &c_t, h_t \in \mathbb{R}^q \\ &f_t, i_t, \tilde{C}_t, o_t \in \mathbb{R}^q \\ &W_{xf}, W_{xi}, W_{xC}, W_{xo} \in \mathbb{R}^{q\text{x}n} \\ &W_{hf}, W_{hi}, W_{hC}, W_{ho} \in \mathbb{R}^{q\text{x}q} \\ &b_f, b_i, b_C, b_o \in \mathbb{R}^q \\ &f_t, i_t, o_i \in \mathbb{R}^q \end{aligned}$$

Figure 20 gives an example of how the input shapes flow through the cell, illustrating the matrix multiplication in each gate. The number of neurons in the LSTM cell, denoted as q, determines the size of the hidden state and cell state, and the number of input features, denoted as n, determines the size of the input. In this example (Figure 20) the number of neurons is $q = 2$ and the number of features $n = 3$. The matrix multiplication in each gate is similar, for example, in the forget gate, the input matrix $W_{xf} \in \mathbb{R}^{q\text{x}n}$ is multiplied with the input vector $x_t \in \mathbb{R}^n$ and the hidden state matrix $W_{hf} \in \mathbb{R}^{q\text{x}q}$ is multiplied with the previous hidden state vector $h_{t-1} \in \mathbb{R}^q$. The outcomes after the multiplication are added, resulting in an output shape of $\mathbb{R}^q$.

So far, we have only discussed a single LSTM cell and its computations. However, in practice, a hidden LSTM layer typically consists of multiple cells. The number of cells in the layer is

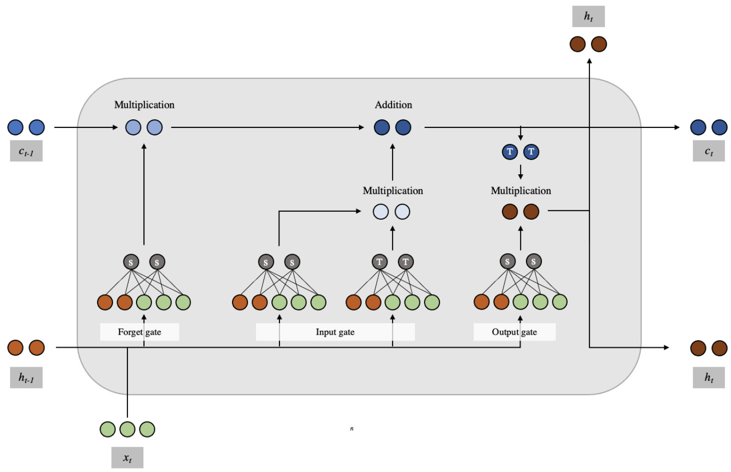

Figure 20: Dimensions within a LSTM cell

determined by the number of lags, which refer to the number of previous time steps used to compute the output. Figure 21 illustrates a LSTM layer with s number of cells. The LSTM layer receives input from the input layer and an initial hidden and cell state. For the entire LSTM layer, the input $x_t \in \mathbb{R}^{nxs}$ is composed of the number of features n and the number of lags s. The initial hidden and cell state vectors are set to zero, which is only the case at the first time step at the start of the training process. The output h_t of the LSTM layer is computed for the previous l time steps with input $x_t \in \mathbb{R}^n$. The s and T in Figure 20 represent the sigmoid (σ) and tanh activation functions.

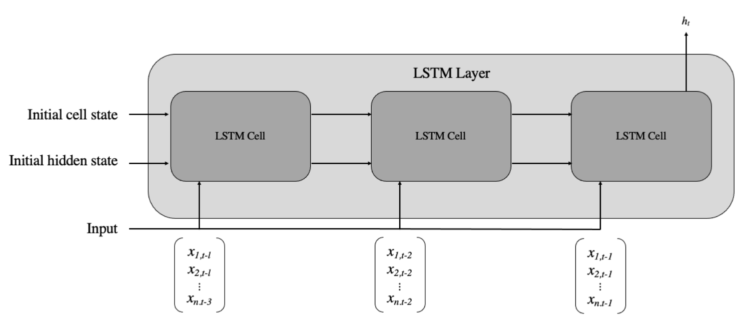

Figure 21: LSTM layer

The output of the LSTM network, $\hat{y}_t$, is calculated according to Equation 24. The formula for calculating this output uses the weight matrix W_y, which is a k x q dimensional matrix and connects the hidden LSTM layer to the output layer. The number of values the model outputs is equal to k. A bias, b_y, is added before the linear activation function is applied to obtain the final output.

$$\hat{y}_t = f(W_y h_t + b_y) \tag{24}$$

Overall, LSTM networks are able to capture long-term dependencies in data by using the memory cell and gates to selectively store and retrieve information from the past. This allows the network to effectively "remember" important information from earlier in the sequence, even when the input data is noisy or otherwise difficult to interpret.

3.4 Training Recurrent Neural Networks

Recurrent neural networks (RNNs) are trained using a variant of the backpropagation algorithm called backpropagation through time (BPTT). The basic idea of BPTT is to unroll the recurrent network over multiple time steps and treat it as a Feedforward Neural Network with multiple layers (Yu et al., 2019). The weights of the network are then updated using the standard backpropagation algorithm, where the error is propagated from the output layer to the input layer, and the gradients are calculated with respect to the weights. The training process begins by initializing the weights and biases of the network with random values. Then, the input sequence is fed into the network one time step at a time, and the network produces an output at each time step. The error between the predicted output and the actual output is then calculated, and the gradients of the error with respect to the weights are calculated using the chain rule of calculus.

The gradients of the error with respect to the weights are calculated by the following equation:

$$\frac{\partial L}{\partial W_j} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h_t} \sum_{k=1}^{T} \left(\left(\prod_{i=k+1}^{T} \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial W_j} \right) \tag{25}$$

where W_j is either weight matrix W_h or W_x. Note that when $k = T$, the value of the product is treated as 1 as there is nothing to multiply:

$$\prod_{i=T+1}^{T} \frac{\partial h_i}{\partial h_{i-1}} = 1 \tag{26}$$

RNNs are a type of neural network that are commonly used to process sequential data. However, when training RNNs using the traditional backpropagation algorithm, known as Backpropagation Through Time (BPTT), the network can suffer from a problem known as vanishing or exploding gradients. The vanishing gradient problem occurs when the gradients become smaller and smaller as the algorithm progresses backwards through the layers, eventually approaching zero and preventing the model from learning. On the other hand, the exploding gradient problem occurs when the gradients keep increasing as the algorithm progresses, resulting in large weight updates and causing the training algorithm to diverge. These problems become more pronounced as the number of previous time steps included in the algorithm increases. This is because the product of $\frac{\partial h_i}{\partial h_{i-1}}$ (in Equation 26) becomes smaller than 1, causing the gradients to vanish or larger than 1, causing the gradients to explode. This can make it difficult for the BPTT algorithm to converge to the optimal solution.

The LSTM model is a solution to the problem of vanishing or exploding gradients that can occur in RNNs. LSTMs are a type of RNN that are able to capture long-term dependencies by introducing a new element called the cell state (c_t). This cell state helps prevent gradients from vanishing by keeping track of the derivative of the cell state. This can be seen in Equation 28 and 29. Additionally, LSTMs have weight matrices between the current and previous hidden layers, as well as the input and hidden layers, which also depend on inputs in previous time steps. These matrices can be computed in a similar fashion as explained in Section 3.3 and the beginning of this Section. However, not all the partial derivatives of the weight matrices are stated. The goal of the training algorithm for LSTMs is to optimize the performance of the model by adjusting the weights to minimize the observed error, which is measured by the Mean Square Error, also known as the loss function (L).

