Master's thesis

NTNU
Norwegian University of Science and Technology

Tormod Rygh
Camilla Vaage

The Future of Inflation Forecasting: LSTM Networks vs. Traditional Models for Accurate Predictions

Can LSTM networks be improved to compete with univariate time series models like SARIMA?

Master's thesis in Financial Economics
Supervisor: Sjur Westgaard
Co-supervisor: Petter Eilif de Lange

June 2024

NTNU
Norwegian University of
Science and Technology

Tormod Rygh
Camilla Vaage

The Future of Inflation Forecasting: LSTM Networks vs. Traditional Models for Accurate Predictions

Can LSTM networks be improved to compete with univariate time series models like SARIMA?

Master's thesis in Financial Economics
Supervisor: Sjur Westgaard
Co-supervisor: Petter Eilif de Lange
June 2024

Norwegian University of Science and Technology

NTNU
Norwegian University of
Science and Technology

,

ABSTRACT

The purpose of this study is to examine if neural networks can contribute to improved forecasts of macroeconomic variables. Specifically, we investigate if the long short-term memory (LSTM) model can produce better forecasts than a range of machine learning models like LASSO, and univariate time series models like SARIMA. We attempt to improve on existing work on LSTM by applying different feature selection approaches and changing the training data of the model. We conduct Root Mean Square Error (RMSE) tests on all models, finding the loss of each model with respect to inflation. We compare the results of our LASSO-LSTM model to LSTM with PCA feature selection. We also test the performance of LASSO-LSTM when data augmentation is applied to training data, using the Moving Block Bootstrapping (MBB) method. LASSO-LSTM is further tested using a different dataset consisting of financial variables from the EIKON database. We find that forecasts of LASSO-LSTM generally perform better than LSTM with PCA feature selection. Despite its strengths LSTM often underperforms univariate models and other machine learning models. More interpret-able models like LASSO generally yield better forecasts than LSTM, and seem more suitable for central bankers and policy makers.

Målet med denne studien er å undersøke om nevrale nettverk kan bidra til forbedrede prognoser av makroøkonomiske variabler. Mer spesifikt så skal vi undersøke om long short-term memory (LSTM) modellen kan produsere bedre prognoser enn maskinlæring modeller som LASSO, og univariate tidsserie modeller som SARIMA. Vi forsøker å forbedre tidligere arbeid gjort på LSTM ved å bruke forskjellige variabelvalg-tilnærminger og ved å forandre på treningsdataen til modellen. Vi utfører en Root Mean Square Error (RMSE) test på alle modellene, og finner tapet til de respektive modellene med hensyn til inflasjon. Vi sammenligner resultatene til LASSO-LSTM modellen med LSTM med PCA variabelvalg. Vi tester også prestasjonen til LASSO-LSTM når dataforsterkning er anvendt på treningsdataen ved hjelp av Moving Block Bootstrapping (MBB) metoden. LASSO-LSTM er videre tested ved hjelp av ulike datasett bestående av finansielle variabler fra EIKON databasen. Vi finner at prognoser lagd av LASSO-LSTM generelt er bedre enn PCA-LSTM. På tross av sine ferdigheter så underpresterer LSTM ofte sammenlignet med univariate modeller og andre maskinlæringsmetoder. Mer tolkbare modeller som LASSO gir generelt bedre prognoser enn LSTM, og virker mer passende som verktøy for sentralbanker og beslutningstakere.

PREFACE

This paper is the result of the work on a Master's thesis in financial economics at the Department of Economics (ISØ) at the Norwegian University of Science and Technology (NTNU) in Trondheim.

We would like to thank our supervisors, Sjur Westgaard and Petter Eilif de Lange, for their valuable guidance and support on writing this thesis. It has been very helpful having someone with such great insight into the problem area to help us refine our thoughts and ideas into a presentable thesis. Their great enthusiasm for our thesis has truly inspired us. We would also like to thank Roderick McCrorie at the University of St.Andrews for being a great interlocutor and providing valuable input for this paper. His insights have helped us understand the broader implications of our work. We would like to thank Livia Paranhos at the Bank of England for her feedback on model calibration and encouraging us to explore data augmentation. Lastly, we would like to thank our friends, families and classmates for their motivation and support throughout this process.

Trondheim, June 2, 2024

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

LSTM Long short term memory

LASSO Least absolute shrinkage and selection operator

LASSO-LSTM Least absolute shrinkage and selection operator - Long short term memory

SARIMA Seasonal autoregressive integrated moving average

ARIMA Autoregressive integrated moving average

AR(p) Autoregressive model

NN Neural network

CNN Conveluted neural net

RNN Reccurent neural net

RF Random forest

GFC Gloabal financial crisis

PCA Principal component analysis

RMSE Root mean square error

MBB Moving block bootstrap

EFS Exhaustive feature selection

PRM Penalized regression method

MLE Maximum likelihood

AIC Akaike information criteria

BIC Baysian information criteria

CHAPTER ONE

INTRODUCTION

Today most central banks aim at providing monetary and financial stability to their domestic economy, as well as ensuring that the financial system operates in the best interest of the wider economy [1]. To do this, maintaining low and stable inflation and high employment is crucial, and one of the reasons why most central banks operate with an inflation target. In the last couple of years inflation has been increasing quite rapidly worldwide, but it is now forecast to decline [2]. Advanced economies are returning to their inflation targets quicker than developing economies and emerging markets, mostly due to them having stronger economic fundamentals. High inflation causes a host of problems for the economy, and it is in a countries' best interest to keep inflation under control. Forecasting inflation is thus a core part of the work done by central banks, and accurate forecasts are necessary to be able to set interest rates to a level trading off the inflation target against economic activity [2].

Inflation affects all agents operating in the economy. An increase in the rate of inflation will reduce the real rate of return on money and assets in general. Hyperinflation makes budgeting and investment planning difficult as it is nearly impossible to establish proper discount factors. This in turn, negatively affects the performance of the financial sector and long-run real activity, and resource allocation will be less efficient. A hostile environment for capital investments makes it harder for businesses to function and grow, and has a negative effect on long-run economic performance, as well as equity market activity. Furthermore, high inflation can result in tandem increases of stock values, which disappear when inflation is low [3]. An increase in inflation can also have a discouraging effect on investments in research and development[4].

Phillips curve models have been widely used for inflation forecasting, and the relationship between unemployment and inflation is a central theme in macroeconomics. This relation has been under a lot of scrutiny, and some researchers suggests that the inclusion of other indicators like those of Stock and Watson, as well as commodity prices and financial data, can be more effective in predicting inflation [5, 6, 7, 8, 9]. Phillips curve models are deemed as outdated by some, which has created the opportunity for other models to appear on the scene.

In the literature on inflation forecasting, there is a wide use of multivariate models attempting to enhance the performance of forecasting. An often used data set, as mentioned above, are the variables applied by Stock and Watson from 1993. Using these time series for multivariate models, it is possible to perform forecasting based on significant factors actually contributing to inflation. This dataset has been employed in several studies following the authors' findings, and we add financial data to the mix hopefully capturing a little more of the information explaining inflation.

It is challenging to find actual out of sample models able to outperform simple univariate models [10]. More recent work has attempted to improve on forecasting by applying non-linear machine learning models. Results from previous work have been mixed, and long-short term memory (LSTM) [11] and Quantile Random Forest models [12] have not been able to consistently outperform univariate models. Specifically the SARIMA model tends to perform better, or on par, with with non-linear approaches. Machine learning methods have proved useful to forecasting, yet there is still room for improvement since there is no significant benefit compared to univariate models [11].

In this study we attempt to produce reliable inflation forecasts for the US using LSTM networks, which have some desirable properties including long-term memory and an ability to capture long term trends that will be further discussed in chapter 4.2 [13]. We attempt to improve LSTM forecasting capability by applying feature selection approaches and changing the training data of the model. We use LASSO for data selection to pick the most important indicators and discard irrelevant information. We compare the results to some benchmark models like SARIMA and AR(p), which have performed well in previous research [11]. We also use other machine learning methods, like random forest, ridge, PCA-LSTM and LASSO, as benchmarks.

We believe that by combining LASSO and LSTM networks we can increase the interpretability of LSTM, reduce the chances of overfitting, as well as take advantage of LSTMs ability to explain non-linearity's and capture complex relationships. We also believe that the choice of variables is of great importance when conducting inflation forecasts. That is why we also have dedicated considerable effort into researching the effect of different types of variables. In addition to the classic Stock and Watson variables, we are including some financial variables, mostly prices, that we think might improve the performance of our forecast.

The main objectives of this work is:

1. Accuracy assessment: To asses the accuracy of machine learning methods, like LSTM networks and LASSO. This will be done by using metrics such as root mean squared error (RMSE).

2. Model comparison: To compare the performance of traditional econometric methods like SARIMA with modern machine learning methods like LSTM networks in forecasting inflation.

3. Policy implications: To examine how the combination of LASSO and LSTM

networks, and including financial data, can enhance the ability of policymakers to anticipate inflationary trends and make good decisions.

4. Financial impact: To investigate what implications our results have for the economy as a whole, focusing on the financial sector and banking.

The remaining part of this work is organized as follows: In section 2 we review relevant literature. In section 3 we briefly describe our data, and offer some key descriptive statistics. In section 4 we briefly describe our models and benchmark models, as well as specifications related to network training and evaluation methodology. In section 5 we present and discuss our results, and in the last section we provide a summary of the main points of the paper.

CHAPTER

TWO

LITERATURE REVIEW

2.1 Inflation forecasting using classical/linear methods

Inflation targeting regimes was introduced to many countries in the late 1980s and early 1990s. Since then, monetary policy decisions have been based of forecasts of key variables, inflation being especially important. Most central banks regularly publish forecasts of a number of variables to support and motivate their monetary policy decisions. To produce these forecasts, central banks usually rely on different forecasting models like VAR, DSGE, factor models and short-term indicator models[1].

Iversen et al [1] analyzed forecasts made in 2007 to 2013, which was a period of financial crisis and global recession, followed by a slow recovery in several parts of the world. Iversen et al. studied Swedish inflation, and during this period the Riksbank consistently overestimated the rate at which inflation would return to target, as well as the pace at which the repo rate would be raised towards normal levels. Therefore, the published inflation and repo rate forecasts display a bias. They found that the inflation and repo forecasts from the DSGE model displayed a similar bias, while forecasts from the BVAR model had smaller bias[1]. Charmeza and Ladley [14] used empirical analysis to show that inflation forecasts produced for monetary policy councils in inflation targeting countries might be subject to bias towards the inflation target.

The accuracy of inflation forecasts can vary, and some argue that the likelihood of accurately predicting a change in inflation using modern inflation forecasting models is small [6]. As a reaction to this statement Fisher et al. [15] reexamined the results of Atkeson and Ohanian [6] and showed that it might be possible to forecast inflation over specific horizons and in certain periods. Their study was conducted using standard Phillips-curve-based inflation forecasting models [15].

Based on conventional wisdom, the modern Phillips curve-based models are perceived as useful tools for inflation forecasting [16]. Modern Phillips curve equations relates the current unemployment rate to future changes in the inflation rate. This is based on the idea that there is a baseline rate of unemployment were inflation remains constant. When unemployment is below the baseline rate, inflation tends to rise over time, and vice versa. This baseline unemployment rate is called the non-accelerating inflation rate of unemployment, or "the NAIRU". The NAIRU Philips curve is widely used to produce inflation forecasts in academic literature, as well as in policy-making institutions [17].

Atkeson and Ohanian challenge this conventional view, and argue that The Phillips curve doesn't explain the nonlinear dynamics displayed by inflation, despite its theoretical appeal. They compared one-year inflation forecasts from Phillip curve models to forecasts from a naive model. The naive model makes the simple prediction that "at any date, the inflation rate over the coming year is expected to be the same as the inflation rate over the past year." [6]. The results of this study is that the NAIRU Phillips-curve based inflation forecasts had been no more accurate than the forecast of the naive model. They conclude that NAIRU Phillips curve models are not useful for inflation forecasting [6].

Fisher et al. [15] focus on the ability to forecast inflation in variations of the Consumer Price Index over a sample period 1985-2000. They find that Phillips curve models forecast the direction of inflation changes quite well across time, but are less suited to forecasting the magnitude of inflation changes. They further suggest that the periods which are difficult for inflation forecasting are those associated with changes in monetary policy regimes.

They suspected that periods of low inflation volatility and periods after regime shifts favor a naive model. The Phillips curve models did poorly for one-year-ahead forecasts, but it did well for two-year-ahead forecast. In general The Phillips curve models performed poorly, and it is not able to forecast the magnitude of inflation accurately. At the same time, policymakers are aware of the flaws of inflation forecasting, and therefore pay more attention to the direction of the change of future inflation. Because of this, Fisher et al. does not evaluate forecast performance by magnitude as the only criteria. Instead, they also analyze how the Phillips curve models predict that inflation will change in the future compared to the current level of inflation. With this new criteria the Phillips curve model performs quite well. Over the entire 1977-2000 period it is able to forecast the direction correctly one year ahead 60-70 percent of the time and more than 70 percent for the two year ahead forecast. The results suggested that the Phillips curve models forecasts the direction of inflation changes quite well across time, but it is less consistent with forecasting the magnitude of inflation changes. Their conclusion is that it is possible to forecast inflation accurately in some periods, but not in others. The periods which it is difficult to predict inflation are those associated with changes in monetary policy regimes. This implies that in a stable monetary regime that is expected to persist, it is sensible for policymakers to pay attention to inflation forecasts [15].

Meyler et al. [18] used ARIMA models to forecast Irish inflation. They emphasize forecasting performance and minimising out-of-sample forecast error, rather than maximizing in-sample goodness of fit. They mention a few weaknesses of ARIMA forecasting. Firstly, some of the traditional model identification techniques are subjective, and the reliability of the model might depend on the qualifications of the forecaster. Secondly, the economic significance of the model is not clear, since it is not embedded in an underlying theoretical model or structural relationship. Lastly, ARIMA models are "backwards looking", which makes them poor at predicting turning points unless this represents a return to long-run equilibrium [18]. At the same time, ARIMA models tend to perform very well on short-run inflation forecasting, and often outperform more sophisticated models [11]

Robinson [19] found that the Vector Autoregressive Regressive (VAR) model was better suited for inflation forecasting than other models. VARs are a multivariate forecasting approach also used by the Federal Reserve Bank and the Bank of England for forecasting economic trends. The VAR approach is a convenient way of identifying small selections of economic variables that appear to have been highly correlated with inflation in the past, and might be useful in forecasting future inflation [19].

