Winter 2023

Recurrent Neural Networks for Flash GDP Estimates in Ireland: A Comparison with Traditional Econometric Methods

Justin Flannery

CCT College Dublin

Recurrent Neural Networks for Flash GDP Estimates in Ireland: A Comparison with Traditional Econometric Methods

Justin Flannery

A Thesis Submitted in Partial Fulfilment

of the requirements for the

Degree of

Master of Science in Data Analytics

cct | College Dublin
Computing • IT • Business

September 2023

Supervisor: Marina Iantorno

Abstract

GDP is the single most important barometer for the health of an economy. It's an important input into the decision making processes of government, industry and state institutions such as central banks. To be useful as an indicator, GDP estimates need to be both timely and accurate. To meet the needs of users, many national statistical institutes publish early or flash estimates of GDP which are produced within 30 days after the end of a quarter. Given the long lags involved in the data collection processes which feed into GDP estimates, these flash estimates are often largely model based. Within the EU, the models utilised are typically the workhorse models of statistics and time series econometrics such as regression and ARIMA models. This study seeks to assess whether deep learning approaches can be used to improve the accuracy of early estimates in a flash GDP context. To assess this a number of number of LSTM models were trained with extensive hyperparameter tuning with their accuracy evaluated based on common metrics along with walk forward validation on a test set. These results were compared to a similar approach with time series econometric models such as ARIMA, ARIMA with additional explanatory variables and VAR. The study concludes that ARIMA models with explanatory variables provide the most accurate estimates. The study also provides recommendations for the improvement of Ireland's flash GDP estimates process. The study recommends the use of additional explanatory variables in the context of ARIMA modelling. This recommendation was based on findings from this study and insights into Ireland's flash estimate processes gained from in-depth interviews with experts.

Contents

Table 1: List of Acronyms

Acronym	**Full Form**
ACF	Auto-Correlation Function
Adam	Adaptive Moment Estimation
ADF	Augmented Dickey Fuller
AIC	Akaike Information Criterion
ANN	Artificial Neural Network
ARDL	Auto-Regressive Distributed Lag
ARIMA	Autoregressive Integrated Moving Average
CPI	Consumer Price Index
CSO	Central Statistics Office
ECM	Error Correction Model
EU	European Union
GDP	Gross Domestic Product
GRU	Gated Recurrent Unit
HS	Harmonized Commodity Description and Coding System
IMF	International Monetary Fund
LASSO	Least Absolute Shrinkage and Selection Operator
LSTM	Long Short Term Memory
MAPE	Mean Absolute Percentage Error
MBGD	Mini-Batch Gradient Descent
MIP	Monthly Industrial Production
MLE	Maximum Likelihood Estimation
MNE	Multi-National Enterprise
MSE	Mean Squared Error
MSI	Monthly Services Index
NACE	Nomenclature statistique des activités économiques dans la Communauté européenne
OECD	Organisation for Economic Cooperation and Development
OLS	Ordinary Least Squares
PACF	Partial Auto-Correlation Function
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RSI	Retail Sales Index
SGD	Stochastic Gradient Descent
SNA	System of National Accounts
STS	Structural Time Series
UN	United Nations
VAR	Vector AutoRegression
VAT	Value Added Tax

1 Introduction

Gross Domestic Product (GDP) is a comprehensive measure of the economic activity within a geographic region. It is defined by the OECD [OECD, 2002] as "an aggregate measure of production equal to the sum of the gross values added of all resident and institutional units engaged in production and services (plus any taxes, and minus any subsidies, on products not included in the value of their outputs)".

The origins of measuring the size of an economy can be traced back to the 17th century where the problem was a practical one of how much tax revenue the state could generate for both its peace and war time needs. However, the modern concept took shape in the 1930s and 1940s. The Great Depression from 1929 created an impetus for better statistics on economic developments while also spurring theoretical developments in economics which had direct implications for national income accounting. One of those theoretical developments was Keynes' publication of the General Theory of Employment, Interest and Money [Keynes, 1936]. The analysis introduced by Keynes in the General Theory created a direct link with national income accounting as both use the same macroeconomic identities, see for example equation 1 where Y represents GDP, C represents Consumption, I represents Investment, G represents Government Spending and NX represents Net Exports. In addition, Keynes' analysis emphasised a role for Government in stabilising economic fluctuations. This necessitated producing economic measurements from a sectoral perspective. The modern concept of GDP was first introduced by Simon Kuznets for a 1934 US congress report [Kuznets, 1934]. However, the first set of international guidelines weren't published until 1947, when the UN published its guidelines.