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{C}_t \tag{27}$$

Via the multivariate chain rule, the derivative of the cell state with respect to the previous cell state is:

$$\begin{aligned} \frac{\partial c_t}{\partial c_{t-1}} &= \frac{\partial c_t}{\partial f_t}\frac{\partial f_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial c_{t-1}} + \frac{\partial c_t}{\partial i_t}\frac{\partial i_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial c_{t-1}} \\ &+ \frac{\partial c_t}{\partial \widetilde{C}_t}\frac{\partial \widetilde{C}_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial c_{t-1}} + \frac{\partial c_t}{\partial c_{t-1}} \end{aligned} \tag{28}$$

Now the four derivative terms are computed:

$$\begin{aligned} \frac{\partial c_t}{\partial c_{t-1}} &= \sigma'(f_t) \cdot W_{hf} \cdot o_{t-1} \circ \tanh'(c_{t-1}) \cdot c_{t-1} \\ &+ \sigma'(i_t) \cdot W_{hi} \cdot o_{t-1} \circ \tanh'(c_{t-1}) \cdot \tilde{C}_t \\ &+ \sigma'\left(\tilde{C}_t\right) \cdot W_{hC} \cdot o_{t-1} \circ \tanh'(c_{t-1}) \cdot i_t \\ &+ f_t \end{aligned} \tag{29}$$

The terms in partial derivative $\frac{\partial c_t}{\partial c_{t-1}}$ are multiplied by the number of time steps that are backpropagated. At any time step, each term can take a value above 1 or between 0 and 1. Thus, when the number of time steps increases it is not guaranteed that $\frac{\partial c_t}{\partial c_{t-1}}$ converges to 0 or infinity. Whereas in the vanilla RNN, $\frac{\partial h_i}{\partial h_{i-1}}$ will eventually take on values that are either always above 1 or between 0 and 1 which is what causes the gradient to vanish or explode (Noh, 2021).

In summary, Recurrent Neural Networks are trained using a variant of the backpropagation algorithm called backpropagation through time (BPTT), which unrolls the network over multiple time steps and treats it as a feedforward neural network with multiple layers. The weights of the network are updated using the standard backpropagation algorithm, and optimization algorithms such as stochastic gradient descent or Adam are used to minimize the error.

4 Empirical Approach

4.1 Data

To achieve reliable inflation rate forecasts, it is essential to gather relevant data that reflects the current economic conditions and trends. Econometric models such as ARIMA and VAR, as well as Neural Networks such as FFNN, RNN, and LSTM, require large amounts of data, including historical inflation rates and macroeconomic indicators. The quality and quantity of data used to train these models significantly impact their accuracy, making data gathering and preparation a critical step in the inflation rate forecasting process. The more accurate the data, the more reliable the forecasts, and the better the decisions that can be made based on them. The data is used to forecast the monthly one step ahead inflation rate in the Netherlands, where $\hat{y}_t$ is the predicted inflation rate.

4.1.1 Stationary data

It is important for data to be stationary when forecasting with econometric models because these models assume that the statistical properties of the data do not change over time. Stationarity is a key assumption for these models, and violating this assumption can lead to spurious forecasts. Stationary data means that the mean, variance, and autocorrelation structure of the data do not change over time. If the data is non-stationary, the model may produce biased forecasts and unreliable confidence intervals. To address this issue, the data can be transformed to achieve stationarity, for example by taking first differences, logarithms, or seasonal differences. If the data is non-stationary, the model may produce unreliable forecasts and biased estimates of the coefficients.

To determine whether data is stationary or not, a common test used in econometrics is the Dickey-Fuller test. The Dickey-Fuller test is a statistical test that checks for the presence of a unit root in a time series dataset. A unit root implies that the data is non-stationary, as it indicates that the statistical properties of the data change over time. The null hypothesis of the Dickey-Fuller test is that the data has a unit root and is non-stationary, while the alternative hypothesis is that the data is stationary. The mathematical formula for the Dickey-Fuller test is as follows:

$$\Delta z_t = \alpha + \beta t + \gamma z_{t-1} + \epsilon_t \tag{30}$$

where Δz_t is the first difference of the dependent variable, α is the intercept term, β is the coefficient on time, γ is the coefficient on the lagged dependent variable, and ϵ_t is the error term. If the calculated test statistic is less than the critical value at the chosen level of significance, then we reject the null hypothesis of non-stationarity and conclude that the data is stationary. Conversely, if the test statistic is greater than the critical value, we fail to reject the null hypothesis and conclude that the data is non-stationary. If the data is found to be non-stationary, it can be adjusted to become stationary by taking the first difference of the data or by using other techniques such as seasonal adjustment or logarithmic transformation. Taking the first difference involves subtracting each observation from the previous observation to remove the trend component from the data. Once the data is stationary, it can then be used for inflation forecasting using econometric models.

For neural networks data does not necessarily have to be stationary. Unlike traditional time series models, neural networks are capable of modeling complex and nonlinear relationships between variables, and can learn to adapt to changes in the statistical properties of the data over time. However, stationarity can still be a useful property to have in the data, as it can simplify the modeling process and improve the performance of the neural network. For example, stationary data may exhibit simpler patterns and relationships that are easier for the neural network to learn and generalize to new data. Additionally, stationarity may help reduce the impact of noise and outliers in the data. If the data is non-stationary, it may be necessary to apply transformations to the data, such as differencing or detrending, to achieve stationarity or at least reduce the degree of non-stationarity. However, care must be taken to avoid introducing biases or distorting the data, as some transformations can affect the patterns and relationships in the data that the neural network is meant to learn.

4.1.2 Data gathering

This research is focused on forecasting the inflation rate in the Netherlands, using the Dutch Consumer Price Index (CPI) as the measure of inflation. The CPI is collected from the "Centraal Bureau voor de Statistiek" (CBS), which is an institution that publishes all kinds of Dutch social and economic statistics. Additionally, the data of other macroeconomic indicators, such as GDP and unemployment rates, are also collected from the CBS. In order to prepare the data for analysis, the historical inflation rate, along with macroeconomic indicators such as GDP, unemployment rates, interest rates, money supply and gold price are collected from various data sources, including the CBS and ECB databases, as well as the Investing.com database. For data for the gold price, GDP and money supply. The descriptive statistics of the dataset are given in Table 2. After differencing, the Augmented Dickey-Fuller (ADF) test rejects the null hypothesis of a unit root at the 5% level.

	CPI	**Interest**	**Gold**	**Unemployment**	**GDP**	**M3**
count	192	192	192	192	192	192
mean	2.25%	2.12%	0.01%	5.85%	0.26%	0.58%
std	2.31%	1.52%	0.05%	1.46%	0.57%	0.45%
min	-0.20%	-0.21%	-0.11%	3.20%	-2.24%	-0.16%
25%	1.10%	0.98%	-0.02%	4.70%	0.11%	0.40%
50%	1.80%	1.92%	0.00%	5.65%	0.24%	0.52%
75%	2.50%	3.49%	0.03%	6.80%	0.43%	0.64%
max	14.50%	4.96%	0.16%	9.00%	2.37%	4.40%
ADF	-3.27	-5.79	-13.69	-3.23	-7.54	-7.2

Table 2: Descriptive statistics of dataset

The data covers a time period of 16 years from 2007-01-01 to 2022-12-01, consisting of 192 monthly observations. The length of the considered time period is mainly driven by the availability of data. The data is then cleaned, transformed, and split into training, validation, and testing sets, with a split ratio of 60%, 20%, and 20%, respectively. The training set, which covers the period from 2007-01-01 to 2019-09-01, is used to train the models, while the out-of-sample period from 2019-10-01 to 2022-01-01 is used to evaluate the out-of-sample forecast. A part of the training set is used as validation data to tune the parameters of the neural network models. A visualization of the data split is provided in Figure 22. The time period considered includes some major events such as the end of the financial crisis, COVID-19 and the Russian invasion of Ukraine which had a major effect on the global economic landscape. The end of the financial crisis marked a turning point for many economies, as they began to recover and experience growth once again. However, the outbreak of COVID-19 in late 2019 had a significant impact on the world, leading to widespread lockdowns, economic disruption, and significant loss of life. Additionally, the ongoing conflict between Russia and Ukraine has strained relations between these countries and other nations, as well as impacting global trade and stability. Overall, the events during this time period have had a profound effect on the world economy. As can be seen in Figure 22, inflation surged to extreme highs after COVID-19 and the Russian invasion of Ukraine.