2.2 Inflation forecasting using deep learning

Atkeson and Ohanian [6] motivated researchers to pursue new methods, and not Phillips curve based models. Some believe deep learning methods have the potential to generate more accurate out-of-sample inflation forecasts than standard models reviewed in the literature [20]. There are quite a few papers using various deep learning methods to predict inflation. Nevertheless, the usual conclusion is that some deep learning models perform well, but they are rarely superior to classical methods [11].

A paper by The Bank of England [11] applies LSTM to forecast inflation, and compares it to more traditional models in the inflation forecasting literature, as well as other machine learning models like LASSO, Random Forest, Ridge and Elastic-net. The authors found that LSTM showed competitive forecasting results both compared to other machine learning methods and traditional benchmarks, however, these results were not outstanding. At long horizons LSTM seemed to produce good forecasts, probably due to its ability to model long term trends and periods of instability, as it is relatively insulated from sudden and short-lived movements in inflation. The paper reveals LSTM to be an interesting tool for reducing dimensionality of the data in a way that is relevant for prediction. The estimated LSTM factors, using the US FRED-MD database, seemed to capture the underlying inflation trend well. The factors also exhibited high correlation with business cycle indicators, especially the output gap, which indicates that such signals are useful inflation predictors. They also did a variance decomposition analysis which revealed the LSTM factors to be significantly loaded on corporate bond spreads and housing starts, and the results generally align with the literature on common inflation predictors [11].

Theoharidis et al. [20] on the other hand, think machine learning methods have the potential to deliver superior performance compared to traditional econometric approaches. Their results suggests that macroeconomic forecasting can take advantage of deep learning models, when encountering nonlinearities and nonstationarity. The main problem for simple univariate econometric models is the nonlinear dynamics displayed by inflation, making the standard linear Philips curve inadequate. Sources for non-linearities that have been identified and documented are nominal rigidity, zero lower bound for interest rates, economic uncertainty, and fixed costs. They emphasize the importance of the choice of variables that can systemically be used for prediction, as well as delivering reliable out-of-sample forecasts. Without tools and criteria to filter them and achieve parsimony, one becomes prone to data mining biases and overfitting [20].

To address these challenges Theoharidis et al. examine several machine learning models, evaluating their performance in forecasting inflation. They combine ConvLSTM networks for their forecasting abilities and ability to take advantage of hierarchical patterns that might exist in the data, and VAEs for dimension reduction. This results in a VAE-ConvLSTM model, which according to their results is superior to 25 benchmarks, including SARIMA, MA, LASSO, LSTM, Ridge and Bayes reg. in terms of out-of-sample accuracy for several forecasting periods [20].

Studying Brazilian CPI inflation Garcia et al. [21] use high-dimensional and machine learning models to forecast real-time inflation, and find that LASSO performs well in data-rich environments. They use shrinkage models like LASSO, as well as AR models and random walk forecasts as benchmarks. They estimated forecasts for horizons between five days before the CPI index is published to 11 months plus five days (12 forecasts in total). LASSO based methods seemed to perform best for short horizons, which Garcia believed to be due to the expert survey forecasts they used as potential candidate predictors. These forecasts are precise in the beginning, but lose their predictive power as other factors become more important [21].

2.3 The importance of inflation forecasting in financial markets

Boyd et al. [3] takes a closer look on the impact of predictable inflation on the financial system, as there is a lot of literature explaining how the financial system influences long-run rates of economic growth. They found that for countries with low-to-moderate rates of inflation, there is a very strong negative association between inflation and financial intermediary development. When inflation rates rise, the partial correlation between intermediary activity and inflation falls. These results align quite well with the theoretical predictions, that predictable increases in the inflation rate interfere with resource allocation and economic growth[3].

The long-run relationship between financial market activity and inflation has been extensively investigated empirically. At sufficiently high levels of inflation there is a negative long-run relationship between inflation and real economic performance. There is a positive correlation between economic performance and the volume of bank lending activity, the quantity of bank liabilities issued, and the volume of trading in equity markets. At low-to-moderate long-run inflation rates there is a strong negative correlation between inflation and the volume of bank lending activity, the quantity of bank liabilities issues, and the volume of trading in equity markets. At higher inflation rates these partial correlations essentially disappear

[22].

The effect of a permanent increase in the inflation rate for long-run activity seems to be complicated, and probably depends strongly on the initial level of inflation. For countries with low-to-moderate inflation rates, there is a negative correlation between real equity returns and inflation. If the inflation is sufficiently high in a country, this correlation disappears. As economies develop, equity markets tend to become more important relative to banks [3].

2.4 Relevant variables in inflation forecasting

Stock and Watson [23] showed that many production-related variables are useful predictors of US inflation, but Atkeson and Ohanian [6] showed that the Philips curve fails to beat even simple naive models in many cases. This inspired researchers to investigate a range of different models and variables in order to improve inflation forecasts. Inflation is a complex phenomenon than affects several parts of the economy, as mentioned in the previous section. Because of this, choosing the relevant variables can be difficult, and there is always the risk of overfitting and unnecessary noise in the model. Still, several researchers have studied the effect of different types of variables and their relation to inflation.

Stock and Watson [23] explored whether asset prices could be useful in predicting inflation, and found that some asset prices predict inflation in some countries in some periods. Asset prices are forward-looking, which make them a class of potentially useful predictors of inflation. Here, asset prices are interpreted as including interest rates, spreads, returns, and other measures related to the value of financial or tangible assets (bonds, stock, housing, gold, etc.). It is a fundamental concept of macroeconomics, that asset prices and interest rates contain information about future economic developments. The literature on forecasting using asset prices has found that interest rates, term spreads, stock returns, dividend yields, and exchange rates are leading indicators of inflation or economic activity [23].

An advantage of using asset prices is that they are usually observed in real time with negligible measurement error. Monetary aggregates, on the other hand, require ongoing redefinition as new financial instruments are introduced. Stock and Watson evaluated the practical value of asset prices for short- to medium-term forecasting using data from developed countries like the US and the UK. They found that forecasts based on individual indicators are unstable, and if an indicator performs well in one period it is not guaranteed that is will do the same in other periods [23].

Forni et al. [7] found that financial variables help in forecasting inflation at all horizons. They divided different variables into six blocks, where financial variables were one of them. They then compared the different blocks, and found that excluding financial variables induces a deterioration of forecasting performance on all horizons. Some of the variables they evaluated were nominal and real interest rates, spreads, stock market prices and exchange rates[7].

Chen et al. [9] investigated the predictive power of commodity prices on inflation. They show that for five small commodity-exporting countries with inflation targeting monetary policies, world commodity price aggregates have predictive power for their CPI and PPI inflation. Commodity prices clearly outperformed the random walk, and they did not observe any forecast improvements by using high frequency data. They concluded that commodity indexes are collectively useful for predicting inflation, which is consistent with theories of price rigidity and gradual exchange rate pass-through [9].

Chen and Rogoff [24] and Amano and van Norden [25] also demonstrated that global commodity prices play a key role in driving the currency value of several major commodity-exporting countries. This is called the "commodity currency" phenomenon, and explain that world commodity price movements have predictive power for CPI inflation.

Groen et al. [8] use activity and expectations variables for inflation forecasting, applying Bayesian regression models. The benchmarks used are random walk, Ridge, and AR. They evaluate the accuracy of real-time inflation forecasting, and find that using a BMA model that allows for structural breaks in the error variance results in very accurate point and density forecasts.

Groen et al. divided the predictor variables into three groups The first group consisted of variables that provide information about either the degree of excess demand in the economy, or the real costs that firms face. Some variables in this group are GDP, PCE, real residential investment, import inflator, unemployment ratio, housing starts, real spot of oil, real food commodities price index, and raw material commodities. The other group consisted of variables that have information about the current of future state of the economy, like M2 monetary aggregate. These variables reflect household spending and firm expenditure, as well as information about the current monetary policy and liquidity in the economy. They also included data on the term structure of interest rates, since this contains information about business cycles, monetary policy, and inflation expectations. Inflation expectations are also included in the model [8].

Bernanke [26] comments on the relationship between monetary policy, inflation, and inflation expectations, and their implication for the Phillips curve. Historically, inflation expectations have had a great impact on actual inflation. Long-run inflation expectations vary over time, and are not perfectly anchored in real economies. If people decide prices and wages with reference to the rate of inflation they expect in the long run and inflation expectations respond less to variations in economic activity than before, this will result in inflation being relatively less sensitive to the level of activity. This implies flattening of the Philips curve, and aligns with the literature. [26].

Staiger et al. [5] suggests that one should deemphasize the non-accelerating inflation rate of unemployment (NAIRU) in public discourse about monetary policy. They found that even though there is a clear empirical relation in the Phillips curve, the NAIRU is imprecisely estimated, and forecasts of inflation are insensitive to the NAIRU. They found other leading indicators of inflation that they view as at least as good as unemployment. They used percentage growth in the personal consumption expenditure (PCE) price index, excluding expenditures on food and energy as a measure of core inflation, and percentage growth in the GDP as a measure of broad inflation. They used 69 business cycle indicators from Stock and Watson as a means of comparison with the unemployment rate as predictors. The capacity utilization rate in manufacturing, the national association of purchasing managers index of new orders, and the federal funds rate were some of the indicators of inflation that outperformed unemployment. They conclude that there exist several variables that are equally or more valuable than unemployment for predicting inflation [5].

CHAPTER

THREE

DATA

Data has been collected from the FRED-MD and the EIKON databases. The two datasets have been transformed separately depending on their characteristics. In this study FRED-MD is used to cover macroeconomic as well as some financial data, while EIKON covers only financial data. The two sets of data gives a broader coverage of inflation indicators.

3.1 Variable-selection

Adaptive LASSO has achieved great success in selecting relevant variables consistently, as well as estimating regression parameters efficiently. Because the square loss function is sensitive to heavy-tailed errors and outliers, adaptive LASSO might fail to produce reliable estimates for datasets with heavy-tailed errors or severe outliers, which are common in financial studies.

It is important to apply a first variable selection step to correctly explain the data and avoid unnecessary noise in statistics. It is also normal that the number of variables p is larger than the number of available samples n (p>n). LASSO is used to overcome this drawback, and is widely used due to its capability of reducing the dimension of the problem.

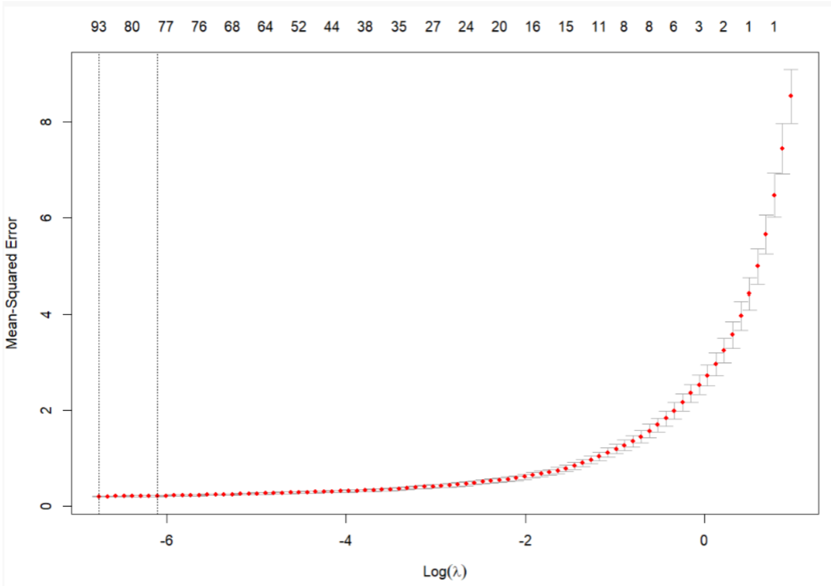

Figure 3.1.1: LASSO feature selection for a 12 month forecast

3.2 FRED-MD database

The FRED-MD dataset contains monthly US data compiled by McCracken and Ng. The data is most recently updated in December 2023. This data consists of 128 variables with 768 observations from January 1959 to December 2023. Before we can use this data to estimate our models, the dataset needs to undergo a series of transformations. To perform forecasting, it is necessary that the data is stationary. Variables that are non-stationary have been differenced such that $X_t - X_{t-12}$ represents the annual change in the variable. 74 variables are non-stationary and have been differenced. These variables are provided in the appendix. We have also treated the dataset for missing data, where missing observations have received the value of the previous observation.

3.3 EIKON database

Refinitiv EIKON is a well-regarded database that is used extensively in academia. It includes global economic, company and financial data. Data retrieved from this database includes some US commodities like gold, platinum, oil, gas and corn, as well as shipping rates and Baker Hughes. We retrieved monthly data covering observations from as early as January 1960 and until December 2023. In total we included and combined 27 financial variables from EIKON and FRED-MD to create a new dataset with financial data. Some descriptive statistics are presented in the table below. Further details on the data are included in the appendix.

Variable	Mean	Standard Deviation	1st Qu.	Median	3rd Qu.	Max
FEDFUNDS	2,74	0,80	0,41	2,27	2,88	0,19
GS1	3,58	0,46	0,59	1,06	2,96	3,02
GS10	3,01	0,78	1,11	2,09	2,86	6,60
TB3MS	2,72	0,12	2,46	0,38	1,12	2,90
TB6MS	2,12	0,64	0,29	2,30	4,76	6,18
AAA	3,23	0,14	2,51	3,72	2,24	4,29
BAA	2,19	0,94	0,29	1,27	4,23	1,35
CP3M	4,46	0,52	2,93	2,67	3,50	2,98
AAA	4,82	0,41	1,41	0,53	1,49	5,70
BAA	1,92	0,26	2,93	2,87	4,07	5,91
SP500	3,96	0,77	1,81	1,16	1,98	5,74
Oil (BFO-N)	40.92	30.39	9.0	36.0	66.0	99.0
Silver Spot Rate	45.33	30.65	17.0	44.0	72.0	98.0
USDA Daily Fresh 50%	44.93	30.19	15.0	45.5	69.0	99.0
Gold Spot Rate	270.09	1127.13	5.0	8.0	61.0	9028.0
Baker Hughes Index	212.6	280.9929	1	1	393	982
ESALQ Brazil Coffee Price	2253	1456.635	1	1	1	209
WIL5000IND	2,64	0,46	2,22	0,73	4,41	6,53
DJIA	2,84	0,57	0,12	2,35	2,91	6,52
NASDAQCOM	4,63	0,02	0,85	0,08	4,41	4,31
VIXCLS	0,36	0,62	0,36	3,32	3,46	8,97
M1SL	0,44	0,61	0,89	0,02	3,63	3,68
SPCS20RSA	0,10	0,62	0,36	2,71	2,51	4,36
MORTGAGE30US	4,16	0,94	0,95	1,08	4,78	8,92
DEXUSEU	3,89	0,68	1,24	2,94	3,22	8,06
CPIAUCSL	4,35	0,36	0,19	3,85	2,12	7,04
CPILFESL	4,89	0,44	2,08	1,00	3,03	1,00

Figure 3.3.1: Descriptive statistics of financial data

Forni et al. and Chen et al. emphasized the importance of financial variables and commodity prices in inflation forecasting, and supports our belief that including these data can improve the accuracy of forecasting inflation.