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

National income accounting has been through many iterations and it continues to be developed with the latest set of international guidelines, known as the System of National Accounts (SNA), published in 2008. The 2008 SNA was produced under the auspices of the United Nations, the European Commission, the Organisation for Economic Co-operation and Development, the International Monetary Fund and the World Bank Group.

GDP is a key metric which enables decision makers within both industry and Government to know how the economy is performing. For example, industry is interested in a barometer of how the economy is performing to inform their decisions around investments in plant, machinery and equipment. In addition, Central Banks are keenly interested in economic developments as it informs their decisions around monetary policy. For example, the Taylor rule [Taylor, 1993] provides an equation for how Central Banks should conduct monetary policy. Under the Taylor rule, the gap between GDP and its potential[1] level enters directly into the rule, see equation 2 where i represents the nominal interest rate, r represents the natural real rate, π represents inflation, π^* represents target inflation, y represents output and $\hat{y}$ represents the potential level of output. However, to be useful as an indicator of economic activity, GDP needs to be both accurate and timely. If the data is not accurate and timely, economic actors may make decisions too late or make them based on incorrect information. Member states of the European Union are legally required to transmit

[1] Potential relates to what production could be if the factors of production, ie capital and labour were fully employed.

quarterly GDP estimates within 60 days of the end of a quarter. To address user needs for more timely data, member states are encouraged but not required to provide early estimates within 30 days of the end of a quarter. Such early estimates are known as flash estimates. As the data for producing GDP estimates is typically only available with considerable lags, flash estimates are largely model based.

$$i_t = r^* + \pi_t + a(\pi_t - \pi^*) + b(y_t - \hat{y}) \tag{2}$$

The academic background of the national accountants who compile GDP estimates would often be Economics. Economics students would typically study various time series econometrics methods such as Auto-Regressive Integrated Moving Average (ARIMA) models and Vector Auto-Regression (VAR) models. However, modern deep learning methods would be less commonly taught. This study seeks to assess whether deep learning methods can be incorporated into flash GDP estimation to improve the accuracy of the estimates. Specifically, a type of Recurrent Neural Network (RNN) called a Long Short Term Memory (LSTM) model is explored. Such a model is considered to be state-of-the-art in the context of time series forecasting.

The design of the study as well as problem identification and clarification will be discussed in chapter 2. The discussion will explain why GDP will not be the focus of this study, instead modified final domestic demand will be. Chapter 3 reviews the literature on flash GDP estimation, as well as the literature on the methods typically utilised within flash GDP estimation. It also considers the literature comparing traditional econometric methods such as ARIMA and the more recent developments in deep learning. Chapter 4 provides a primer on all the key concepts and models utilised for this study. Chapter 5 will discuss implementation details and results. Chapter 6 will discuss the results in the context of the literature review and will asses the implications of the findings. Conclusions are set out in chapter 7.

2 Research Design

2.1 Primary Data

The primary data for this research was in-depth interviews conducted with experts in the area of flash GDP estimates. In-depth interviews complemented the insights gained from the secondary data. They provided greater insight into existing processes for producing flash GDP estimates within the Central Statistics Office. This included information on the types of models utilised and challenges faced. To identify individuals to interview, judgement sampling was implemented.

2.2 Secondary Data

The secondary data for this study relies on publicly available economic time series data. There are a number of high frequency data sources which could be useful as predictors in the context of flash GDP estimates. High frequency in an economics context typically means monthly as opposed to GDP estimates which are only available on a quarterly basis. Variables included for exploration in this study were:

- **Retail Sales:** An indicator of changes in the level of consumer spending on retail goods in Ireland.
- **Monthly Services Index:** An indicator of output in the non-financial traded services sector.
- **Monthly Industrial Production:** An indicator of the change in the volume of production.
- **Imports:** Monthly trade data on the movement of goods between countries. Aggregate data was mostly utilised, however for some models detailed trade data was used instead. As a way to capture information on expenditure on building and construction in domestic demand, import data classed as HS68[2] was used. HS84[3], HS85[4], HS86[5], HS87[6], HS89[7] and HS93[8] were also used to capture aspects of gross capital formation in domestic demand, specifically machinery and equipment. Note that these HS codes are part of the Harmonized Commodity Description and Coding System developed by the World Customs Organization (WCO).
- **Exports:** Monthly trade data on the movement of goods between countries.
- **Dwellings:** Dwellings completions in a quarter based on connections to the ESB network.
- **Departures:** The number of overseas departures on air and sea transport per month.
- **Consumer Price Index:** The official measure of consumer price changes.
- **Unemployment** The official measure of the unemployment rate.