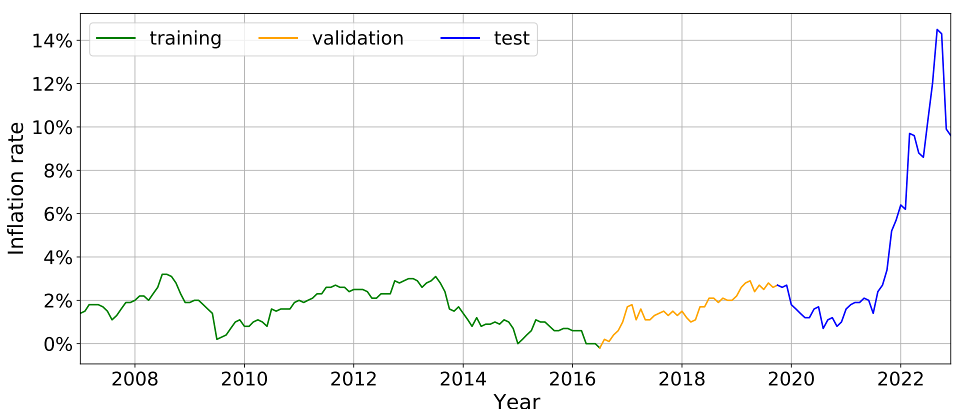

Figure 22: Historical inflation rate split into a training, validation and test set

4.1.3 Data preparation

Once the data is collected, it must be prepared for analysis. This involves cleaning the data by removing any missing or inaccurate data points, transforming the data into a suitable format for analysis, and splitting the data into a training, validation and test set. They are used to train and evaluate the model's performance during the development phase and to assess the model's accuracy in making predictions on new, unseen data.

The training set is the portion of the data used to train the machine learning model. The model is trained on this data to learn the underlying patterns and relationships between the input features and output targets. The training set typically represents the largest portion of the available data, typically around 60% to 80% of the total dataset. The validation set is used to evaluate the performance of the model during training. After the model has been trained on the training set, it is evaluated on the validation set to assess its performance. The validation set helps to identify whether the model is overfitting or underfitting. Overfitting occurs when the model is too complex and performs well on the training set but poorly on the validation set. Underfitting occurs when the model is too simple and performs poorly on both the training and validation sets. The test set is used to assess the accuracy of the model's predictions on new, unseen data. After the model has been trained and evaluated on the training and validation sets, it is then evaluated on the test set. The test set provides an unbiased evaluation of the model's performance on new data that it has not been trained on.

The next step is to normalize the data to ensure that it is on the same scale. Normalization is essential for neural networks because it helps to prevent any single feature from dominating the analysis. This can be done by scaling the data to certain range, for example between -1 and 1.

4.2 Performance measures

Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are two common performance measures used to evaluate the accuracy of a model's predictions. Both measures are used to compare the predicted values from a model to the true values in a dataset, and provide an indication of the model's prediction error. RMSE is calculated as the square root of the mean squared error (MSE), which is the average of the squared differences between the predicted values and the true values. The RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2} \tag{31}$$

where n is the number of samples, y_i is the true value for sample i, and $\hat{y}_i$ is the predicted value for sample i. RMSE is commonly used for regression tasks, where the goal is to predict a continuous value. It is a popular choice because it is easy to interpret, as it is expressed in the same units as the predicted and true values (Ghasemi and Zahedi, 2010).

MAE is another common performance measure for regression tasks, and is calculated as the average of the absolute differences between the predicted and true values. The MAE is defined as:

$$MAE = \frac{1}{n}\sum_{i}^{n} |y_i - \hat{y}_i| \tag{32}$$

where n is the number of samples, y_i is the true value for sample i, and $\hat{y}_i$ is the predicted value for sample i. MAE is also easy to interpret, as it is expressed in the same units as the predicted and true values. However, unlike RMSE, it is not sensitive to outliers in the data, as it only takes into account the magnitude of the error, rather than the squared error (Zhang and Yang, 2015).

Both RMSE and MAE have their advantages and disadvantages, and the appropriate performance measure will depend on the specific task and requirements. Both RMSE and MAE are easy to interpret, as they are expressed in the same units as the predicted and true values. This can be helpful for understanding the magnitude of the prediction error (Ghasemi and Zahedi, 2010). Howeverm RMSE is sensitive to outliers in the data, as the squared error will be larger for larger errors. This can be a disadvantage if there are outliers in the data that are not representative of the majority of the samples. MAE is not sensitive to outliers, as it only takes into account the magnitude of the error (Zhang and Yang, 2015).

Both MSE and MAE provide an indication of the model's prediction error, and can be used to compare the performance of different forecasting models. MSE is often used when the goal is to minimize the prediction error, as it penalizes larger errors more heavily due to the squared difference. MAE is often used when the goal is to minimize the prediction error, and the data contains outliers or the range of possible values is important to consider, as it is not sensitive to outliers and only takes into account the magnitude of the error.

In general, RMSE is a good choice when the goal is to minimize the prediction error, and the data is free of outliers. MAE is a good choice when the goal is to minimize the prediction error and the data contains outliers, or when the range of possible values is important to consider (Zhang and Yang, 2015). It is also worth noting that there are many other performance measures that can be used for regression tasks, depending on the specific requirements and characteristics of the data. Some examples include mean squared logarithmic error (MSLE), mean absolute percentage error (MAPE), and coefficient of determination (R^2).

4.3 Hyperparameters of a Neural Network

Hyperparameters are values that are set before training a neural network model and are used to control the learning process. They are typically chosen through a process called hyperparameter optimization, which involves searching for the best combination of hyperparameters to achieve the best performance on a given task. There are many different hyperparameters that can be adjusted in a neural network, and the specific hyperparameters that are relevant will depend on the specific architecture and type of model being used. Some common hyperparameters include:

1. Learning rate: This is a scalar value that determines the step size at which the optimizer makes updates to the model weights during training. A larger learning rate can lead to faster convergence, but also carries a higher risk of overshooting the optimal weights and leading to poor generalization (Goodfellow et al., 2016).

2. Batch size: This is the number of training examples that are processed at once by the model during training. A larger batch size can lead to faster training, but may also require more memory and can result in worse generalization (Chollet, 2018).

3. Number of epochs: This is the number of times the model will see the entire training dataset during training. A larger number of epochs can lead to better model performance, but also carries a higher risk of overfitting (Raschka, 2015).

4. Activation function: This is a function applied to the output of each neuron in the model that determines the output of the neuron. Common activation functions include sigmoid, tanh, and ReLU (Chollet, 2018).

5. Number of hidden layers: The number of hidden layers in a neural network can affect the model's capacity and ability to learn complex patterns in the data. A deeper network with more hidden layers can potentially capture more intricate relationships, but also carries a higher risk of overfitting (Goodfellow et al., 2016).

6. Size of hidden layers: The size of the hidden layers, or the number of neurons in each hidden layer, can also affect the model's capacity and ability to learn complex patterns. A larger number of neurons in the hidden layers can allow the model to capture more intricate relationships, but again carries a higher risk of overfitting (Goodfellow et al., 2016).

7. Optimizer: The optimizer is the algorithm used to update the model weights during training. Different optimizers have different properties and may work better or worse for different tasks and model architectures. Common optimizers include stochastic gradient descent (SGD), Adam, and RMSprop (Chollet, 2018).

8. Loss function: The loss function is a measure of how well the model is able to predict the desired output for a given input. Different loss functions are suitable for different tasks and can have a significant impact on the model's performance. Common loss functions include mean squared error (MSE) for regression tasks and cross-entropy loss for classification tasks (James et al., 2013).

Hyperparameter optimization is typically done through a process called grid search, in which a set of hyperparameters is defined and the model is trained and evaluated for each combination of hyperparameters. The combination that leads to the best performance on the validation set is then chosen as the final set of hyperparameters. Another approach to hyperparameter optimization is called random search, in which a set of hyperparameters is randomly generated and the model is trained and evaluated with those hyperparameters. This process is repeated a number of times, and the combination of hyperparameters that leads to the best performance is chosen as the final set (Bergstra and Bengio, 2012). There are also more advanced optimization techniques, such as Bayesian optimization and evolutionary algorithms, that use probabilistic models and search heuristics to more efficiently search the hyperparameter space (Snoek et al., 2012).