CHAPTER

FOUR

MODELS

4.1 LASSO

LASSO, Least Absolute Shrinkage and Selection Operator, is a method for shrinkage and selection for regression and generalized regression problems. For complicated phenomenons like inflation, where datasets with a lot of variables are included, it is not unlikely that some of the explanatory variables are irrelevant. This can cause noise in the model, giving us a less accurate forecast. LASSO is used to select only the most important covariates, discarding irrelevant information and keeping the error of the prediction as small as possible [27].

LASSO combines properties from both subset selection and ridge regressions. This makes it able to produce interpretative models (like subset selection), and be as stable as a ridge regression. The lasso minimizes the residual sum of squares subject to the sum of the absolute value of coefficients being less than a constant. Because of this constraint LASSO tends to produce coefficients that are exactly 0, thus giving us interpretive models [28].

Tibshirani [28] defines the Lasso model as follows:

We have data

$$(x^i, y_i), i = 1, 2, ..., N$$

predictor variables

$$x^i = (x_{il}, ..., x_{ip})^T$$

and responses

$$y_i$$

We either assume that the observations are independent or that the y_is are conditionally independent given the x_{ij}s.

We assume that the x_ij are standardized so that:

$$\left(\frac{\sum_i x_{ij}}{N}\right) = 0, \quad \frac{\sum_i x_{ij}^2}{N} = 1$$

Letting $\hat{B} = (\hat{B}_1, ..., \hat{B}_p)^T$, the lasso estimate $(\hat{\alpha}, \hat{B})$ is defined by:

$$(\hat{\alpha}, \hat{B}) = arg \quad min[\sum_{i=1}^{N}(y_i - \alpha - \sum_j \beta_j x_{ij})^2]$$

Subject to

$$\sum_j |\beta_j| t.$$

Here $t \geq 0$ is a tuning parameter. Now, for all t, the solution for α is $\hat{\alpha} = \bar{y}$. We can assume without loss of generality that $\bar{y} = 0$ and hence omit α.

The parameter $t \geq 0$ controls the amount of shrinkage that is applied to the estimates. Let $\hat{B_j^0}$ be the full least squares estimates and let $t_0 = \sum |\hat{B_j^0}|$. Values of $t < t_0$ will cause shrinkage of the solutions toward 0, and some coefficients may be exactly equal to 0. For example, if $t = t_0/2$ the effect will be roughly similar to finding the best subset of size $p/2$. The design matrix does not need to be full rank [28]

The reason for including LASSO in our model is to tackle the problems of overfitting and optimism bias. A LASSO regression tries to identify variables and corresponding regression coefficients that constitute a model that minimize prediction error. This is done by imposing a constraint on the model parameters which shrinks the regression coefficient towards zero, forcing the sum of the absolute value of the regression coefficients to be less than a fixed value (λ).

$$(\hat{\alpha}, \hat{B}) = arg \quad min[\sum_{i=1}^{N}(y_i - \alpha - \sum_j \beta_j x_{ij})^2 + \lambda \sum_j |\beta_j|]$$

After the shrinkage, variables with regression coefficients equal to zero are excluded from the model[29].

To choose λ one uses an automated k-fold cross-validation approach. To obtain this the dataset is randomly partitioned into k sub-samples of the same size. When k-1 sub-samples are used for developing a prediction model, the remaining sub-sample is used to validate this model. This is done k times, with each of the k sub-samples in turn being used for validation and the other for model development. By combining the k separate validation results for a range of λ values and choosing the preferred λ one gets the results that are used to determine the final model.

An advantage of this technique is that one can reduce overfitting without restricting a subset of the dataset to be used exclusively for internal validation. A disadvantage of the LASSO approach is that one may not be able to reliably interpret the regression coefficients in terms of independent risk factors, since the focus is on the best combined prediction, and not on the accuracy of the estimation [29].

4.2 LSTM

The LSTM model is a variant of the recurrent neural network (RNN) [30]. Unlike other neural networks, a recurrent neural network updates by time step. This means that the model will adjust forecasts based on previous time steps. RNN models have proven particularly useful for data sensitive sequences such as time series analysis, natural language processing and sound recognition [31]. For example in the context of music recognition, it would be possible to observe a pattern in the sound, making it possible to predict what is to come next or which song it is [32]. For such models it is crucial that there is a pattern in the data, and that the sequence of the data anticipates later values.

The RNN model is able to update its memory based on previous steps and consider long term trends and patterns in the data [13]. Consider an abnormal drop in inflation for one month, deviating with previous time steps in the data. The RNN takes into account the underlying pattern in the data based on previous observations, and considers the fall in inflation as an abnormality. What makes inflation behavior abnormal, and which patterns the model detects to label the drop in inflation as abnormal, is inherently difficult to grasp.

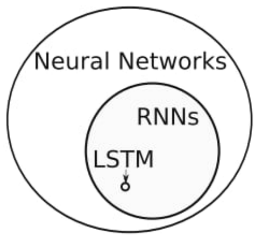

Figure 4.2.1: Classification of neural networks. LSTM is a specific type of neural networks within the group recurrent neural neural networks (RNN)[30].

LSTM on the other hand, differs from RNNs as it possesses an enhanced capability of capturing long term trends in the data [13]. Consider an inflationary event in the 1970s that has a similar pattern as one observed recently. The LSTM will see the similarities in pattern of the two events, and take this into consideration when making its next prediction. It is important to state that the event occurring in the 70s will not be fully weighted, but adjusted for short term events seen in the data. LSTM thus has the ability to consider both distant and recent events, when making its predictions[12].

LSTM has proven to be highly efficient for sequential data and has been used to compute univariate forecasts of monthly US CPI inflation. LSTM slightly outperforms autoregressive models (AR), Neural Networks (NN), and Markov-switching models, but its performance is on par with the SARIMA model [30]. Recently, it has become harder to outperform naive univariate random walk-type forecasts of US inflation, but since the mid-80s, inflation has also become less volatile and easier to predict. Atkeson and Ohanian [6] show that averaging over the last 12 months gives a more accurate forecast of the 12-month-ahead inflation than a backwards looking Phillips curve. Macroeconomic literature argues that the inflation process might be changing over time, making a nonlinear model more precise in predicting inflation. According to Almosova and Andresen [30] there are four main advantages of the LSTM method.

1. LSTMs are flexible and data-driven. It means that researchers don't have to specify the exact form of the non-linearity. Instead, the LSTM will infer this from the data itself.

2. Under some mild regulatory conditions LSTMs and neural networks of any type in general can approximate any continuous function arbitrarily accurately. At the same time, these models are more parsimonious than many other nonlinear time series models.

3. LSTMs were developed specifically for the sequential data analysis and have proved to be very successful with this task.

4. The recent development of the optimization routines for NNs and the libraries that employ computer GPUs made the training of NNs and recurrent neural networks significantly more feasible.

In contrast to classical time-series models, the LSTM-network does not suffer from data instabilities or unit root problems. Nor does it suffer from the vanishing gradient problem of general RNNs, which can destroy the long-term memory of these networks. LSTM may be applied to forecasting any macroeconomic time-series, provided that there are enough observations to estimate the model.

Theoretically, Convolutional Neural Networks (CNN), originally developed for images, could also be used for time series forecasting, if one treats the input as a one-dimensional image. LSTMs performs particularly well at long horizons and during periods of high macroeconomic uncertainty. This is due to their lower sensitivity to temporary and sudden price changes compared to traditional models in the literature. One should note that their performance is not outstanding, for instance compared to the random forest model [12]. Neural nets as well demonstrate competitive, but not outstanding, performance against common benchmarks, including other machine learning methods.

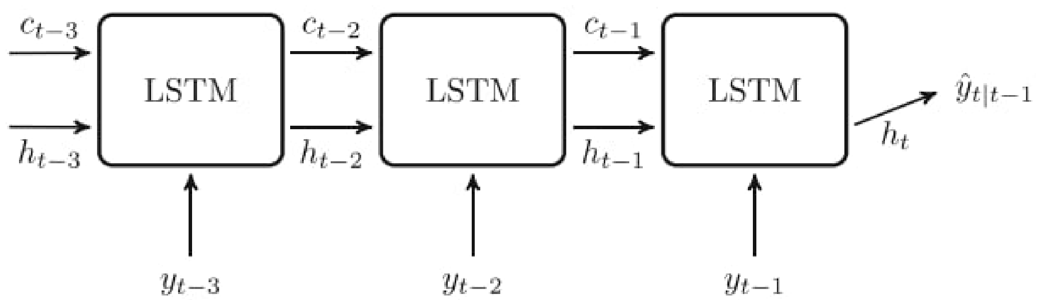

Figure 4.2.2: Representation of LSTM recurrent structure. LSTM has a cell state (c_t) and a hidden state (h_t). As t increases, more information (y) is put into the cell state and memory state. This new information in the cell and memory state contribute to the prediction (h)[30].

.

A common weakness of machine learning techniques, including neural networks, is the lack of interpretability [31]. For inflation in particular this could be a problem, since much of the effort is devoted to understanding the underlying inflation process, sometimes at the expense of marginal increases in forecasting gains. LSTM is on average less affected by sudden, short-lived movements in prices compared to other models. Random forest has proved sensitive to the downward pressure on prices caused by the global financial crisis (GFC). Machine learning models are more prone to instabilities in performance due to their sensitivity to model specification [30]. This also applies to the LSTM-network. Lastly, LSTM-implied factors display high correlation with business cycle indicators, informing on the usefulness of such signals as inflation predictors.

The LSTM model can be described by the cell state function and the internal memory. These functions start out with their initial value, before new information attained from new observations enter and impact the value of the function. We apply the sigmoid and tanh function, giving us the updated values of the internal memory and cell state.

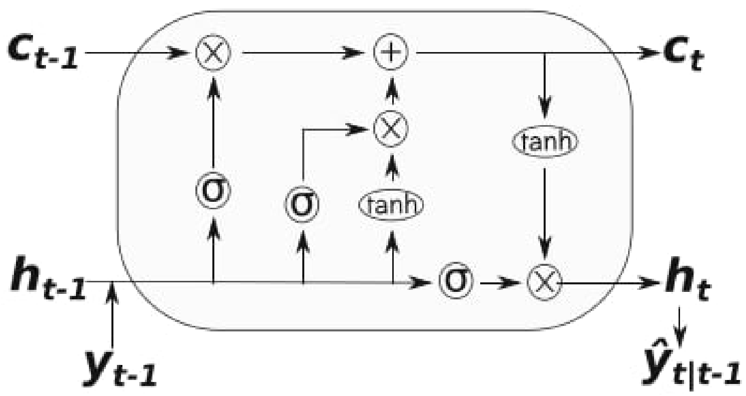

Figure 4.2.3: The figure illustrates the schematic of a LSTM cell. The cell state $c_{\ t-1}$ and hidden state $h_{\ t-1}$ from the previous time step, along with the current input y_{t-1}, are processed through forget, input, and output gates. The forget gate determines how much of the previous cell state should be retained, while the input gate decides how much new information should be added. These combined results update the cell state c_t. The output gate determines the next hidden state h_t, which, combined with the updated cell state, forms the output $y_{t|t-1}$. Activation functions like tanh and sigmoid are used to regulate the flow of information within the cell, ensuring that the LSTM effectively captures long-term dependencies in the data [30].

4.3 LASSO-LSTM

The LASSO-LSTM model is an integrated machine learning, neural network model. It integrates the strengths of LASSO and LSTM. The initial step is LASSO, for feature selection. Predictors are fitted, reducing errors of the residuals in a similar fashion to that of OLS. With LASSO a shrinkage parameter (λ) is applied to coefficients, shrinking the size of less significant predictors. The size of the hyper-parameter (λ) is important, as it decides the number of predictors that the LSTM model will be trained on.

The regularisation term of LASSO has the function of feature selection. The predictors that are determined to be most significant will not receive large penalties to their coefficients, rendering them important in the forecasting of inflation. The regularisation term is set to three sizes. In this study LASSO-LSTM is constructed with three sizes of architecture, large, medium and small. The regularisation term is set larger to make the LASSO-LSTM architecture smaller. This approach will contribute in assessing how to balance underfitting and overfitting, in the context of macro-economic forecasting.

When dealing with medium sized sample datasets and high dimensional data, the LSTM model, while known to handle dimensionality well, can run into problems of overfitting. Work done on LSTM for macroeconomic forecasting has shown that larger architectures do not necessarily outperform smaller ones [11]. Feature selection performed by LASSO detects the features that can contribute to the forecasting performance of the LSTM.

The features considered most important after regularisation, proceed to the LSTM input layer. Different sizes of architectures then have different amounts of layers. Larger architectures, more prone to overfitting, receive fewer layers of full connected nodes, and receive drop out layers. Smaller architectures, less prone to overfitting, can have more layers and/or fewer dropout layers. The LSTM layer structures can then be trained on forecasting inflation based on the number of predictors deemed most important by LASSO. The LASSO-LSTM model, as an augmented version of the LSTM model integrating feature selection, contributes to model regularization.

An alternative approach commonly used for feature selection, is principal component analysis (PCA) [13]. The two approaches deviate in their goals. PCA deems variables important based on variance. LASSO, by shrinking coefficients, retains the variables considered important. Thus LASSO-LSTM retains some interpretability, as forecasts are based on important factors, which is of interest to central banks in their decision making.

4.4 ARIMA and SARIMA

SARIMA, Seasonal Autoregressive Integrated Moving Average, is an extension to ARIMA that supports the direct modeling of the seasonal component of a time series. ARIMA does not support a time series with a repeating cycle, and it expects that data is either not seasonal or that the seasonal component is removed, for example via seasonal differencing [33].

ARIMA was introduces by Box and Jenkins in 1976, and uses the series past values to produce forecasts. It uses lagged and forecast error lags to predict future values. It is derived by general modification of an autoregressive moving average (ARMA) model [33].