[2] 'Articles of Stone, Plaster, Cement, Asbestos, Mica Or Similar Materials'

[3] 'Nuclear reactors, boilers, machinery and mechanical appliances; parts.'

[4] 'Electrical Machinery And Equipment And Parts Thereof; Sound Recorders. . . '

[5] 'Railway Or Tramway Locomotives, Rolling Stock And Parts Thereof;. . . '

[6] 'Vehicles other than railway or tramway rolling stock, and parts and accessories thereof'

[7] 'Ships, boats and floating structures'

[8] 'Arms and ammunition; parts and accessories thereof'

- **Covid-19 Adjusted Unemployment:** During the Covid-19 pandemic, although many people were out of work as a result of the pandemic, many technically didn't meet the criteria to be unemployed as set out by the International Labour Organisation. This is in part because one is required to be actively seeking work to be considered unemployed. To address the need for estimates of the impact of the Covid-19 pandemic on the labour market, the CSO developed an unofficial measure of unemployment which counted those in receipt of the pandemic unemployment payment as being unemployed.

2.3 Problem Identification and Clarification

There are three approaches to measuring GDP: the income, expenditure and output method.

The income method measures GDP by adding together:

- The gross profit of companies and the self-employed.
- Compensation of employees which is the cost to an employer in employing someone. This includes wages and salaries but also other items such as pension contributions paid by employers.
- Taxes on products (such as VAT) minus all subsidies on products.

The expenditure method captures how much is spent on all final goods and services. Final here means that the product isn't an intermediate input to some other product. This is necessary to avoid double counting. The spending is broken down as:

- Consumer spending by individuals.
- Net expenditure by Central and Local Government.
- Capital spending, for example on building and machines. This also includes changes in the value of stocks, ie the inventory of businesses.
- Net exports, ie exports minus imports. Exports are added to GDP since these are Irish produced goods and services. Imports are subtracted seen as this is production which happened in another country and therefore is not part of Irish GDP.

Finally the output method measures GDP as:

- Output, ie the values of goods and services produced in a year. This is similar to the accounting concept of turnover.
- Minus intermediate consumption which is the value of the goods and services used up in production. For example, flour is an intermediate consumption item for bread. Output minus intermediate consumption is known as gross value added.
- Plus all taxes on products (such as VAT).
- Minus all subsidies on products (such as renewable energy subsidies).

In theory these three methods would all produce the same result. For example, expenditure by one person counts as income for another person. However in practice these methods vary considerably as they are derived from different data sources. In Ireland, the official level of GDP is defined as the average of the income and expenditure estimates. As shown above, each one of these approaches to estimating GDP is made up of many variables which sum together to produce GDP. In fact the number of variables is much larger than suggested by the high level description above. For example, the output method would in practice be calculated for each NACE[9] sector of the economy. In producing flash GDP estimates, ideally estimates would be produced for each of the relevant variables for both the income and expenditure method seen as this is what forms the official estimate in Ireland. Unfortunately, this would be too broad for this study. In addition, early estimates of Irish GDP are very difficult because of the presence of a large Multi-National Enterprise (MNE) sector. These companies can have large and unpredictable transactions which have a significant impact on Irish GDP. Given the magnitude of these transactions, they also limit the value of GDP as a metric of economic activity within Ireland. For more on this see Fitzgerald [2018]