In general, it is important to carefully choose the hyperparameters for a neural network, as the choice of hyperparameters can have a significant impact on the model's performance. It is also important to use a held-out validation set to evaluate the performance of the model during the hyperparameter optimization process, to ensure that the chosen hyperparameters will generalize well to unseen data. It is also worth noting that finding the optimal set of hyperparameters can be a time-consuming and iterative process, as it often involves training multiple models with different hyperparameter combinations and evaluating their performance. Automated hyperparameter optimization techniques, such as those mentioned above, can be helpful in speeding up this process and finding good hyperparameter values more efficiently.

In summary, selecting the relevant hyperparameters for a neural network is an important aspect of the model training process, and can have a significant impact on the model's performance. Careful selection of hyperparameters can help to ensure that the model is able to effectively learn and generalize to new data.

5 Results & Discussion

5.1 Introduction

In this section the hyperparameters of the econometric and neural networks are selected. Once the hyperparameters of the econometric and Neural network models have been selected, the models can be constructed. The remainder of this section presents and interprets the results of the inflation forecasting models, including econometric and neural network models. The models will be trained on the training and validation set to optimize the parameters and weights. Furthermore, for each model the in and out of sample prediction are generated and evaluated using RMSE and MAE. The lower the values of these measures, the better the performance of the model.

5.2 Hyperparameter selection

5.2.1 Hyperparameter selection of ARIMA

The selection of the appropriate parameters for an ARIMA(p,d,q) model is crucial to ensure that the model fits the data well and produces accurate forecasts. One common approach to parameter selection is to use the Root Mean Squared Error (RMSE) as a measure of accuracy and select the model with the lowest RMSE. The optimal values of p, d, and q for the ARIMA model are selected by iterating through different combinations of p, d, and q. The model with the lowest RMSE is considered the best-fitting model and is selected for out of sample forecasting. The parameter combination with the lowest RSME is ARIMA(1,1,3), as can be seen in Table 3.

ARIMA(p,d,q)	**RMSE**
ARIMA (1,1,1)	0.316
ARIMA (2,1,1)	0.312
ARIMA (3,1,1)	0.312
ARIMA (1,1,2)	0.314
ARIMA (1,1,3)	**0.308**
ARIMA (1,2,0)	0.371
ARIMA (2,2,0)	0.364
ARIMA (0,2,1)	0.306
ARIMA (0,2,2)	0.318
ARIMA (0,2,3)	0.315
ARIMA (1,2,1)	0.318
ARIMA (1,2,2)	0.314
ARIMA (1,2,3)	0.315
ARIMA (2,2,1)	0.317
ARIMA (2,2,2)	0.322
ARIMA (2,2,3)	0.362
ARIMA (3,2,0)	0.360
ARIMA (3,2,1)	0.315
ARIMA (3,2,2)	0.314

Table 3: In sample RMSE of ARIMA(p,d,q) model

The model parameter selection results in a constructed model with the following coefficients represented in Table 4. The L1 AR coefficient is the autoregressive (AR) parameters of the model, while the L1, L2 and L3 MA coefficients are the moving average (MA) parameter. The AR parameter represent the correlation between the current value of the dependent variable (CPI in this case) and its past values, where L indicates the number of time periods back in the past that the model considers. Similarly, the MA parameters represent the correlation between the current value of the dependent variable and its past errors, where L indicates the number of time periods back in the past that the model considers. The ARIMA(1,1,1) model has R^2 of 0.89 on the combined training and validation set.

variable	**coefficient**	**std error**	**z-value**
AR L1	0.506	0.77	0.66
MA L1	-0.460	0.75	-0.61
MA L2	-0.063	0.12	0.53
MA L3	-0.130	0.09	-1.44

Table 4: Coefficient ARIMA model

5.2.2 Hyperparameter selection of VAR

The selection of appropriate parameters for a VAR(p) model is also crucial to ensure that the model fits the data well and produces accurate forecasts. The model with the lowest RSME value is considered the best-fitting model and is selected for out of sample forecasting. The model with the lowest RMSE is considered the best-fitting model and is selected for out of sample forecasting. The parameter combination with the lowest RSME is VAR(1), as can be seen in Table 5.

VAR(p)	**RSME**
VAR(1)	**0.317**
VAR(2)	0.318
VAR(3)	0.322
VAR(4)	0.324
VAR(5)	0.340
VAR(6)	0.343
VAR(7)	0.338

Table 5: In sample RMSE of VAR(p) model

The model parameter selection results in a constructed model with the following coefficients of the CPI factor represented in Table 6. L1 refers to the lag order of the variables included in the model, the value of the variable from the previous time period. The variable included in the VAR model are the historical CPI, the interest rate, gold price, unemployment, GDP and the money supply (M3). The VAR(1) model has R^2 of 0.89 on the combined training and validation set.

variable	**coefficient**	**std. error**	**t-statistic**
const	0.123	0.13	0.95
L1 CPI	0.060	0.08	0.72
L1 Interest	-0.199	0.11	-1.81
L1 Gold	0.002	0.00	0.46
L1 Unemployment	-0.021	0.02	-1.16
L1 GDP	-0.090	0.14	-0.64
L1 M3	0.014	0.11	0.13

Table 6: The Coefficients for the VAR(1) CPI equation

5.2.3 Hyperparameter selection of neural networks

In this section, the hyperparameters for the FFNN, RNN and LSTM are determined. This process involves tuning the hyperparameters, which are the external configuration variables of the model. Since there are countless possible parameter combinations, an initial architecture is set up, and different hyperparameters are investigated to optimize the neural network. The initial network consists of an input layer, a single hidden layer, and an output layer. According to Stinchcombe and White (1992), a network with a single hidden layer can approximate a wide range of linear and nonlinear functions, provided that a reasonable number of neurons are included in the hidden layer. Therefore, a single hidden layer is used for all three types of neural networks. Including too many hidden layers can also cause overfitting of the model. To ensure generality and comparability of the FFNN, RNN, and LSTM, the same hyperparameters will be used for all three.

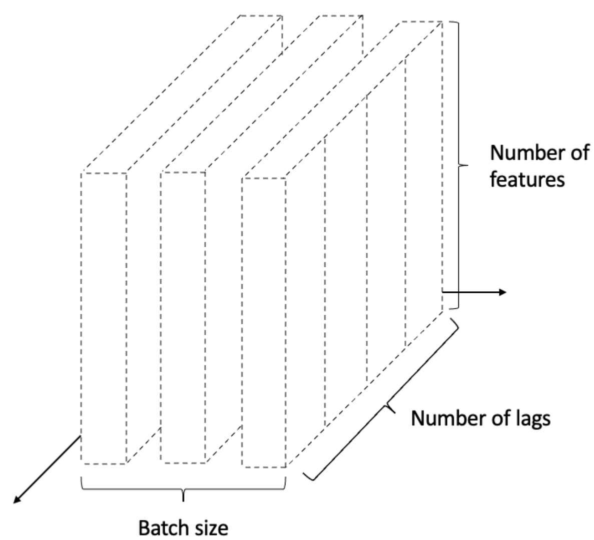

Figure 23: RNN & LSTM input shape

To tune the hyperparameters, it is crucial to understand the input shape of RNNs, including LSTMs, in comparison to FFNNs. The input shape for RNNs is a three-dimensional array (batch size, number of time steps, number of features) as shown in Figure 23. The first dimension represents the batch size, which is a hyperparameter that specifies the number of samples to be processed before updating the model's internal parameters. The second dimension is the number of time-steps, often referred to as lags, and this hyperparameter determines the number of previous time-steps to include in the input. The third dimension is the number of features, which are the different input variables. For the RNN and LSTM models, the number of features is equal to the number of variables used to forecast the inflation rate: lagged inflation rate, interest rate, GDP, gold price, unemployment rate and the M3 money supply. The number of lags and batch size are determined later on in this section. The inputs are passed through the hidden layer to the output layer, which consists of a single neuron that outputs the inflation rate forecast using a linear activation function.