An ARIMA(p, d, q) model can be represented by equation (1) below [33]:

$$\Delta^d Y_t = \beta_0 + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=1}^{q} \theta_j e_{t-j} + e_t \quad (1)$$

Here β_0 is a constant, ϕ_i are the coefficients of the autoregressive part with p lags, and θ_j are the coefficients of the moving average part with q lags. e_t is the error term at time t.

We have a linear combination of lags, and the aim is to identify the p, d, and q values. The minimum difference (d) is selected in the order by which the autocorrelation reaches zero. The p is determined by the order of the AR, and should be equal to the lags in the PAC which significantly crosses the limit set. Equation (2) shows the Partial Autocorrelation (PAC), where y is considered the response variable and x_1, x_2, and x_3 are the predictor variables. The PAC between y and x_3, (2), is calculated as the correlation between the regression residuals of y on x_1 and x_2 with the residuals of x_3 on x_1 and x_2.

$$PAC = \frac{\text{cov}(y, x_3 | x_1, x_2)}{\sqrt{\text{var}(y|x_1, x_2) \cdot \text{var}(x_3|x_1, x_2)}} \quad (2)$$

The h^{th} order partial autocorrelation can be represented as (3):

$$PAC^h = \frac{\text{cov}(y_i, y_{i-h}|y_{i-h+1}, \ldots, y_{i-1})}{\sqrt{\text{var}(y_i|y_{i-1}, \ldots, y_{i-h+1}) \cdot \text{var}(y_{i-h}|y_{i-1}, \ldots, y_{i-h+1})}} \quad (3)$$

The q is calculated based on the Autocorrelation (AC) and denotes the error of the lagged forecast:

$$AC = \frac{\sum_{i=1}^{N-k}(y_i - \bar{y})(y_{i+k} - \bar{y})}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \quad (4)$$

Here,

- $\bar{y}$: The mean of the time series
- k: The lag, where $k \geq 0$
- N: The complete series value

If one requires seasonal patterns in the time series, a seasonal term can be added, which produces a SARIMA model. This model can be written as (5):

$$ARIMA(p, d, q) * (P, D, Q)s \quad (5)$$

Here (p,d,q) represent the non-seasonal part, and (P,D,Q) represents the seasonal part of the model. S represents the period number in a season. In this study we employ SARIMA as we assume there exists seasonality in inflation data.

A seasonal ARIMA model uses differencing at a lag equal to the number of seasons (s) to remove additive seasonal effects. As with lag 1 differencing to remove a trend, the lag s differencing introduces a moving average term. The seasonal ARIMA model includes autoregressive and moving average terms at lag s. The trend elements can be chosen through careful analysis of AFF and PACF plots looking at the correlations of recent time steps. Similarly, ADC and PACF plots can be analyzed to specify values for the seasonal model by looking at correlation at seasonal lag time steps.

In short, SARIMA supports univariate time series data with a seasonal component, and adds three new hyper-parameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality. The reason for comparing NNs with SARIMA is that their celebrated performance might be due to their ability to capture seasonality. Consequently NNs should be compared to a linear seasonal model. Commonly economic time series variables change in a cyclical pattern with time, i.e., exhibit seasonality. In relation to inflation; sales, holidays, and production cycles can cause seasonal price variations that affect The Consumer Price Index.

According to most of the literature on inflation forecasting, SARIMA is the top performing classical model and usually outperforms VAR, AR and ARIMA [11]. Also, compared to the newer machine learning methods like Recurrent neural networks, LSTM and feed forward neural networks, SARIMA performs on par or better. This makes the SARIMA model a natural choice for our main benchmark, as we want to compare machine learning methods to classical methods, as well as look for ways to improve these methods.

4.5 Benchmark

To determine the performance of the LASSO-LSTM model, we employ standard benchmarks from the literature. These are the autoregressive model (AR(p)), the seasonal autoregressive integrated moving average (SARIMA), the random forest (RF) and the least absolute shrinkage and selection operator (LASSO). To evaluate out of sample performance, the models has been trained on two sets of data. Forecasts for the period of 2010 to 2023 has been trained on data from 1960 up to 2010. Forecasts for the period of 1997 to 2009 are produced by models trained on data for the period from 1960 to 1997.

Once forecasts have been made, we employ standard practice with RMSE tests to evaluate the performance of the forecast against actual values. We then compare the LASSO-LSTM model to the benchmark models using Diebol-Mariano test (DM).

4.6 Network training

4.6.1 LSTM

We start by splitting the data into training data, two validation sets, and an out-of-sample set. The training and validation data cover the period from 1960 to 1997, and are used to train the model. The out-of-sample data ranges from 2010 to the end of 2023.

The first step is the tuning of the model, and starts with a set of hyper-parameters. The LSTM applies these hyper-parameters and creates thousands of epochs, which are different versions of LSTM using the parameters. The epochs are tested on the first validation set, and the best tuned epoch is chosen. The chosen epoch is then tested on the second validation set. This procedure is repeated several times with different sets of hyper-parameters. All of the chosen epochs are then compared on the second validation set, and the best tuned one is selected and applied on the out-of-sample set. This procedure is the standard approach used in the literature [11]. Refer to Figure 4.6.1 for the specifications related to the different models.

Feature selection occurs ex-ante of the LSTM tuning, and is done using LASSO and PCA. The features for both LASSO-LSTM and PCA-LSTM are based on the training data and the first validation sample. The specification of the LSTM model can be divided into four distinct parts:

1) Feature selection: Features are selected based on their relevance to the data and the problem at hand. This is an independent step that occurs before training and optimization.

2) Model configuration: This involves setting a range of structure and parameters of the LSTM model, including the incorporation of lagged versions, the number and order of layers, the number of dropout layers, the dropout rate percentage, and the learning rate.

3) Training and optimization: This step includes setting the number of epochs, batch sizes, and validation strategies.

4) Model evaluation: This involves comparing different versions of the LSTM model and selecting the best one based on performance metrics.

Model specification
LSTM specification
Optimal choice of hyperparameters

	PCA LSTM	LASSO LSTM	EIKON LASSO-LSTM	AUG LASSO-LSTM
Stage 1				
Lags	12	24	24	24
Layers	3	3	3	3
Dropout layers	2	3	3	3
Layer order	FFFDD	FDFDFD	FDFDFD	FDFDFD
Dropout rate	0.1	0.1	0.1	0.1
Learning rate	Adam	0.001	Adam	Adam
Stage 2				
Epochs	300	300	300	200
Batch	128	128	128	128
Restore best weight:	Yes	Yes	Yes	Yes
Validation sample	0.2	0.2	0.2	0.2

The table reports the optimal choice of hyperparameters for the different LSTM models based on testing within the validation sample.
F = fully connected layer, D = dropout layer

Figure 4.6.1: The table presents the optimal specifications for the applied LSTM models, showing the best values for hyper-parameters such as lags, layers, dropout layers, dropout rate, learning rate, epochs, batch size, and validation sample

4.6.2 Other machine learning models

For the other machine learning models the data is split into three parts; training, validation and out-of-sample. Data from 1960 to 2010 is used for the training data and validation data.

The first step is embedded feature selection, where features are selected to be included in the model using training data and the validation sample. This process is crucial for the Random Forest algorithm. Various hyper-parameters are tested to identify the best-performing version of the model. These hyper-parameters remain constant for all forecasts once selected.

Unlike the other models, Random Forest has been specified to sequentially update. This means that each forecast utilizes all available data up to a certain point in time. As new forecasts are made, more data is incorporated into the model. The process of sequential updating allows the Random Forest to continually fit the available data, ensuring that the model remains up-to-date with the most recent information. However, the initial feature selection and hyper-parameters chosen during training remain constant throughout the forecasting period. This approach ensures that the Random Forest model is both dynamic and robust, adapting to new data while maintaining a consistent set of features and hyper-parameters.

LASSO and Ridge regression models are trained and fitted using the training and validation samples, with the penalty term optimized based on the validation sample performance. LASSO employs an L1 penalty, while Ridge uses an L2 penalty. After determining the best model specifications, these models are tested on out-of-sample data.

Model specification
Machine learning model specification
Optimal choice of hyperparamers

	Random forest	LASSO	Ridge
Hyperparamters			
Trees	500	-	-
Variables tried at each split	4	-	-
Lambda	-	0.018-0.35	0.219-11.5
Regularization	-	L1	L2
Model design			
Sequential updating	Yes	No	No
Feature selection	EFS	PRM	PRM
Features	18	10 - 50*	126

Features are the number of variables that are in the model and forecest inflation
Embedded feature selction (EFS)
Penalized regression method (PRM)
*Dependent on forecast horizon

Figure 4.6.2: Optimal hyper-parameters and model specifications for Random Forest, LASSO, and Ridge regression models. Random Forest uses 500 trees and 4 variables per split, with sequential updating and Embedded Feature Selection (EFS) incorporating 18 features. LASSO applies a lambda range of 0.018-0.35 with an L1 penalty, using Penalized Regression Method (PRM) with 10-50 features and no sequential updating. Ridge regression uses a lambda range of 0.219-11.5 with an L2 penalty, employing PRM with 126 features and no sequential updating.

4.6.3 Univariate time series models

The specification of the AR(p) model was based on results from the ACF, PACF and BIC. The SARIMA model was decided based on the same tests. Both approaches use maximum likelihood, and other approaches were not tested. Both time series models are sequentially updated as forecasts are made, adjusting only the coefficients of the model, not the hyper-parameters.

Model specification
Univariate model specification
Optimal choice of hyperparamers

	SARIMA	AR(p)
Hyperparamers		
p	0	3
d	1	-
q	1	-
P	0	-
D	0	-
Q	1	-
m	12	-
Selection criteria		
ACF	Yes	No
PACF	Yes	No
AIC	No	No
BIC	Yes	Yes
Parameter Estimation		
Method	MLE	MLE
Sequential updating	Yes	Yes

Maximum likelihood (MLE)

Figure 4.6.3: The table presents the optimal hyper-parameters and model specifications for SARIMA and AR(p) models. Selection criteria include ACF and PACF for SARIMA but not for AR(p). Both models use BIC for selection, with maximum likelihood estimation (MLE) and sequential updating.

4.7 Model evaluation methodology

Once results from out of sample for all forecast horizons are determined, the results are measured against the actual values out of sample. The RMSE test is applied giving a measure of how well each model performs at each forecasting horizon. The models are then compared to the benchmarks, which helps determine the significance of the results. The lower the RMSE results for a forecast the better. This approach gives a comprehensive understanding of how each model performs for each forecasting horizon in comparison to other models. RMSE is a nice evaluation metric, as it penalizes larger errors more severely than smaller errors. This is particularly relevant as the larger swings and trends within inflation are more important than small errors.

CHAPTER

FIVE

RESULTS

5.1 Our approach

In this paper we conduct forecasts using a broad range of models ranging from univariate, machine learning and recurrent neural network models. Benchmark models such as univariate SARIMA, and machine learning LASSO and random forest apply the FRED-MD data set covering different combinations of the 128 variables available in the dataset. The recurrent neural network models applied in the form of LSTM models, apply different combinations of datasets. LSTM was specified with FRED-MD, FRED-MD and EIKON data combined, as well as data augmentation approaches, increasing the number of observations.

All data was split in the same way, where the out of sample forecasts is for the time period of 2010 to 2023. The forecast horizon ranges from 1 to 48. LSTM struggles with initializing problems which makes the 1 month ahead forecast non-representative, and we will therefore not judge LSTM on this forecast horizon.

Out-of sample forecast performance for CPI inflation

	2010-2023			
	3	6	12	24
Benchmarks model				
AR(p)	0.58509	0.58196	0.67284	0.97440
SARIMA	0.51331	0.68603	0.85049	0.49029
Ridge	0.86746	1.43485	1.29971	2.26067
LASSO	0.61715	0.85281	1.57786	2.24279
Random forest	0.78364	1.05726	1.59568	1.41198
PCA-LSTMs	1. 64029	1.88628	1.65776	1.07852
PCA-LSTMm	1. 66826	1.81439	1.54394	1.01928
PCA-LSTMl	1. 65173	1.70043	1.57851	1.05971
LSTM all variables	1. 69833	2.24060	1.81466	1.34875
Models of interest				
FRED-MD data				
LASSO-LSTM small	1.56954	1.63074	1. 54964	1.14359
LASSO-LSTM medium	1.63345	1. 52513	1.71905	1. 43005
LASSO-LSTM large	2. 604	1.48998	1. 50488	1.32112
FRED-MD and EIKON data with engineered features				
LSTM	1.42853	1.78244	1.96479	1.73864
FRED-MD MBB data				
LASSO-LSTM small	2. 19806	2.19972	2.47337	2.19842
LASSO-LSTM medium	2. 64607	2.71433	2. 82160	1. 62318
LASSO-LSTM large	2.37562	2.65963	2.81938	2.04927

The table present RMSE with respect the CPI for horizons h = 3,6,12,24.
Small, medium and large refers to the number of predictors in the model.

Figure 5.1.1: The table displays the out-of-sample forecast performance for CPI inflation using the different models over the period from 2010 to 2023. The performance metric used is the Root Mean Squared Error (RMSE), evaluated over different forecast horizons.

5.2 Benchmark performance

5.2.1 Univariate models

Both the AR(p) model and the SARIMA model perform well for all forecasting horizons, reaffirming the existing findings in the literature. For very short and very long forecasting horizons, the SARIMA is superior, while for medium long forecasting horizons the AR(p) model is the superior performer. The performance of these naive univariate models is superior to most other models for almost all forecasting horizons.

5.2.2 Machine learning models

In our study there are three benchmark machine learning models that are used for comparison with the LSTM model. LASSO is superior compared to other machine learning models on a three month forecasting horizon. The 6 month forecast horizon is also quite good, and LASSO is competitive with other models. However as the forecasting horizons increases the performance of the LASSO model worsens, showing signs of high bias.

Ridge, with a lot of similarities to LASSO using the L2 penalty term and not the L1 penalty term, performed similar to LASSO. Performance starts well for the short forecast horizons, but as the horizon increases the performance of the model suffers, showing signs of high bias. This indicates that shrinking coefficients to zero is a better approach when forecasting, as Ridge will adapt to noise in the model. LASSO beats the Ridge model for all forecasting horizons except for the 12 month forecast.

The last machine learning benchmark model is random forest. Random forest performs worse than LASSO for the short forecast horizons, but displays a more consistent performance all over. While LASSO and Ridge perform poorly for the 24 month forecast horizon, random forest is able to produce competitive results.