In light of these issues, this study will consider flash estimates of modified domestic demand. This is a sub-component of GDP which is a much more useful indicator of economic activity in an Irish context because globalisation related elements are removed. Domestic demand is a concept from the expenditure method. It's the sum of all the items in the expenditure method excluding net exports. Note that we say Total Domestic Demand if the value of physical changes in stocks are included and Final Domestic Demand if stocks are excluded. As already mentioned, Ireland has a large MNE sector which has a significant impact on Irish national accounts. This can obscure what's happening in the domestic sectors of the economy. To address this problem, the CSO developed a modified domestic demand estimate which largely excludes globalisation related items [Casey, 2023]. Specifically, aircraft purchases related to leasing and purchases related to intellectual property (IP) have been removed. It's necessary to remove aircraft purchases because Ireland has a large aviation leasing industry. As a result, there are many purchases of planes by leasing companies in Ireland which are then operated in other countries. While the amounts involved are substantial, ranging from 10-16 billion a year over the period 2017-2022, it has a limited impact on the domestic economy. Similarly, large IP related transactions which generally relate to foreign-owned corporations have very limited impact on the domestic economy. However, as the amounts involved over the period 2017-2022 ranged from 37-136 billion, it can obscure developments in the domestic economy. See figure 1 for an illustration of the impact.

In conclusion, this study will consider flash estimates of modified domestic demand. Note that flash here means estimates produced within thirty days after the end of a quarter (T+30). This study will assess whether incorporating deep learning approaches can improve the accuracy of flash GDP estimates when compared to non-flash estimates. Non-flash estimate here means the official T+60 estimate as published by the Central Statistics Office.

2.4 Research Objectives

The objectives of this study are to:

[9] 'Statistical Classification of Economic Activities in the European Community'

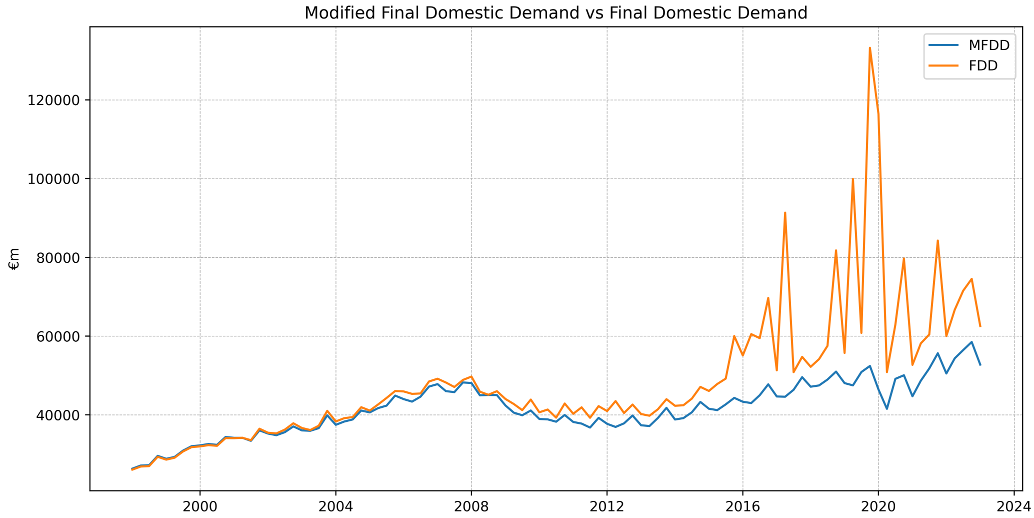


Figure 1: Final Domestic Demand vs Modified Final Domestic Demand

- Investigate the factors affecting the accuracy of flash GDP estimates for Ireland, focusing on the variable modified final domestic demand, including the availability of timely data features for training models.

- Develop and train multiple forecasting models, including traditional econometric approaches such as ARIMA and VAR, as well as deep learning models such as LSTMs, using the features identified in the first objective.

- Compare and contrast the performance of RNNs and traditional econometric models in terms of their accuracy on standard metrics such as mean absolute percentage error in predicting modified domestic demand in the context of flash GDP estimates.

- Explore and evaluate the potential benefits and costs of incorporating RNNs into the flash GDP estimation process, considering factors such as accuracy, complexity and interpretability.

- Develop actionable recommendations for improving the Central Statistics Offices flash GDP estimation process by incorporating the most effective forecasting models and data sources identified through the research.

2.5 Validity Type

Construct validity evaluates whether a particular construct represents the thing we are ultimately interested in. For this study, which is interested in the accuracy of early estimates of modified domestic demand, there is no construct in so for as the thing we are interested in is directly observable. The early estimates of modified domestic demand from the study can be compared to the official non-flash estimates with an appropriate metric. In this study, the mean absolute percentage error is utilised because of its scale invariance, ie its value doesn't depend on the scale of the data.