In contrast to recurrent neural networks, FFNNs require a two-dimensional input shape (batch size, number of features) since they are unable to take prior time steps as input. In order to compare the input sizes of three different types of neural networks investigated, the FFNN input size is adjusted to (batch size, (number of features + number of lags)). Rather than the prior lags flowing through the hidden state, they are included in the input array. The sigmoid function is a typical choice for the activation function in the hidden layer of FFNNs, according to Bucci (2020). In order to predict inflation rates, a linear activation function is used in the output layer.

The model makes a prediction of the inflation rate for each time step. The actual inflation rate from the test set is then used to forecast the inflation rate for the subsequent time step. This process imitates a real-world scenario in which the inflation rate of the prior time step is known and utilized to estimate the inflation rate for the next time step.

The optimization algorithm used is Adaptive Moment Estimation (Adam), as opposed to the classical stochastic gradient descent (SGD). Unlike SGD, Adam employs a variable learning rate throughout the model training process. Adam offers a significant advantage over other optimization algorithms, such as SGD, in that it converges more rapidly to the local minima, which leads to faster results Sang and Di Pierro (2019).

In order to avoid overfitting, a dropout rate is implemented in the hidden layer. Dropout randomly removes neurons, reducing the sensitivity to specific weights of the individual neurons (Srivastava et al., 2014). The dropout value is a trade-off between retaining accuracy and avoiding overfitting. In the FFNN, a dropout rate of 0.1 is applied between the hidden layer and the output layer, while for the RNN and LSTM models, a dropout rate of 0.1 is applied to the recurrent input signals in the hidden layer.

The hyperparameters of the NNs are tuned after its construction. The order in which the hyperparameters are selected is as follows: the number of lags and neurons in the hidden layer, batch size, and number of epochs. It is noted by Breuel (2015) that successful hyperparameter selection on one dataset does not guarantee the same outcome on another dataset with different characteristics. There is no set formula for selecting the right set of hyperparameters, and as such, decision-making often involves prior knowledge and trial-and-error (Young et al., 2015). The performance of the model on the validation set is used to determine the optimal hyperparameters.

The first step in tuning the hyperparameters involves finding the optimal combination of lags per feature and the number of neurons in the hidden layer. The learning capacity of the network is determined by the number of neurons. However, too many lags and neurons can result in overfitting, where the model performs exceptionally well on the training data but poorly on the unobserved test data. The number of lags is the same for all features, and the root-mean-square error (RMSE) is calculated for each combination of lags and neurons to determine the best NN architecture. The number of lags ranges from 1 to 6, while the number of neurons in the hidden layer ranges from 2 to 20, with increments of 2. The results, presented in Table 7, indicate that the combination of 2 lags and 16 neurons yields the lowest RMSE.

Number of Lags	Number of Neurons: 2	4	6	8	10	12	14	16	18	20
1	0.389	0.375	0.385	0.412	0.398	0.341	0.369	0.341	0.367	0.361
2	0.515	0.508	0.346	0.349	0.406	0.471	0.426	**0.286**	0.315	0.330
3	0.493	0.363	0.344	0.328	0.384	0.407	0.361	0.308	0.326	0.338
4	0.489	0.368	0.350	0.376	0.371	0.394	0.363	0.326	0.331	0.322
5	0.489	0.497	0.391	0.401	0.386	0.418	0.357	0.347	0.417	0.318
6	0.495	0.341	0.380	0.359	0.361	0.341	0.358	0.342	0.450	0.324

Table 7: Average MSE of FFNN, RNN & LSTM for various Number of Lags & Neurons combinations

In addition, the batch size hyperparameter is determined, which specifies the number of samples to process before updating the model's internal parameters. Often, batch sizes are chosen to be powers of 2. A batch size of 1 and a batch size equal to the length of the training dataset are also included. With these batch sizes, mini-batch-gradient descent is reduced to SGD and converted batch gradient descent, respectively. The RMSE is calculated for each batch size, and the batch size resulting in the lowest error is selected. As shown in Figure 24, a batch size of 16 yields the lowest RMSE for all three NNs.

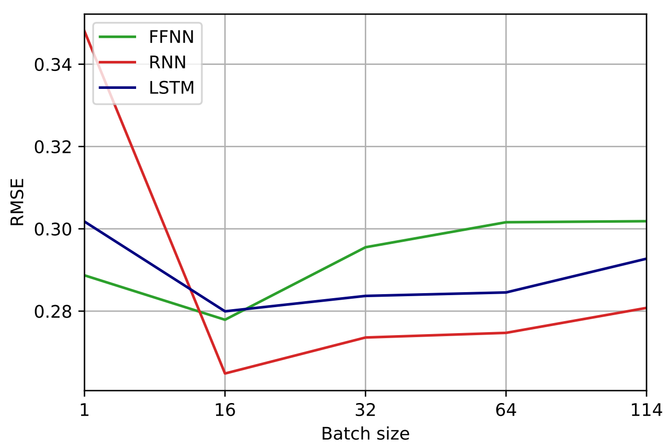

Figure 24: Batch size

The final hyperparameter to be tuned is the number of epochs. It is an essential hyperparameter for the performance of a NN. The number of epochs represents the number of times the learning algorithm passes through the entire training dataset. When the number of epochs is too low, it will be difficult for the training data to converge, and if it is too high, overfitting can occur. The train and validation RMSE scores are plotted in Figure 25, which is often called the learning curve. It can be observed from the plot that after approximately

1000 epochs, the learning rate stabilizes.

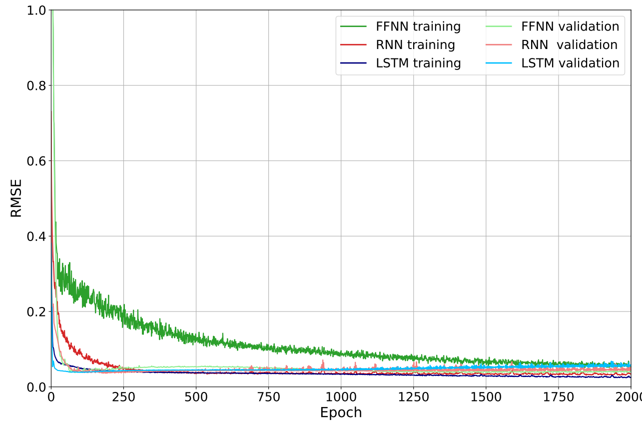

Figure 25: Number of Epochs

Table 8 provides a summary of the architecture and hyperparameters used to train the FFNN, RNN, and LSTM models. It includes the number of input features, the number of lags used, the number of neurons in the hidden layer, the activation function used in the hidden layer, the optimizer used for training, the dropout rate applied, the batch size used, and the number of epochs trained. The FFNN, RNN and LSTM model have a R^2 of 0.89, 0.92, 0.94 on the combined training and validation set respectively.

NN Model	
Number of Hidden Layers	Single hidden layer
Number of Neurons	16
Number of Lags	2
Inputs	Historical inflation, interest rate
Output	1
Activation function output layer	Linear
Number of epochs	1000
Batch Size	16
Loss Function	Mean Square Error
Dropout Ratio	0.1
Training Set as of Total Data	60%
Validation Set as of Total Data	20%
Test Set as of Total Data	20%
Optimization Algorithm	Adam

Table 8: NN model specifications

5.3 Result Naive predictor

Naive forecasting is a simple method of forecasting where the future values of a variable are assumed to be equal to the most recent value observed in the data. This method is commonly used as a benchmark to compare the performance of more sophisticated forecasting models, such as econometric and neural network models. The idea behind using naive forecasting as a benchmark is to compare the accuracy of the more complex models with a simple and easily understandable benchmark. The naive predictor used is the value of the variable at time $t-1$, as expressed in Equation 33.

$$\hat{y}_t = y_{t-1} \tag{33}$$

Where $\hat{y}_t$ is the forecasted value for the next time period (t) and y_{t-1} is the observed value at time $t-1$. If a model cannot outperform the naive forecast, it may not be useful for practical forecasting purposes. While naive forecasting is a very simple method, it can still provide valuable insights into the forecasting problem. It is often used as a baseline model to assess the accuracy of more complex models. A model that performs worse than the naive forecast is unlikely to provide any real forecasting value. Figure 26 illustrates the performance of the naive predictor, showing that the predicted forecast is lagging behind the actual values due to its reliance on the previous observed value as the sole predictor. The naive predictor has a R2 of 0.89 on the combined training and validation set.