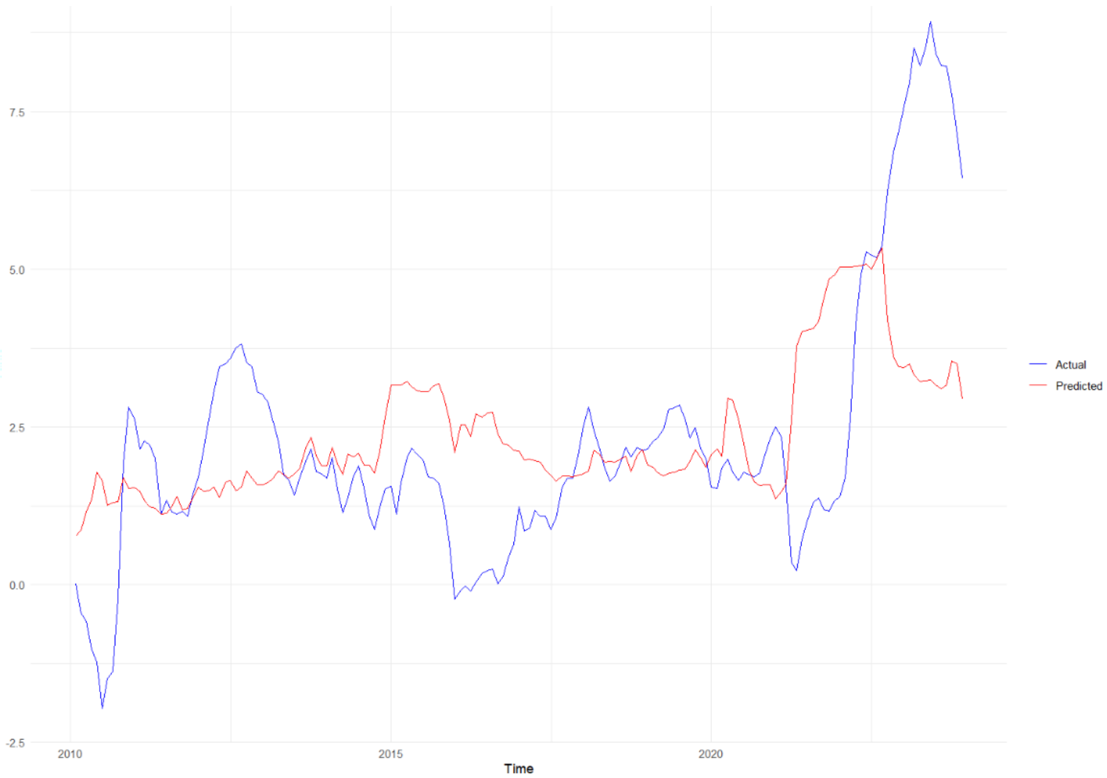

Figure 5.2.1: 12-month forecast using the Random Forest model, comparing actual (blue line) and predicted (red line) values. The model captures general trends but diverges significantly at certain points, particularly towards the end of the forecast period. This indicates some limitations in the model's predictive accuracy, especially during periods of high volatility.

5.2.3 PCA-LSTM performance

The LSTM benchmark models are applied with PCA, where feature selection comes in three forms; LSTM small, LSTM medium, LSTM large. For shorter forecast horizons the LSTM models performs poorly compared to the machine learning models and the univariate models. This deviates somewhat from findings in the literature, but it is not too surprising since there is little work on inflation forecasting using LSTM models. Our findings of LSTM on shorter forecasting horizons affirm the need for further studies to determine the ability of neural networks for forecasting inflation. On longer forecasting horizons the LSTM model outperforms machine learning models, and is partially on par with univariate models. Comparing the three architecture sizes of the LSTM model illustrates that for short to medium forecasting horizons, the differences are minimal. One should thoroughly evaluate if the effort needed to train larger models is worth the small gain. For longer forecasting horizons the medium sized architecture performs best, which is probably a result of the balance between overfitting and underfitting the model.

5.3 Model performance

5.3.1 LASSO-LSTM - FRED-MD data

The first set of models being evaluated for performance are the LASSO-LSTM models with three sizes of architecture. The LASSO-LSTM models differ from LSTM in the process of feature selection. Notably, we are interested in comparing how LASSO-LSTM is doing compared to LSTM applying the standard PCA feature selection approach. We want to assess if the new approach for feature selection improves model performance. Results show that for small architectures the LASSO-LSTM model is able to perform better than the LSTM small architecture with PCA. This is consistent for all horizons except the 24 months ahead, where LSTM is marginally better. For medium and large size architecture there is little difference between the LASSO LSTM and PCA-LSTM. As the number of predictors become larger, and closer to the total number of available predictors, the two models will converge in performance. As LSTM with LASSO and PCA converge in performance for larger architectures, this highlights the difference in performance when only a few features are allowed in the model.

Between the different LASSO-LSTM models the smallest architecture is the best performer. On the 3 and 24 month forecasts it beats the other two, but it is beaten, but still competetive, for 6 and 12 month forecasts. The medium architecture is quite consistent for all forecasting horizons, but performs worse than the small architecture. The large architecture has some outstanding results, but also some quite poor ones. This is probably due to overfitting. Notably, the large LASSO-LSTM architecture is able to outperform the LSTM model without any feature selection, meaning any feature selection is better than none. Despite some good forecasts from larger architectures, fewer predictors ensuring no overfitting seem better when forecasting inflation.

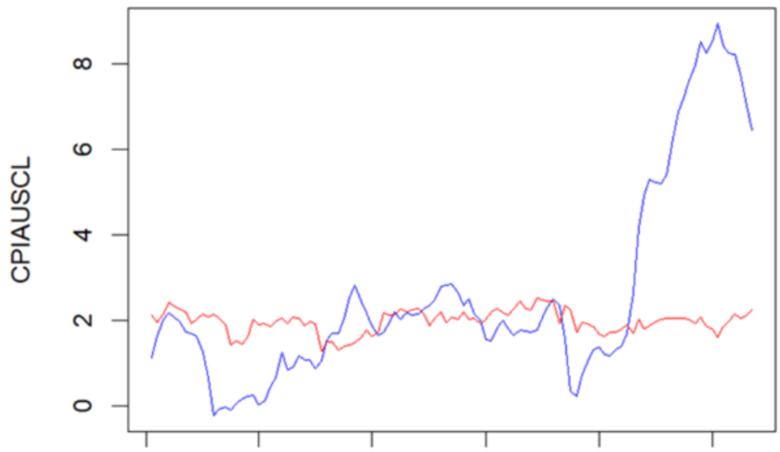

Figure 5.3.1: The figure illustrates a 6-month forecast using the LASSO-LSTM model, showing actual (blue line) and predicted (red line) CPIAUSCL values. The model closely follows the actual values in periods of low volatility but struggles to capture sharp increases, particularly towards the end of the forecast period. This indicates the model's limitations in predicting sudden changes in the data.

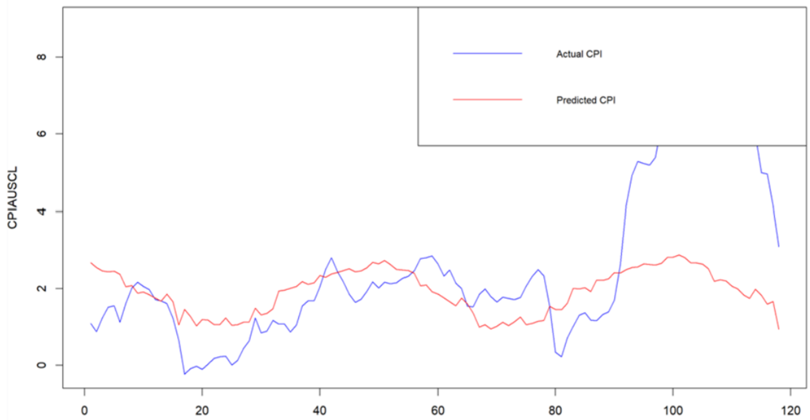

Figure 5.3.2: The observed under-performance during these volatile periods suggests that while the LASSO-LSTM model is robust in stable conditions, it may need further tuning or additional features to improve its predictive accuracy when the regime in the data changes. This highlights a common challenge in time series forecasting, where models often need to be continuously adapted to handle the complexities of real-world data.

5.3.2 FRED-MD and EIKON data with feature engineering

The next model is the LSTM model with financial data from the EIKON database and FRED-MD. This model consists mostly of prices, and the results are most impressive at the short forecasting horizon. As the forecasting horizon becomes longer the performance gets poorer. This model utilizes prices and not economic activity. As economic activity is important for longer forecasting horizons, it makes sense that this model performs best for shorter forecasting horizons. Out of all the LSTM models the financial data LSTM is the best performing model on the short forecasting horizons, reflecting the importance of prices at shorter forecasting horizons.

5.3.3 FRED-MD - MBB data

The final set of models has undergone a similar feature selection approach as the first set of LASSO-LSTM and consist of the same dataset. They only differ because of the moving block bootstrapping. The most important model for comparison in this case is the LASSO-LSTM with FRED-MD data. The results related to the data augmented LASSO-LSTM model is quite uninspiring, and the model is not able to deliver superior results at any forecasting horizon for any size of architecture. Data augmentation as an attempt to increase training data was for this reason unsuccessful. The training time of the model also increases considerably, and it is a lot more time consuming than all other LSTM models. Presumably, the augmented data does not retain the sequence well enough, and is ineffective in capturing different regimes and other patterns in macroeconomic data. As LSTM is sensitive to the structure of the data, better approaches to data augmentation is needed.

5.4 Benchmarks versus models of interest

While seeing an improvement in performance when switching from PCA to LASSO feature selection, LSTM is still not able to deliver competitive forecasts compared to the univariate time series models. For shorter forecasting horizons there are other machine learning models able to provide better forecasts than LSTM. Specifically, LASSO is able to deliver good forecasts for short term forecast horizons, beating all others. Random forest is also better at shorter forecast horizons. As forecast horizons increase the performance of LSTM becomes more competitive. The LSTM model is better than all of the machine learning models for the 24 month forecast, and competitive for 12 month forecasts. While LSTM produces good results for 24 month forecasts, all other machine learning models show signs of high bias. However, during the specification in the validation sample this issue was not possible to solve, as LASSO and random forest did not improve when features were removed or added. This indicates that the LSTM model could be a good option compared to other machine learning models, when dealing with longer forecast horizons.

5.5 Discussion of findings

5.5.1 Feature selection

We find that the LSTM model does improve in performance when paired with the LASSO feature selection approach. The model performance is best when few predictors are included. Performance is converging with PCA-LSTM when the number of features increases. This suggests the difference in performance is a result of the approach to feature selection. Although we anticipated better results from the financial data based LSTM model, it performs well for the shorter forecast horizon. This finding indicates that the choice of predictors should be dependent on the forecast horizon, as features that forecast well short-term differ from the ones performing well long-term.

Specifically, our findings indicate that prices and stock market indices should be heavily weighted for short forecast horizons, while predictors like industrial production or unemployment are more suited for inflation forecasting at longer forecast horizons. These factors should be considered for all models, but particularly for LSTM, which is highly sensitive to the choice of predictors. A model designed with only longer term predictors would capture the extent of this effect more comprehensively. We do however see this effect playing out in the feature selection process, where LASSO put more weight on prices for the shorter horizons compared to longer ones.

5.5.2 Data availability and model transparency

The data-augmented LASSO-LSTM approach failed to produce competitive results. We believe the primary reason being the retention of the sequence. LSTM is highly sensitive to sequences, as its recurrent neural network updates based on the previous step. If the bootstrapped data fails to retain the sequential structure, the LSTM may start fitting to random noise in the data. Proper data augmentation that preserves this sequential integrity could allow the LSTM to utilize the additional data more effectively and improve forecasting accuracy. Since the lack of sufficient data is considered the biggest obstacle for LSTM in delivering highly competitive inflation forecasts, well-executed data augmentation could be a viable solution.

While LSTM has proven competitive for the 24 month horizon compared to benchmark machine learning models, there is still a gap in performance compared to other models for most forecast horizons. This is concerning and we cannot recommend LSTM for the short forecast horizons. As mentioned previously in the thesis, LSTM is sensitive to initialization issues, and there is still no clear solution for this problem. The findings indicate that there is quite a lot of work to be done for LSTM to handle short term forecasts. We contribute this mediocre performance for shorter forecast horizons to the availability of data.

Another key point from our work is the challenge of determining the optimal hyper-parameters for the model. There is a wide range of hyper-parameters that influence the model's performance. While general guidelines exist for setting these hyper-parameters, extensive validation testing is required to find models with minimal loss. In reviewing similar work on LSTM for inflation forecasting, the trial-and-error approach often appears arbitrary and lacks a clear pattern explaining why certain hyper-parameter combinations perform better than others. This lack of transparency in neural networks is concerning, as it obscures the underlying reasoning behind the model behaviour. For forecast recipients such as central bankers, understanding the relationships and reasons behind forecasts is crucial. A model with obscure hyper-parameter choices is inherently problematic. While knowing the predictors used in the forecast provides some clarity, it may not be sufficient to overcome the interpretability issues. Therefore, improving transparency of such models, and understanding the rationale behind hyper-parameter settings is essential for enhancing the reliability and acceptance of these models.

Another issue with the LSTM model is the difficulty related to reproducing results. Due to its complex structure with multiple layers, dropout layers, batches, and numerous hyper-parameters, replicating the results achieved in other studies is challenging. Ensuring that the choice of hyper-parameters are from validation and hyper-parameters are based on in-sample, is difficult. Unfortunately, our requests for code from authors of papers on LSTM for inflation forecasting were denied. Additionally, the hyper-parameter descriptions in the most comprehensive papers are often incomplete. This lack of transparency is concerning for the broader adoption of LSTM in the field. Model transparency is nearly as crucial to forecast recipients as the forecasts themselves. Models like LASSO offer transparency, making it easier to understand the driving forces behind inflation. This understanding is vital for central bankers when deciding on policy rates. This discrepancy should be considered, especially when the interpretability of the model is essential for decision-making.

5.5.3 Broader implications

We highlight some major issues through our findings. Non-parametric models could potentially be confronting a similar crisis as seen in the 80s and post global financial crisis, when the univariate time series models started to outperform other models. Other crisis's, like when the Philips curve models stopped working after the GFC are also relevant. It is worth mentioning that we should not understate the importance of our findings to only cover inflation forecasts, as they also are relevant to more general problems with macroeconomic forecasting when applying non-parametric models.

While findings in the literature indicate that there is a use case for neural networks to forecast macroeconomics data, as such models can capture highly complex non-linearity in macroeconomic data, we point out a range of concerns in regards to the use of such models. While non-parametric models are capable of capturing non-linearity in macroeconomic data for certain windows, problems arise when there are shifts in regimes or states. It seems apparent that while LSTM is able to capture non-linearity well when confronted with a new regime, it is not able to handle the changes related to transitioning to a new economic state. If neural networks are not able to respond well to regime shifts, the use case of such models are severely compromised.

The use case of highly complex models like LSTM should be applied to reliably detect the large movements in inflation. In reality however, it is not unusual for the model to be fitted well to capture small changes, resulting in large misses during jumps in inflation. In cases where LSTM is fitted to capture the large changes in inflation, it comes at a large cost to the models ability to capture small changes in inflation. This is because it can produce seemingly random jumps in the forecasts of inflation during periods where only small changes to inflation exist. It is difficult to find a use case for inflation forecasting with LSTM when small changes are captured more accurately with univariate models, as LSTM cannot reliably forecast jumps in inflation. This is our main finding, and it is difficult to argue for the use of LSTM when small changes are consistently forecast better by parametric models, and sudden changes cannot be reliably forecast. Based on this finding we do not recommend the use of neural networks for forecasting within macroeconomics.

Lastly, it is worth pointing out the performance of univariate time series models. The performance shown is in a class of their own when compared to benchmarks and models of interest. As they are naive models they do not inherently tell anything about a underlying relationship between predictors and inflation. If there is no interest in cause, these models prove their utility by performing consistently for all forecasting horizons. When considering the performance of LSTM, and how it is not possible to understand the underlying relationship between predictors and inflation, univariate time series models seem more appropriate. As for cases where causality is of interest, other machine learning models like LASSO or random forest seem more appropriate than LSTM. As a result of the forecasting performance and the lack of transparency of LSTM, it is difficult to recommend and support the application of the model. It is however worth noting that transparency of neural networks are improving, as well as forecast performance. Yet given their current state it is difficult to make the case for LSTM applications within inflation forecasting.

In summary, the poor performance of neural networks during regime shifts makes them less suitable for macroeconomic forecasting. These models also suffer from a lack of interpretability, which is crucial in this field. Additionally, the current literature on LSTM in macroeconomics is difficult to reproduce, indicating a need for better documentation and transparency. Our work, which focuses on using LSTM for inflation forecasting, highlights several significant issues in the application of deep learning to macroeconomics.

5.6 Limitations of our research

5.6.1 Data

The greatest limitation of LSTM for macroeconomic forecasting is the availability of data. LSTM requires large sets of data to perform. With macroeconomic data only being reported on a monthly basis, and data quality deterring a lot before 1960, it is difficult for LSTM to capture the patterns within the data. For any specification of LSTM, or any size of the data available, it is unlikely that LSTM will be able to outperform other machine learning or univariate time series models on a consistent basis.

Further data limitations exist in the patterns within the data. The relationship between inflation and predictors are highly complex and non-linear. While LSTM can capture these non-linearities, there is the problem of the non-linearity being non-stationarity. Having non-stationary non-linearity makes the complexity within the data that much greater for the model to understand. LSTM is good at capturing non-linearity, but when these non-linearities change over time it becomes increasingly difficult..

5.6.2 Forecast horizon

As shown in the results, LSTM tends to perform better long-term compared to short-term. In comparison to models like LASSO or SARIMA, which perform well at shorter forecast horizons, this inconsistency makes LSTM difficult to recommend. LSTM also has a complex architecture consisting of many layers and a vast number of hyper-parameters, making it difficult to find a structure right for forecasting. This also makes it hard to reproduce results. This complexity also needs to be justified when comparing with other models with fewer layers and hyper-parameters, where implementation is easier and more accessible.

5.7 Future research

There is a vast range of possibilities to expand on the current work on LSTM. While LSTM currently does not perform on par with the univariate time series models, there are still possibilities to improve model performance. As shown previously, the LSTM model is sensitive to choice of hyper-parameters as well as feature selection, but just as importantly the number of observations the model is trained on. While the application of moving block bootstrapping to augment training data did not improve model performance, data augmentation has shown promising results in other research. Future work should focus on better implementing data augmentation, retaining regimes in the data as well as the random walk characteristics within the data. Other approaches of data augmentation as well as better MBB implementation could potentially yield significant results.

To enhance the performance of LSTM models, integrating LASSO is a promising approach among several potential improvements. Future research could explore ensemble models utilizing the strengths of LASSO or random forest, which already demonstrate strong results. Such ensemble models applied to compete with traditional time series models is the best direction for future work. Ensemble models may provide significant performance improvements. We generally believe that future work should focus on tree based models and penalty regression, rather than recurrent neural networks. Although current LSTM results are not highly competitive, advancements in RNN models could make them more viable for macroeconomic forecasting. Additionally, exploring other neural networks, such as convolutional neural networks and transformers, which show considerable promise, could yield valuable insights.

CHAPTER

SIX

SUMMARY AND CONCLUSION

We have evaluated a range of models forecasting inflation in the US, focusing particularly on LSTM models. Employing machine learning methods like LASSO for feature selection improves LSTM performance, but the LSTM model still falls short compared to other machine learning and time series models for most forecast horizons. While LSTM performs well for longer horizons, its performance is only competetive, and not superior, to other methods. Notably, univariate models such as SARIMA and AR(p) often match or outperform LSTM for most forecast horizons. This implies that simpler "backward-looking" models can be as effective as more complex approaches.

From the results arise the question as to why the performance of LSTM is so lackluster, when LSTM has the ability to deliver good results for other purposes. Data limitations might be a contributing factor as to why LSTM performed poorly. There are potentially too few inflationary events to capture any deep learning within the model. As neural networks require large datasets to be able to capture deep learning, the real world limitations, with only monthly reporting on inflation, makes it difficult to capture complex relationship in the model.

With 750 observations spanning significant events like changes in central bank priorities, the GFC, and the COVID-19 pandemic, the relationships within the data are complex and unique. This combination of unique events and limited observations may explain the mediocre performance of LSTM and other machine learning methods. Although LSTM can understand non-linear events, the non-constant nonlinearities in macroeconomics pose problems. Future work should explore data augmentation to enhance LSTM performance and mitigate overfitting.

In terms of interpretability, the LASSO-LSTM model shows clear advantages compared to LSTM with different approaches to feature selection. LASSO inherently provides a certain understanding of the importance of certain variables, as it delivers some coefficients and sets others to zero. The fact that including LASSO-LSTM resulted in better RMSE values, indicates that for macroeconomics LASSO is a good approach for feature selection. Understanding the underlying dynamics within the neural network is still complex and difficult to comprehend. However, smaller architectures with few features illustrates the importance of the factors in a regression, removing some fear of fitting to random noise.

While findings in the inflation forecasting literature indicate that LSTM and random forest are the most competitive compared to time series models, we are not able to replicate this performance. LASSO-LSTM is also not capable of significantly outperforming other machine learning models or univariate time series models. Reasons for this can be the time horizons used in previous studies. Our out of sample are based on the period going from post-GFC and after the spike in inflation post-covid. This period could potentially be subject to more volatility and nonlinear relationships than LASSO-LSTM is capable of capturing.

Other possible reasons as to why our findings deviate from other studies on LSTM, is how frequently we performed re-training of the network. Re-training the LSTM model can yield better performance, resetting the weights in the model. Such an approach should be used with caution, considering the importance of keeping out of sample data unseen and the computational intensity involved in retraining the network. The findings in this thesis should be a warning for policy makers and business owners in application of machine learning models for macroeconomic data. While neural networks and machine learning models have proven useful in a range of areas, they are no silver bullet, and should be used with caution. We recommend for this reason that any future use of machine learning and neural nets should be done in addition to univariate time series models like SARIMA.

BIBLIOGRAPHY

[1] Jens Iversen et al. "Real-time forecasting for monetary policy analysis: The case of Sveriges Riksbank". In: *Riksbank Research Paper Series* 142 (2016).

[2] International Monetary Fund. *World Economic Outlook April 2024*. Accessed on 05/30/2024. 2024.

[3] John H Boyd, Ross Levine, and Bruce D Smith. "The impact of inflation on financial sector performance". In: *Journal of monetary Economics* 47.2 (2001), pp. 221–248.

[4] Rodrigo Costamagna. "Inflation and R&D investment". In: *Journal of Innovation Economics & Management* 2 (2015), pp. 143–163.

[5] Douglas Staiger, James H Stock, and Mark W Watson. "The NAIRU, unemployment and monetary policy". In: *Journal of economic perspectives* 11.1 (1997), pp. 33–49.

[6] Andrew Atkeson, Lee E Ohanian, et al. "Are Phillips curves useful for forecasting inflation?" In: *Federal Reserve bank of Minneapolis quarterly review* 25.1 (2001), pp. 2–11.

[7] Mario Forni et al. "Do financial variables help forecasting inflation and real activity in the euro area?" In: *Journal of Monetary Economics* 50.6 (2003), pp. 1243–1255.

[8] Jan JJ Groen, Richard Paap, and Francesco Ravazzolo. "Real-time inflation forecasting in a changing world". In: *Journal of Business & Economic Statistics* 31.1 (2013), pp. 29–44.

[9] Yu-chin Chen, Stephen J Turnovsky, and Eric Zivot. "Forecasting inflation using commodity price aggregates". In: *Journal of Econometrics* 183.1 (2014), pp. 117–134.

[10] James H Stock and Mark W Watson. *Phillips Curve Inflation Forecasts*. Working Paper 14322. National Bureau of Economic Research, Sept. 2008.

[11] Livia Paranhos. "Predicting Inflation with Recurrent Neural Networks". In: *Journal of Economic Forecasting* 58.4 (2024), pp. 567–589.

[12] Michele Lenza, Inès Moutachaker, and Joan Paredes. "Forecasting euro area inflation with machine-learning models". In: *Research Bulletin* 112 (2023).

[13] Albert K Tsui, Cheng Yang Xu, and Zhaoyong Zhang. "Macroeconomic forecasting with mixed data sampling frequencies: Evidence from a small open economy". In: *Journal of Forecasting* 37.6 (2018), pp. 666–675.

[14] Wojciech Charemza and Daniel Ladley. "Central banks' forecasts and their bias: Evidence, effects and explanation". In: *International Journal of Forecasting* 32.3 (2016), pp. 804–817.

[15] Jonas DM Fisher, Chin T Liu, and Ruilin Zhou. "When can we forecast inflation?" In: *Economic Perspectives-Federal Reserve Bank of Chicago* 26.1 (2002), pp. 32–44.

[16] James H Stock and Mark W Watson. "Forecasting inflation". In: *Journal of monetary economics* 44.2 (1999), pp. 293–335.

[17] Flint Brayton et al. "The evolution of macro models at the Federal Reserve Board". In: *Carnegie-Rochester Conference Series on Public Policy*. Vol. 47. Elsevier. 1997, pp. 43–81.

[18] Aidan Meyler, Geoff Kenny, and Terry Quinn. "Forecasting Irish inflation using ARIMA models". In: (1998).

[19] Wayne Robinson. "Forecasting inflation using VAR analysis". In: *Econometric Modelling of Issues in Caribbean Economics and Finance. CCMS: St. Augustine, Trinidad and Tobago* (1998).

[20] Alexandre Fernandes Theoharidis, Diogo Abry Guillén, and Hedibert Lopes. "Deep learning models for inflation forecasting". In: *Applied Stochastic Models in Business and Industry* 39.3 (2023), pp. 447–470.

[21] Márcio GP Garcia, Marcelo C Medeiros, and Gabriel FR Vasconcelos. "Real-time inflation forecasting with high-dimensional models: The case of Brazil". In: *International Journal of Forecasting* 33.3 (2017), pp. 679–693.

[22] Sangmok Choi, Bruce D Smith, and John H Boyd. "Inflation, financial markets, and capital formation". In: *Review-Federal Reserve Bank of Saint Louis* 78 (1996), pp. 9–35.

[23] James H Stock and Mark W Watson. "Forecasting output and inflation: The role of asset prices". In: *Journal of economic literature* 41.3 (2003), pp. 788–829.

[24] Yu-chin Chen and Kenneth Rogoff. "Commodity currencies". In: *Journal of international Economics* 60.1 (2003), pp. 133–160.

[25] Robert A Amano and Simon Van Norden. "Exchange rates and oil prices". In: *Review of international economics* 6.4 (1998), pp. 683–694.

[26] Ben S Bernanke et al. "Inflation expectations and inflation forecasting". In: *Speech at the Monetary Economics Workshop of the National Bureau of Economic Research Summer Institute, Cambridge, Massachusetts*. Vol. 10. 2007, p. 11.

[27] Laura Freijeiro-González, Manuel Febrero-Bande, and Wenceslao González-Manteiga. "A critical review of LASSO and its derivatives for variable selection under dependence among covariates". In: *International Statistical Review* 90.1 (2022), pp. 118–145.

[28] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58.1 (1996), pp. 267–288.

[29] Jonas Ranstam and Jonathan A Cook. "LASSO regression". In: *Journal of British Surgery* 105.10 (2018), pp. 1348–1348.

[30] Anna Almosova and Niek Andresen. "Nonlinear inflation forecasting with recurrent neural networks". In: *Journal of Forecasting* 42.2 (2023), pp. 240–259.

[31] Sendhil Mullainathan and Jann Spiess. "Machine learning: an applied econometric approach". In: *Journal of Economic Perspectives* 31.2 (2017), pp. 87–106.

[32] Christopher M Bishop. “Pattern recognition and machine learning”. In: *Springer google schola* 2 (2006), pp. 1122–1128.

[33] Ashutosh Kumar Dubey et al. “Study and analysis of SARIMA and LSTM in forecasting time series data”. In: *Sustainable Energy Technologies and Assessments* 47 (2021), p. 101474.

CHAPTER

SEVEN

APPENDICES

7.1 Benchmark specification

7.1.1 Autoregressive Model *AR(p)*

The autoregressive model of order p (AR(p)) is defined as:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \phi_3 Y_{t-3} + \epsilon_t$$

where:

- Y_t: the value of the time series at time t,
- c: a constant term (intercept),
- ϕ_1, ϕ_2, ϕ_3: the parameters of the model which quantify the influence of the first, second, and third lag of the series on the current value, respectively,
- ϵ_t: the error term at time t, which is assumed to be white noise with a mean of zero and a constant variance.