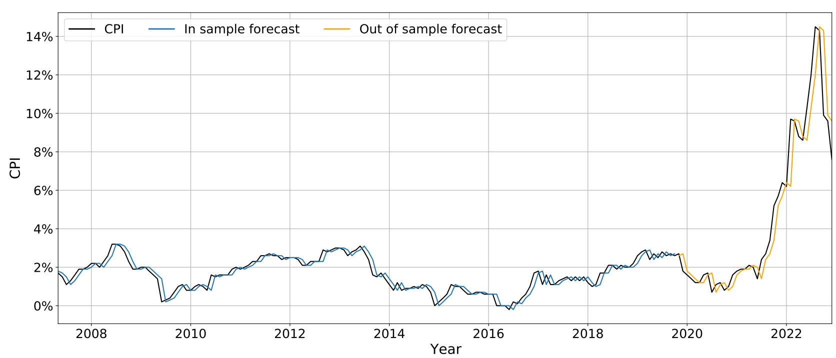

Figure 26: Naive predictor result of Dutch inflation rate forecast

5.4 Results Econometric models

The econometric models used in this study include VAR and ARIMA. The results in Table 9 show that both ARIMA and VAR models outperform the naive predictor. The naive predictor simply uses the last observed value as the prediction for the next period, and it serves as a baseline for comparison. The results indicate that the VAR model has a slightly lower RMSE in and out of sample compared to the ARIMA model, making it the better performing econometric model. However, the both the ARIMA and the VAR model have a higher in sample MAE. A possible reason for the higher in-sample MAE of the ARIMA and VAR models could be due to the presence of outliers or unusual observations in the training data. Figure 27 visualized the ARIMA prediction compared to the actual inflation and Figure 28 the prediction of the VAR model. As can be seen from both figures, both models accurately forecast the one month ahead inflation rate. The VAR model captures the dynamic interrelationships between the historical inflation rate, money supply, GDP, interest rate, unemployment rate and the gold price. As can be seen in Figure 28, the VAR model forecasts an overreacted drop in inflation as a reaction to the start of the COVID-19 pandemic in the beginning of 2020. This could be because the pandemic caused a sudden and significant shock to the economy, causing the GDP to drop and unemployment and money supply to increase.

Figure 27: ARIMA result of Dutch inflation rate forecast

Figure 28: VAR result of Dutch inflation rate forecast

5.5 Results Neural Network models

The neural network models used in this study include FFNN, RNN, and LSTM. Table 9 shows the RMSE and MAE values of the models for in-sample and out-of-sample forecasts. All three NNs have a lower RSME in sample compared to the naive predictor, indicating the the NNs fit the in sample data well. However, RNN and LSTM models do not perform better than the naive predictor out of sample. The results show that the LSTM model has the lowest RMSE and MAE values among all the models tested for in-sample forecasting, with RMSE of 0.2055 and MAE of 0.1459. For out-of-sample forecasting, the FFNN model has the lowest RMSE and MAE values, with RMSE of 1.2203 and MAE of 0.7449. The in and out of sample predicted inflation rate of the FFNN, RNN and LSTM models are depicted in Figures 29, 30 & 31 respectively. The results also suggest that the LSTM model is the most accurate model for predicting inflation based on historical data, while the FFNN model can better generalize to future data. A reason for the better performance of the FFNN model compared to the RNN and LSTM model could be because of the rapid increase of inflation in out of sample test set. As the RNN and LSTM models have longer-term memory and are able to retain information over a longer period of time, they may struggle to adjust to rapid changing market dynamics compared to the FFNN model. This is because the FFNN model relies on a simpler feedforward architecture that is better suited to processing and reacting to rapid changes in data. In contrast, the RNN and LSTM models may be slower to adapt to sudden shifts in market dynamics, which could explain their relatively poorer performance on the out-of-sample data. It is important to note that this does not necessarily mean that the RNN and LSTM models are inferior to the FFNN model in all contexts, but rather that their strengths and weaknesses may differ depending on the specific characteristics of the data being analyzed.

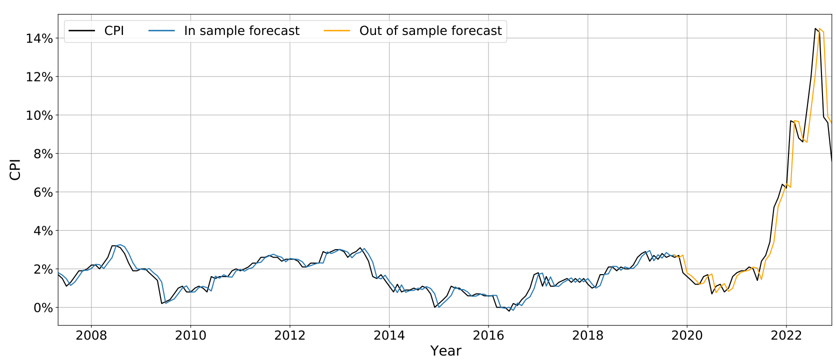

Figure 29: FFNN result of Dutch inflation rate forecast

Another reason for the poor out of sample performance of the RNN and LSTM model could the the nature of the out of sample data. Namely, the out of sample data set includes the COVID-19 pandemic and Russia's invasion of Ukraine. These extreme events caused sudden and significant shock to the economy, and the RNN and LSTM models may have struggled to incorporate these unusual events into their forecasts. This is because these models rely on patterns and trends in past data to make predictions, and may not be well-suited to handling sudden and unexpected shocks to the system. Especially, the in the out of forecast the LSTM it can be seen that the model over and under shoot when trying to predict the next month inflation (Figure 31).