7.1.2 Seasonal Autoregressive Integrated Moving Average *SARIMA*

The Seasonal Autoregressive Integrated Moving Average (SARIMA) model is defined as:

$$(1 - B)^1 Y_t = c + (1 + \theta_1 B)\epsilon_t + (1 + \Theta_1 B^{12})\epsilon_t$$

where:

- Y_t: the time series data,
- B: the backshift operator, where $B^k Y_t = Y_{t-k}$,

- ϵ_t: the error terms, which are assumed to be white noise,
- c: constant term (if included in the model),
- θ_1: parameter of the non-seasonal MA component,
- Θ_1: parameter of the seasonal MA component at lag 12.

There are a six steps to consider using SARIMA:

Step 1: Check if the series are stationary or not. If a time series has seasonality or varying trend mean over time, it varies at specific time frames, and must be transformed into a stationary time series.

Step 2: Applying differencing mechanisms . This is to make the time series stationary, if that´s not the case originally, as well as to check for seasonal differencing. It is done by taking the first difference and checking for stationarity until its stationarized.

Step 3: Create validation samples.

Step 4: Include AR and MA terms based on AC and PAC.

Step 5: Now the model is ready for prediction

Step 6: Validate the model by comparing the predicted values.

7.1.3 Least Absolute Shrinkage and Selection Operator *LASSO*

The data was split into training, validation, and test sets. Lambda (λ) was determined based on results within the validation sample. Introducing some noise to the Mean Squared Error (MSE) when setting λ yielded the best results. By introducing some noise to the specified LASSO model, forecasts improved. The model was trained only on the training and validation sets, without using a Bayesian approach. Lambda varies between 0.018 and 0.35 for different forecast horizons. Not sequentially updating, using PRM feature selection with L1 regularization.

7.1.3.1 LASSO Functions

The LASSO (Least Absolute Shrinkage and Selection Operator) method is a regression analysis technique that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. The mathematical representation of LASSO is as follows:

Objective Function:

$$\hat{\beta} = \arg\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

where:y_i is the response variable. x_i is the vector of predictor variables. β is the vector of coefficients. λ is the regularization parameter.

Regularization Term:

$$\lambda \sum_{j=1}^{p} |\beta_j|$$

This term adds a penalty equal to the absolute value of the magnitude of the coefficients, promoting sparsity in the model coefficients (i.e., some coefficients may become zero).

Feature Selection with L1 Regularization: The L1 regularization used in LASSO performs feature selection by shrinking some coefficients to zero, effectively excluding them from the model.

7.1.3.2 Training and Validation

1. **Data Splitting**: The dataset is divided into training, validation, and test sets.
2. **Hyperparameter Tuning**: Lambda (λ) is chosen based on the performance within the validation set. A range of λ values (0.018 to 0.35) is tested to determine the optimal value for different forecast horizons.
3. **Model Training**: The LASSO model is trained on the training and validation sets using the selected λ.
4. **Noise Introduction**: Introducing some noise to the Mean Squared Error (MSE) during λ selection helps to improve forecasts.

The LASSO model described here uses Predictive Recursive Modeling (PRM) for feature selection and does not update sequentially.

7.1.4 Ridge Regression

Ridge regression is a technique used to analyze multiple regression data that suffer from multicollinearity. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. The specification for the Ridge model is as follows: Lambda (λ) varies between 0.219 and 11.5, uses L2 regularization, and does not update sequentially. Predictive Recursive Modeling (PRM) is used for feature selection with a total of 126 features.

7.1.4.1 Ridge Regression Functions

Ridge regression (also known as Tikhonov regularization) is a method of estimating the coefficients of multiple-regression models in scenarios where independent variables are highly correlated. The mathematical representation of Ridge regression is as follows:

Objective Function:

$$\hat{\beta} = \arg\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

where:y_i is the response variable. x_i is the vector of predictor variables. β is the vector of coefficients. λ is the regularization parameter.

Regularization Term:

$$\lambda \sum_{j=1}^{p} \beta_j^2$$

This term adds a penalty equal to the square of the magnitude of the coefficients, which helps to shrink the coefficients and reduce their variance.

Feature Selection with L2 Regularization: The L2 regularization used in Ridge regression does not perform feature selection by shrinking coefficients to zero. Instead, it reduces the impact of less important features by shrinking their coefficients towards zero.

7.1.4.2 Training and Validation

1. **Data Splitting**: The dataset is divided into training, validation, and test sets.
2. **Hyperparameter Tuning**: Lambda (λ) is chosen based on the performance within the validation set. A range of λ values (0.219 to 11.5) is tested to determine the optimal value for different forecast horizons.
3. **Model Training**: The Ridge regression model is trained on the training and validation sets using the selected λ.
4. **Noise Introduction**: Introducing some noise to the Mean Squared Error (MSE) during λ selection helps to improve forecasts.

The Ridge regression model described here uses Predictive Recursive Modeling (PRM) for feature selection and does not update sequentially.

7.1.5 Random Forest Model

The Random Forest model is an ensemble learning method used for classification and regression tasks. The model consists of multiple decision trees and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. Below are the specifics of the random forest model used:

7.1.5.1 Model Specifications

- **Number of Trees**: 500
- **Variables Tried at Each Split**: 4
- **Feature Selection Method**: Exhaustive Feature Selection (EFS)
- **Total Number of Features**: 18

7.1.5.2 Random Forest Functions

1. **Bootstrap Sampling**: Randomly sample with replacement from the training data to create multiple subsets. Let $D = \{(x_i, y_i)\}_{i=1}^n$ be the training dataset with n samples. For each tree t in the forest, generate a bootstrap sample D_t by sampling n times with replacement from D.
2. **Tree Construction**: For each tree t in the forest:
   (a) **Select Features**: Randomly select 4 features from the 18 available features.

(b) **Node Splitting**: At each node, split based on the feature that provides the best split according to a specified criterion. For a node with data D, evaluate splits using a feature j and threshold θ:

$$\text{Split Criterion} = \arg\max_{j,\theta} \Delta I(D, j, \theta)$$

where ΔI is the improvement in the splitting criterion (e.g., Gini impurity for classification or Mean Squared Error for regression).

(c) **Recursive Splitting**: Recursively repeat the feature selection and node splitting process until a stopping criterion is met (e.g., maximum tree depth, minimum number of samples per leaf).

3. **Prediction Aggregation**: For each observation x to be predicted:

   (a) **Individual Tree Predictions**: Obtain the prediction from each of the 500 trees $h_t(x)$.

   (b) **Final Prediction**: Aggregate the predictions by taking the majority vote for classification tasks:

$$\hat{y} = \text{mode}\{h_t(x)\}_{t=1}^{500}$$

   or the average for regression tasks:

$$\hat{y} = \frac{1}{500} \sum_{t=1}^{500} h_t(x)$$

4. **Feature Importance**: Calculate the importance of each feature based on the improvement in the splitting criterion they bring about, averaged over all trees.

7.1.5.3 Exhaustive Feature Selection (EFS)

The Exhaustive Feature Selection (EFS) method involves evaluating all possible combinations of the features to determine the subset that results in the best model performance. The process is as follows:

1. **Initial Feature Set**: Begin with the full set of 18 features $F = \{f_1, f_2, \ldots, f_{18}\}$.
2. **Sequential Update**: Sequentially update the model by evaluating all possible subsets of features.
3. **Model Evaluation**: For each subset, train the random forest model and evaluate its performance using a specified metric.

7.1.5.4 Model Retraining and Forecasting

The model is retrained at each point in time, incorporating all available data up to that point, before making forecasts. The steps are as follows:

(a) **Data Accumulation**: At time t, use all data $D_t = \{(x_i, y_i)\}_{i=1}^{t}$ available up to time t to train the model.

(b) **Model Training**: Train the random forest model with the specified hyperparameters (500 trees, 4 variables tried at each split) using the accumulated data D_t.

(c) **Forecasting**: Use the trained model to make predictions for the next time step.

7.1.6 LSTM Cell Functions

The Long Short-Term Memory (LSTM) network is a type of recurrent neural network (RNN) that avoids the long-term dependency problem of traditional RNNs. An LSTM cell has three main gates that control the cell state and the hidden state (memory state):

- **Forget Gate**: Decides what information to throw away from the cell state.
- **Input Gate**: Decides which values from the input will update the cell state.
- **Output Gate**: Decides what the next hidden state should be.

7.1.6.1 Notations

$$\begin{aligned}
x_t &: \text{Input at time step } t \\
h_{t-1} &: \text{Hidden state at the previous time step } t-1 \\
C_{t-1} &: \text{Cell state at the previous time step } t-1 \\
h_t &: \text{Hidden state at the current time step } t \\
C_t &: \text{Cell state at the current time step } t \\
W_f, W_i, W_o, W_c &: \text{Weight matrices for forget, input, output gates, and cell state respectively} \\
b_f, b_i, b_o, b_c &: \text{Biases for forget, input, output gates, and cell state respectively} \\
\sigma &: \text{Sigmoid activation function} \\
\tanh &: \text{Hyperbolic tangent activation function} \\
\odot &: \text{Element-wise multiplication}
\end{aligned}$$

7.1.6.2 Forget Gate

The forget gate determines what proportion of the previous cell state to keep.

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

7.1.6.3 Input Gate

The input gate determines what proportion of the new information to update the cell state with.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

7.1.6.4 Candidate Cell State

The candidate cell state $\tilde{C}_t$ is created using a tanh layer.

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

7.1.6.5 Cell State Update

The new cell state C_t is a combination of the previous cell state and the candidate cell state, modulated by the forget gate and the input gate.

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

7.1.6.6 Output Gate

The output gate determines the next hidden state, based on the cell state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

7.1.6.7 Hidden State

The hidden state h_t is computed by applying the output gate to the tanh of the cell state.

$$h_t = o_t \odot \tanh(C_t)$$

7.1.6.8 Summary of LSTM Cell Functions

$$\begin{aligned}
&\text{Forget Gate :} && f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
&\text{Input Gate :} && i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
&\text{Candidate Cell State :} && \tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\
&\text{Cell State Update :} && C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
&\text{Output Gate :} && o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
&\text{Hidden State :} && h_t = o_t \odot \tanh(C_t)
\end{aligned}$$

7.2 Code Repository

In this thesis a range of models are applied to benchmark and test the model performance of LSTM. As discussed in the thesis, application of non-parametric models are problematic as there is a range of hyper-parameters. In the thesis it is addressed the difficulty of replication. The repositories below cover models applied. The repositories should be enough such that replication should be possible. All models run in R, with LSTM running python in R. Data applied, and handling of data is also presented in the repositories.

The code developed for this thesis can be accessed at the following GitHub repository:

[https://github.com/TormodRy/Masters-thesis-models]

[https://github.com/TormodRy/Masters-thesis-data]

The repositories should contain all scripts, data files, and documentation required to replicate the results presented in this thesis.

7.3 FRED-MD data

The FRED-MD (Federal Reserve Economic Data - Monthly Database) is a comprehensive macroeconomic database maintained by the Federal Reserve Bank of St. Louis. It was developed by Michael W. McCracken and Serena Ng and provides a wide range of monthly economic and financial data for the United States.

The dataset includes 128 monthly time series variables, covering aspects such as output and income, labor market, housing, consumption, orders and inventories, money and credit, interest rates, prices, and stock markets.

The data spans several decades, with many series starting from the 1950s or 1960s and extending to the present. For more detailed information you will find a table below or refer to the original paper by McCracken and Ng:

Michael W. McCracken and Serena Ng, "FRED-MD: A Monthly Database for Macroeconomic Research," Journal of Business and Economic Statistics, 2016.

Appendix

The column TCODE denotes the following data transformation for a series x: (1) no transformation; (2) Δx_t; (3) $\Delta^2 x_t$; (4) $log(x_t)$; (5) $\Delta \log(x_t)$; (6) $\Delta^2 \log(x_t)$; (7) $\Delta(x_t/x_{t-1} - 1.0)$. The FRED column gives mnemonics in FRED followed by a short description. The comparable series in Global Insight is given in the column GSI.

Some series require adjustments to the raw data available in FRED. We tag these variables with an asterisk to indicate that they been adjusted and thus differ from the series from the source. A summary of the adjustments is detailed in the paper https://research.stlouisfed.org/wp/2015/2015- 012.pdf

Group 1: Output and income

	id	tcode	fred	description	gsi	gsi:description
1	1	5	RPI	Real Personal Income	M_14386177	PI
2	2	5	W875RX1	Real personal income ex transfer receipts	M_145256755	PI less transfers
3	6	5	INDPRO	IP Index	M_116460980	IP: total
4	7	5	IPFPNSS	IP: Final Products and Nonindustrial Supplies	M_116460981	IP: products
5	8	5	IPFINAL	IP: Final Products (Market Group)	M_116461268	IP: final prod
6	9	5	IPCONGD	IP: Consumer Goods	M_116460982	IP: cons gds
7	10	5	IPDCONGD	IP: Durable Consumer Goods	M_116460983	IP: cons dble
8	11	5	IPNCONGD	IP: Nondurable Consumer Goods	M_116460988	IP: cons nondble
9	12	5	IPBUSEQ	IP: Business Equipment	M_116460995	IP: bus eqpt
10	13	5	IPMAT	IP: Materials	M_116461002	IP: matls
11	14	5	IPDMAT	IP: Durable Materials	M_116461004	IP: dble matls
12	15	5	IPNMAT	IP: Nondurable Materials	M_116461008	IP: nondble matls
13	16	5	IPMANSICS	IP: Manufacturing (SIC)	M_116461013	IP: mfg
14	17	5	IPB51222s	IP: Residential Utilities	M_116461276	IP: res util
15	18	5	IPFUELS	IP: Fuels	M_116461275	IP: fuels
16	20	2	CUMFNS	Capacity Utilization: Manufacturing	M_116461602	Cap util