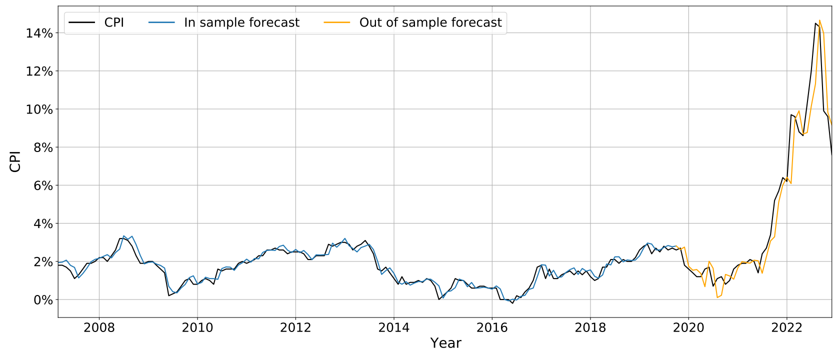

Figure 30: RNN result of Dutch inflation rate forecast

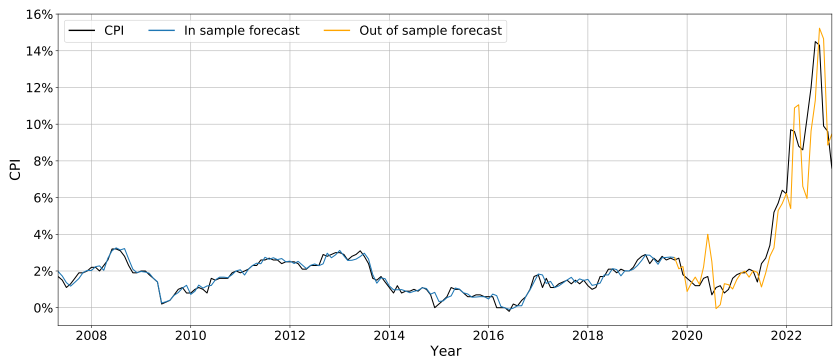

Figure 31: LSTM result of Dutch inflation rate forecast

5.6 Comparison between the Econometric and Neural Network models

The results show that the neural network models outperform the econometric models in the in-sample forecasts. In terms of in-sample forecasting, LSTM has the best in-sample performance, indicating that it is the most accurate model for predicting inflation based on historical data. However, in terms of out-of-sample forecasting, econometric models have a better performance, indicating that it can better generalize to future data. Due to the challenging data set, containing the COVID-19 pandemic and the invasion of Ukraine by Russia, the RNN and LSTM poorly performed in the out-of-sample forecast. These extreme events caused sudden and significant shocks to the economy, resulting in rapid changes in market dynamics that the RNN and LSTM models had difficulty adjusting to. Additionally, the RNN and LSTM models may not have been able to capture the complex interactions between economic variables during such unprecedented events, as they rely on past patterns in data to make predictions. It is worth noting that the study highlights that a more complex model does not necessarily result in better and more accurate performance. This is evident from the performance of the RNN and LSTM models, which, despite their complexity, did not perform as well as the simpler econometric models in out-of-sample forecasting. This suggests that the appropriate model choice should depend on the specific context and dataset being analyzed, rather than simply selecting the most complex model available.

	RMSE_in	**RMSE_out**	**MAE_in**	**MAE _out**
Naive predictor	0.2812	1.2286	0.2033	0.7526
ARIMA	0.2798	1.1965	0.2062	0.7239
VAR	0.2765	1.1964	0.2076	0.7107
FFNN	0.2760	1.2203	0.2035	0.7449
RNN	0.2413	1.2672	0.1814	0.8195
LSTM	0.2055	1.7197	0.1459	1.2044

Table 9: Performance measure of the Econometric & Neural Network models

The computation time per model is stated in Table 10. As expected, the ARIMA and VAR model are much faster than the Neural Networks. The econometric models are estimated using the Maximum Likelihood Estimation method and the NNs are trained using the methods explained in Section 3.3 & 3.4. The running time of a VAR model is faster than that of the ARIMA model because VAR model has fewer parameters to estimate than ARIMA model. Regarding the NNs, the FFNN model has the shortest running time as it is the simplest NN of the three. The RNN and LSTM also take output from prior time steps as input which increases the number of computation needed to train the network. The LSTM model takes the longest to train as it is the most complex model.

Model	**Running Time (s)**
ARIMA	0.105
VAR	0.010
FFNN	27
RNN	35
LSTM	42

Table 10: Running time of Econometric and Neural Network Models

6 Conclusion & Future Research

In conclusion, inflation forecasting is highly relevant for business valuation as it affects the expected cash flows and risk of a company, which are crucial factors in determining its value. Proper handling of inflation is necessary in discounted cash flow models and can significantly impact the valuation of a business. Incorporating inflation into the WACC calculation and forecasting inflation-adjusted cash flows can help minimize the impact of inflation on the value of a business. Therefore, businesses should focus on increasing prices, improving operations, and revisiting finances to mitigate the impact of inflation on their value. Inflation is a complex phenomenon that is influenced by a wide variety of factors, including short-term and long-term factors, as well as structural changes in the economy. Traditional econometric models such as VAR and ARIMA have been successful in predicting inflation rates, but they have limitations and are based on linear and stationary assumptions. Neural networks, or artificial neural networks (ANNs), have become increasingly popular for inflation forecasting due to their ability to handle large and complex datasets, model nonlinear and non-stationary relationships, and incorporate high dimensional time-series data. In this study, we have explored the forecasting performance of econometric and neural network models in predicting the Dutch inflation rate. The macroeconomic factors used as predictors of inflation are the historical inflation rate, money supply, GDP, interest rate, unemployment rate and the gold price. The time period that is considered to train the models and forecast inflation is challenging, as it includes the effects of the financial crisis, COVID-19 and the Russian invasion of Ukraine. Especially as COVID-19 and the invasion are in the out of sample data set.

The findings of this study indicate that the neural network models outperform the econometric models in the in-sample forecasts. Specifically, the LSTM model shows the best in-sample performance, suggesting that it is the most accurate model for predicting inflation based on historical data. However, when it comes to out-of-sample forecasting, the econometric models perform better, indicating that they can better generalize to future data. The RNN and LSTM model also did not perform better than the naive predictor out of sample. The challenging nature of the dataset posed significant difficulties for the RNN and LSTM models in their out-of-sample forecast. These extreme events caused sudden and significant shocks to the economy, resulting in rapid changes in market dynamics that the RNN and LSTM models struggled to adjust to. Furthermore, the RNN and LSTM models may not have been able to capture the complex interactions between economic variables during such unprecedented events, as they rely on past patterns in data to make predictions. It is worth noting that the study highlights that a more complex model does not necessarily result in better and more accurate performance. This is evident from the performance of the RNN and LSTM models, which, despite their complexity, did not perform as well as the simpler econometric models in out-of-sample forecasting. This suggests that the appropriate model choice should depend on the specific context and dataset being analyzed, rather than simply selecting the most complex model available.

However, there are several limitation of this research that could be further investigated. Future research should explore the use of other machine learning models, as only the forecasting performance of two econometric models and three NN have been examined. Evaluating other models could be of interest to achieve an even more reliable result. Furthermore, the performance of neural networks could also be tested with different architectures, for example by modifying the the number of features, or enlarging the number of hidden layers. Another interesting possibility is to investigate the performance of the models on different data frequencies, inflation predictors and forecast horizons.

References

Almosova, A. and Andresen, N. (2023). Nonlinear inflation forecasting with recurrent neural networks. *Journal of Forecasting*, 42(2):240–259.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Angelina, S. and Nugraha, N. M. (2020). Effects of monetary policy on inflation and national economy based on analysis of bank indonesia annual report. *Technium Soc. Sci. J.*, 10:423.

Arsene, M. F. and Guy-Paulin, D. D. (2013). An econometric analysis of the nexus between credit to the private sector, inflation and economic growth: Case of cameroon 1965 a 2010. *Global Journal of Management and Business Research*, 13(G7):41–53.

Auray, S., Eyquem, A., and Poutineau, J.-C. (2009). Inflation targets in a monetary union with endogenous entry. *Cahier de recherche/Working Paper*, 9:02.

Baldwin, R. (2018). The great convergence. In *The Great Convergence*. Harvard University Press.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

Binner, J. M., Bissoondeeal, R. K., Elger, T., Gazely, A. M., and Mullineux, A. W. (2005). A comparison of linear forecasting models and neural networks: an application to euro inflation and euro divisia. *Applied economics*, 37(6):665–680.

Blanchard, O. J. and Gali, J. (2007). The macroeconomic effects of oil shocks: Why are the 2000s so different from the 1970s?

Box, G. and Jenkins, G. (1970). *Time Series Analysis: Forecasting and Control*. Holden-Day series in time series analysis and digital processing. Holden-Day.

Breuel, T. M. (2015). The effects of hyperparameters on sgd training of neural networks. *arXiv preprint arXiv:1508.02788*.

Brigham, E. F. and Ehrhardt, M. C. (2013). *Financial management: Theory & practice*. Cengage Learning.

Bucci, A. (2020). Realized volatility forecasting with neural networks. *Journal of Financial Econometrics*, 18(3):502–531.

Cecchetti, S. G. and Moessner, R. (2008). Commodity prices and inflation dynamics. *BIS Quarterly Review*, pages 55–66.

Chollet, F. (2018). *Deep Learning with Python*. Manning Publications Co.

Cornand, C. and M'baye, C. K. (2018). Does inflation targeting matter? an experimental investigation. *Macroeconomic Dynamics*, 22(2):362–401.

Cornell, B. and Gerger, R. (2017). A note on estimating constant growth terminal values with inflation. *Business Valuation Review*, 36(3):103–105.

Cornell, B., Gerger, R., Jarrell, G. A., and Canessa, J. L. (2021). Inflation, investment and valuation. *Journal of Business Valuation and Economic Loss Analysis*, 16(1):1–13.

Damodaran, A. (2022). Musings on markets. Accessed January 14, 2023.

Dematos, G., Boyd, M. S., Kermanshahi, B., Kohzadi, N., and Kaastra, I. (1996). Feed-forward versus recurrent neural networks for forecasting monthly japanese yen exchange rates. *Financial Engineering and the Japanese Markets*, 3(1):59–75.

Diewert, W. E. (2002). Harmonized indexes of consumer prices: Their conceptual foundations. *Available at SSRN 357342*.

Dräger, L., Gründler, K., and Potrafke, N. (2022). Political shocks and inflation expectations: Evidence from the 2022 russian invasion of ukraine.

Ehrhardt, M. C. and Brigham, E. F. (2013). *Corporate finance: A focused approach*. Cengage Learning, Boston, MA, 5th edition.

Eickmeier, S. and Hofmann, B. (2022). What drives inflation? disentangling demand and supply factors.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Enders, W. (2014). *Applied Econometric Times Series*. Wiley.

European Central Bank (2022). Monetary policy measures. `https://sdw.ecb.europa.eu/`.

Federal Reserve (2022). Monetary base. `https://fred.stlouisfed.org/series/WM2NS`.

Fisher, I. (1930). *Theory of interest: as determined by impatience to spend income and opportunity to invest it*. Augustusm Kelly Publishers, Clifton.

Fritzer, F., Moser, G., and Scharler, J. (2002). Forecasting austrian hicp and its components using var and arima models. Technical report, Working paper.

Fulton, C. and Hubrich, K. (2021). Forecasting us inflation in real time. *Econometrics*, 9(4):36.

Ghasemi, A. and Zahedi, F. (2010). Survey of the applications of mean absolute error. *World Academy of Science, Engineering and Technology*, 4:496–502.

Ghosh, D., Levin, E. J., Macmillan, P., and Wright, R. E. (2004). Gold as an inflation hedge? *Studies in Economics and Finance*, 22(1):1–25.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press.

Graves, A. (2012). Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45.

Gurney, K. (2018). *An introduction to neural networks*. CRC press.

Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press, Princeton, NJ.

Heenan, G., Peter, M., and Roger, S. (2006). Implementing inflation targeting: institutional arrangements, target design, and communications.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Işığıçok, E., Öz, R., and Tarkun, S. (2020). Forecasting and technical comparison of inflation in turkey with box-jenkins (arima) models and the artificial neural network. *International Journal of Energy Optimization and Engineering (IJEOE)*, 9(4):84–103.

Iversen, J., Laséen, S., Lundvall, H., and Soderstrom, U. (2016). Real-time forecasting for monetary policy analysis: The case of sveriges riksbank. *Riksbank Research Paper Series*.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*. Springer.

Kasuya, M., Tanemura, T., et al. (2000). Small scale bayesian var modeling of the japanese macro economy using the posterior information criterion and monte carlo experiments. Technical report, Bank of Japan.

Khalid, A. M. (2005). Economic growth, inflation, and monetary policy in pakistan: Preliminary empirical estimates. *The Pakistan Development Review*, pages 961–974.

Lack, C. et al. (2006). Forecasting swiss inflation using var models. Technical report, Swiss National Bank.

Liu, L.-M., Hudak, G. B., Box, G. E., Muller, M. E., and Tiao, G. C. (1992). *Forecasting and time series analysis using the SCA statistical system*, volume 1. Scientific Computing Associates DeKalb, IL.

Lütkepohl, H. (2005). *New Introduction to Multiple Time Series Analysis*. Springer, Berlin.

Mallik, G. and Chowdhury, A. (2001). Inflation and economic growth: evidence from four south asian countries. *Asia-Pacific Development Journal*, 8(1):123–135.

Meyler, A., Kenny, G., and Quinn, T. (1998). Forecasting irish inflation using arima models.

Mishkin, F. S. (2001). *From monetary targeting to inflation targeting*. Number 2684. World Bank Publications.

Montgomery, D. C., Jennings, C. L., and Kulahci, M. (2015). Introduction to time series analysis and forecasting.

Moshiri, S. and Cameron, N. (2000). Neural network versus econometric models in forecasting inflation. *Journal of forecasting*, 19(3):201–217.

Muktadir-Al-Mukit, D., Shafiullah, A., and Ahmed, M. R. (2013). Inflation led import or import led inflation: Evidence from bangladesh. *Asian Business Review*, 2(2):65–69.

Nielsen, M. A. (2015). *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA.

Noh, S.-H. (2021). Analysis of gradient vanishing of rnns and performance comparison. *Information*, 12(11):442.

Ofori, C. F., Danquah, B. A., and Zhang, X. (2017). The impact of money supply on inflation, a case of ghana. *Imperial Journal of Interdisciplinary Research*, 3(1):2312–2318.

Ogunc, F., Ozmen, M. U., Sarikaya, C., et al. (2018). Inflation dynamics in turkey from a bayesian perspective. Technical report.

Öner, C. (2012). Inflation: Prices on the rise. *International Monetary Fund*.

Papavangjeli, M. (2019). Forecasting the albanian short-term inflation through a bayesian var model. Technical report, The Graduate Institute of International and Development Studies ....

Phillips, A. W. (1958). The relation between unemployment and the rate of change of money wage rates in the united kingdom, 1861–1957. *Economica*, 25(100):283–299.

Ramakrishnan, U. and Vamvakidis, A. (2002). Forecasting inflation in indonesia.

Raschka, S. (2015). *Python machine learning*. Packt Publishing.

Rashid, A. and Saeed, M. (2017). Firms' investment decisions–explaining the role of uncertainty. *Journal of Economic Studies*, 44(5):833–860.

Reis, R. (2016). Funding quantitative easing to target inflation.

Rocheteau, G., Wright, R., and Xiao, S. X. (2018). Open market operations. *Journal of Monetary Economics*, 98:114–128.

Rodríguez-Vargas, A. (2020). Forecasting costa rican inflation with machine learning methods. *Latin American Journal of Central Banking*, 1(1-4):100012.

Ross, S., Westerfield, R., and Jordan, B. (2003). *Fundamentals of Corporate Finance*. McGraw-Hill Irwin.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Sang, C. and Di Pierro, M. (2019). Improving trading technical analysis with tensorflow long short-term memory (lstm) neural network. *The Journal of Finance and Data Science*, 5(1):1–11.

Šestanović, T. and Arnerić, J. (2021). Can recurrent neural networks predict inflation in euro zone as good as professional forecasters? *Mathematics*, 9(19):2486.

Sharma, S., Sharma, S., and Athaiya, A. (2017). Activation functions in neural networks. *towards data science*, 6(12):310–316.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Stinchcombe, M. and White, H. (1992). Using feedforward networks to distinguish multivariate populations. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 1, pages 788–793. IEEE.

Tsay, R. (2010). *Analysis of Financial Time Series*. John Wiley & Sons.

Ulke, V. and Ergun, U. (2011). Econometric analysis of import and inflation relationship in turkey between 1995 and 2010. *Journal of Economic and Social Studies*, 1(2).

Vapnik, V. N. (1999). *The Nature of Statistical Learning Theory*. Springer, second edition.

Whittle, P. (1951). *Hypothesis testing in time series analysis*, volume 4. Almqvist & Wiksells boktr.

Yadav, O., Gomes, C., Kanojiya, A., and Yadav, A. (2019). Inflation prediction model using machine learning. *International journal of information and computing science*, 6(5):121–128.

Young, S. R., Rose, D. C., Karnowski, T. P., Lim, S.-H., and Patton, R. M. (2015). Optimizing deep learning hyper-parameters through an evolutionary algorithm. In *Proceedings of the workshop on machine learning in high-performance computing environments*, pages 1–5.

Yu, W., Li, X., and Gonzalez, J. (2019). Fast training of deep lstm networks. In *International Symposium on Neural Networks*, pages 3–10. Springer.

Zhang, G., Patuwo, B. E., and Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, 14(1):35–62.

Zhang, X. and Yang, J. (2015). A survey on multi-objective optimization methods. *Mathematical Problems in Engineering*, 2015.