Group 2: Labor market

	id	tcode	fred	description	gsi	gsi:description
1	21*	2	HWI	Help-Wanted Index for United States		Help wanted indx
2	22*	2	HWIURATIO	Ratio of Help Wanted/No. Unemployed	M_110156531	Help wanted/unemp
3	23	5	CLF16OV	Civilian Labor Force	M_110156467	Emp CPS total
4	24	5	CE16OV	Civilian Employment	M_110156498	Emp CPS nonag
5	25	2	UNRATE	Civilian Unemployment Rate	M_110156541	U: all
6	26	2	UEMPMEAN	Average Duration of Unemployment (Weeks)	M_110156528	U: mean duration
7	27	5	UEMPLT5	Civilians Unemployed - Less Than 5 Weeks	M_110156527	U < 5 wks
8	28	5	UEMP5TO14	Civilians Unemployed for 5-14 Weeks	M_110156523	U 5-14 wks
9	29	5	UEMP15OV	Civilians Unemployed - 15 Weeks & Over	M_110156524	U 15+ wks
10	30	5	UEMP15T26	Civilians Unemployed for 15-26 Weeks	M_110156525	U 15-26 wks
11	31	5	UEMP27OV	Civilians Unemployed for 27 Weeks and Over	M_110156526	U 27+ wks
12	32*	5	CLAIMSx	Initial Claims	M_15186204	UI claims
13	33	5	PAYEMS	All Employees: Total nonfarm	M_123109146	Emp: total
14	34	5	USGOOD	All Employees: Goods-Producing Industries	M_123109172	Emp: gds prod
15	35	5	CES1021000001	All Employees: Mining and Logging: Mining	M_123109244	Emp: mining
16	36	5	USCONS	All Employees: Construction	M_123109331	Emp: const
17	37	5	MANEMP	All Employees: Manufacturing	M_123109542	Emp: mfg
18	38	5	DMANEMP	All Employees: Durable goods	M_123109573	Emp: dble gds
19	39	5	NDMANEMP	All Employees: Nondurable goods	M_123110741	Emp: nondbles
20	40	5	SRVPRD	All Employees: Service-Providing Industries	M_123109193	Emp: services
21	41	5	USTPU	All Employees: Trade, Transportation & Utilities	M_123111543	Emp: TTU
22	42	5	USWTRADE	All Employees: Wholesale Trade	M_123111563	Emp: wholesale
23	43	5	USTRADE	All Employees: Retail Trade	M_123111867	Emp: retail
24	44	5	USFIRE	All Employees: Financial Activities	M_123112777	Emp: FIRE
25	45	5	USGOVT	All Employees: Government	M_123114411	Emp: Govt
26	46	1	CES0600000007	Avg Weekly Hours : Goods-Producing	M_140687274	Avg hrs
27	47	2	AWOTMAN	Avg Weekly Overtime Hours : Manufacturing	M_123109554	Overtime: mfg
28	48	1	AWHMAN	Avg Weekly Hours : Manufacturing	M_14386098	Avg hrs: mfg
29	127	6	CES0600000008	Avg Hourly Earnings : Goods-Producing	M_123109182	AHE: goods
30	128	6	CES2000000008	Avg Hourly Earnings : Construction	M_123109341	AHE: const
31	129	6	CES3000000008	Avg Hourly Earnings : Manufacturing	M_123109552	AHE: mfg

Group 3: Housing

	id	tcode	fred	description	gsi	gsi:description
1	50	4	HOUST	Housing Starts: Total New Privately Owned	M_110155536	Starts: nonfarm
2	51	4	HOUSTNE	Housing Starts, Northeast	M_110155538	Starts: NE
3	52	4	HOUSTMW	Housing Starts, Midwest	M_110155537	Starts: MW
4	53	4	HOUSTS	Housing Starts, South	M_110155543	Starts: South
5	54	4	HOUSTW	Housing Starts, West	M_110155544	Starts: West
6	55	4	PERMIT	New Private Housing Permits (SAAR)	M_110155532	BP: total
7	56	4	PERMITNE	New Private Housing Permits, Northeast (SAAR)	M_110155531	BP: NE
8	57	4	PERMITMW	New Private Housing Permits, Midwest (SAAR)	M_110155530	BP: MW
9	58	4	PERMITS	New Private Housing Permits, South (SAAR)	M_110155533	BP: South
10	59	4	PERMITW	New Private Housing Permits, West (SAAR)	M_110155534	BP: West

Group 4: Consumption, orders, and inventories

	id	tcode	fred	description	gsi	gsi:description
1	3	5	DPCERA3M086SBEA	Real personal consumption expenditures	M_123008274	Real Consumption
2	4*	5	CMRMTSPLx	Real Manu. and Trade Industries Sales	M_110156998	M&T sales
3	5*	5	RETAILx	Retail and Food Services Sales	M_130439509	Retail sales
4	64	5	ACOGNO	New Orders for Consumer Goods	M_14385863	Orders: cons gds
5	65*	5	AMDMNOx	New Orders for Durable Goods	M_14386110	Orders: dble gds
6	66*	5	ANDENOx	New Orders for Nondefense Capital Goods	M_178554409	Orders: cap gds
7	67*	5	AMDMUOx	Unfilled Orders for Durable Goods	M_14385946	Unf orders: dble
8	68*	5	BUSINVx	Total Business Inventories	M_15192014	M&T invent
9	69*	2	ISRATIOx	Total Business: Inventories to Sales Ratio	M_15191529	M&T invent/sales
10	130*	2	UMCSENTx	Consumer Sentiment Index	hhsntn	Consumer expect

Group 5: Money and credit

	id	tcode	fred	description	gsi	gsi:description
1	70	6	M1SL	M1 Money Stock	M_110154984	M1
2	71	6	M2SL	M2 Money Stock	M_110154985	M2
3	72	5	M2REAL	Real M2 Money Stock	M_110154985	M2 (real)
4	73	6	BOGMBASE	Monetary Base	M_110154995	MB
5	74	6	TOTRESNS	Total Reserves of Depository Institutions	M_110155011	Reserves tot
6	75	7	NONBORRES	Reserves Of Depository Institutions	M_110155009	Reserves nonbor
7	76	6	BUSLOANS	Commercial and Industrial Loans	BUSLOANS	C&I loan plus
8	77	6	REALLN	Real Estate Loans at All Commercial Banks	BUSLOANS	DC&I loans
9	78	6	NONREVSL	Total Nonrevolving Credit	M_110154564	Cons credit
10	79*	2	CONSPI	Nonrevolving consumer credit to Personal Income	M_110154569	Inst cred/PI
11	132	6	DTCOLNVHFNM	Consumer Motor Vehicle Loans Outstanding	N.A.	N.A.
12	133	6	DTCTHFNM	Total Consumer Loans and Leases Outstanding	N.A.	N.A.
13	134	6	INVEST	Securities in Bank Credit at All Commercial Banks	N.A.	N.A.

Group 6: Interest and exchange rates

	id	tcode	fred	description	gsi	gsi:description
1	84	2	FEDFUNDS	Effective Federal Funds Rate	M_110155157	Fed Funds
2	85*	2	CP3Mx	3-Month AA Financial Commercial Paper Rate	CPF3M	Comm paper
3	86	2	TB3MS	3-Month Treasury Bill:	M_110155165	3 mo T-bill
4	87	2	TB6MS	6-Month Treasury Bill:	M_110155166	6 mo T-bill
5	88	2	GS1	1-Year Treasury Rate	M_110155168	1 yr T-bond
6	89	2	GS5	5-Year Treasury Rate	M_110155174	5 yr T-bond
7	90	2	GS10	10-Year Treasury Rate	M_110155169	10 yr T-bond
8	91	2	AAA	Moody's Seasoned Aaa Corporate Bond Yield		Aaa bond
9	92	2	BAA	Moody's Seasoned Baa Corporate Bond Yield		Baa bond
10	93*	1	COMPAPFFx	3-Month Commercial Paper Minus FEDFUNDS		CP-FF spread
11	94	1	TB3SMFFM	3-Month Treasury C Minus FEDFUNDS		3 mo-FF spread
12	95	1	TB6SMFFM	6-Month Treasury C Minus FEDFUNDS		6 mo-FF spread
13	96	1	T1YFFM	1-Year Treasury C Minus FEDFUNDS		1 yr-FF spread
14	97	1	T5YFFM	5-Year Treasury C Minus FEDFUNDS		5 yr-FF spread
15	98	1	T10YFFM	10-Year Treasury C Minus FEDFUNDS		10 yr-FF spread
16	99	1	AAAFFM	Moody's Aaa Corporate Bond Minus FEDFUNDS		Aaa-FF spread
17	100	1	BAAFFM	Moody's Baa Corporate Bond Minus FEDFUNDS		Baa-FF spread
18	101	5	TWEXAFEGSMTHx	Trade Weighted U.S. Dollar Index		Ex rate: avg
19	102*	5	EXSZUSx	Switzerland / U.S. Foreign Exchange Rate	M_110154768	Ex rate: Switz
20	103*	5	EXJPUSx	Japan / U.S. Foreign Exchange Rate	M_110154755	Ex rate: Japan
21	104*	5	EXUSUKx	U.S. / U.K. Foreign Exchange Rate	M_110154772	Ex rate: UK
22	105*	5	EXCAUSx	Canada / U.S. Foreign Exchange Rate	M_110154744	EX rate: Canada

Group 7: Prices

	id	tcode	fred	description	gsi	gsi:description
1	106	6	WPSFD49207	PPI: Finished Goods	M110157517	PPI: fin gds
2	107	6	WPSFD49502	PPI: Finished Consumer Goods	M110157508	PPI: cons gds
3	108	6	WPSID61	PPI: Intermediate Materials	M_110157527	PPI: int matls
4	109	6	WPSID62	PPI: Crude Materials	M_110157500	PPI: crude matls
5	110*	6	OILPRICEx	Crude Oil, spliced WTI and Cushing	M_110157273	Spot market price
6	111	6	PPICMM	PPI: Metals and metal products:	M_110157335	PPI: nonferrous
7	113	6	CPIAUCSL	CPI : All Items	M_110157323	CPI-U: all
8	114	6	CPIAPPSL	CPI : Apparel	M_110157299	CPI-U: apparel
9	115	6	CPITRNSL	CPI : Transportation	M_110157302	CPI-U: transp
10	116	6	CPIMEDSL	CPI : Medical Care	M_110157304	CPI-U: medical
11	117	6	CUSR0000SAC	CPI : Commodities	M_110157314	CPI-U: comm.
12	118	6	CUSR0000SAD	CPI : Durables	M_110157315	CPI-U: dbles
13	119	6	CUSR0000SAS	CPI : Services	M_110157325	CPI-U: services
14	120	6	CPIULFSL	CPI : All Items Less Food	M_110157328	CPI-U: ex food
15	121	6	CUSR0000SA0L2	CPI : All items less shelter	M_110157329	CPI-U: ex shelter
16	122	6	CUSR0000SA0L5	CPI : All items less medical care	M_110157330	CPI-U: ex med
17	123	6	PCEPI	Personal Cons. Expend.: Chain Index	gmdc	PCE defl
18	124	6	DDURRG3M086SBEA	Personal Cons. Exp: Durable goods	gmdcd	PCE defl: dlbes
19	125	6	DNDGRG3M086SBEA	Personal Cons. Exp: Nondurable goods	gmdcn	PCE defl: nondble
20	126	6	DSERRG3M086SBEA	Personal Cons. Exp: Services	gmdcs	PCE defl: service

Group 8: Stock market

	id	tcode	fred	description	gsi	gsi:description
1	80*	5	S&P 500	S&P's Common Stock Price Index: Composite	M_110155044	S&P 500
2	82*	2	S&P div yield	S&P's Composite Common Stock: Dividend Yield		S&P div yield
3	83*	5	S&P PE ratio	S&P's Composite Common Stock: Price-Earnings Ratio		S&P PE ratio
4	135*	1	VIXCLSx	VIX		

7.4 Financial data

The following table provides a comprehensive overview of various economic and financial datasets, capturing a wide range of financial activities through various indexes of commodities and freight rates. Each entry details the dataset's name, frequency of data collection, the time period covered, and a brief description of the variables included. Additionally, the data has been differenced to make it stationary, often using a year-over-year (YoY) differencing method, such as $x_t - x_{t-12}$. This transformation helps to remove trends and seasonality, making the data more suitable for time series analysis. The datasets span from January 1960 to December 2023, offering a broad historical perspective on these key economic measures.

Dataset	Frequency	Time Period	Description
Baker Hughes	Monthly	January 1960 to December 2023	Date (DD.MM.YYYY format), Baker Hughes rig count, Coffee prices (Arabica in USD, Sao Paulo), HARPEX index, Oil prices, Silver prices, Agriculture index, Economic impacts (Paul Volcker, Oil Crisis, GFC, Covid)
ESALQ Brazil Coffee Arabica Price	Monthly	January 1960 to December 2023	Date (DD.MM.YYYY format), Arabica coffee prices in USD (Sao Paulo)
Financial Data	Monthly	January 1960 to December 2023	Various financial indicators and metrics, potentially including stock prices and exchange rates
Gold Spot Rate	Monthly	January 1960 to December 2023	Date (DD.MM.YYYY format), Gold spot rate (USD per ounce)
HARPEX Charter Rate	Monthly	January 1960 to December 2023	Date (DD.MM.YYYY format), HARPEX index (container ship charter rates)
Oil (BFO-N)	Monthly	January 1960 to December 2023	Date (DD.MM.YYYY format), Oil price (Brent Crude Oil)
Silver Spot Rate	Monthly	January 1960 to December 2023	Date (DD.MM.YYYY format), Silver spot rate (USD per ounce)
USDA Daily Fresh 50% Lean Trimmings	Monthly	January 1960 to December 2023	Date (DD.MM.YYYY format), Price of 50% lean trimmings (USD), as reported daily by USDA

Figure 7.4.1: Financial dataset

7.5 Moving Block Bootstrap

The Moving Block Bootstrap (MBB) is a resampling technique used to create bootstrap samples for time series data while preserving the dependence structure within the data. Unlike traditional bootstrap methods, which resample individual observations, MBB resamples blocks of consecutive observations. This method is particularly useful for time series data where the observations are not independent.

7.5.1 Procedure for Moving Block Bootstrap

The MBB procedure involves the following steps:

1. **Choose Block Length**: Select an appropriate block length l. The choice of l is crucial as it affects the balance between bias and variance in the bootstrap samples.

2. **Create Blocks**: Divide the original time series $\{X_t\}_{t=1}^{n}$ into overlapping blocks of length l. Specifically, create $n - l + 1$ blocks:

$$B_i = \{X_i, X_{i+1}, \ldots, X_{i+l-1}\} \quad \text{for} \quad i = 1, 2, \ldots, n - l + 1$$

3. **Resample Blocks**: Randomly sample $k = \lceil n/l \rceil$ blocks with replacement from the set of overlapping blocks $\{B_i\}$. Denote the selected blocks as $\{B_{i_1}, B_{i_2}, \ldots, B_{i_k}\}$.

4. **Construct Bootstrap Samples**: Repeat the above steps to generate multiple bootstrap samples. Each bootstrap sample preserves the time series dependence structure within each block.

5. **Original data as the end of the time series:** The last part of the training data should be the original dataset. This is to retain a sequence when moving from training data to out of sample.

NTNU
Norwegian University of
Science and Technology