Content validity concerns the extent to which a measure includes all relevant aspects. As it relates to this study, an important consideration related to content validity is whether the features identified in the study are related to modified domestic demand. Some of the features identified are directly capturing economic activity which would be classified as domestic demand while others could reasonably be expected to be related to domestic demand such as unemployment. However, exploratory data analysis and feature selection is also performed to ensure content validity.

Face validity is more subjective and relates to whether a test appears to be capturing what's of interest on the surface. In so far as this research is broadly following a similar approach to flash GDP estimates as identified in both the literature review and the primary research, it is deemed to have high face validity.

Criterion validity relates to the robustness of one's output, ie how well it can predict a concrete outcome. For this study, criterion validity can be readily evaluated by comparing the predictions of the models of modified final domestic demand to the official non-flash estimates.

2.6 Ethics

There are no ethical concerns in the secondary research aspect of this study. However, the primary research for this study included in-depth interviews which poses several ethical considerations.

Firstly, informed consent was sought from participants. It was made clear to participants what the purpose and nature of the interview was in advance including that an audio recording would be made. Participants were ensured that the purpose of the audio recording was to assist production of transcripts and that it would be deleted once the transcripts were produced.

Secondly, experts could reveal confidential and sensitive information in the interviews. In the context of flash GDP estimates this could include commercially sensitive company data. To address this concern, participants were offered the opportunity to review the transcripts from the meeting.

Finally, research integrity is another important consideration. Care must be taken so as to not unintentionally misrepresent the views of experts being interviewed. We are all subject to cognitive biases such as confirmation bias which can distort how we interpret information or how it's later recalled. To help ensure research integrity, transcripts of each interview were produced and provided to the relevant participant. The participant was then able to offer an opinion as to whether the transcript provided an accurate account of the interview.

3 Literature Review

3.1 Introduction

Gross Domestic Product or GDP [CSO, 2023] is one of the key metrics of economic activity. However, to be of value to users it must be timely. It was in this context in 2003 that Eurostat published quarterly flash GDP estimates for the first time. Flash estimates [IMF, 2017] are early estimates of GDP which by Eurostat in 2003 were published 45 days (T+45) after the end of the reference period. This compares to T+60 for the non-flash estimate. The timeframe for flash estimates has since been reduced to T+30.

There are a number of modelling approaches utilised in the production of flash GDP estimates as discussed in Eurostat [2016]. These include Auto-Regressive Distributed Lag (ARDL), dynamic factor, Autoregressive Integrated Moving Average (ARIMA) and Vector Auto-Regression (VAR) models among others. This paper seeks to explore the use of deep learning methods in the context of flash GDP estimates. In so doing, it will compare these modern methods to the econometric approaches traditionally used. In this literature review, two themes will be discussed related to using deep learning for flash GDP estimates: (1) current methods used for flash GDP estimates, (2) the comparison of deep learning methods and traditional econometric methods for time series predictions.

3.2 Flash GDP Estimates

Eurostat [2016] published a methodology providing an overview of methods for producing flash GDP estimates. Eurostat outlines two aspects to forecasting, the first being model strategy, the second the availability of features which can be utilised for prediction. The two broad strategies outlined by Eurostat are direct and indirect approaches. Under a direct approach, the variable of interest is directly extrapolated using relevant features from available datasets. In the indirect approach, the extrapolation is carried out in two steps. Firstly, an indicator from the most recent quarter is extrapolated. The second step is to then utilise temporal disaggregation methods for the GDP component of interest.

3.3 Indirect approach

Under the indirect approach, Eurostat [2016] outline that most methods of temporal disaggregation are based on simple linear regression between a dependent variable and a set of explanatory variables. The dependent variable must be representable as a sum or average over a more aggregated time period while the explanatory variables are a higher frequency variable. The first model introduced is a general Auto-Distributed Lag (ADL) model with one lag of both the dependent and independent variables. The authors note that the dependent variable could also be a differenced series, see Proietti [2004]). The ADL(1, 1) model is a more general case of temporal disaggregation developed by Chow and Lin [1971], Fernandez [1981] and Litterman [1983]. As the authors explain, the Chow and Lin model is a special case where the series is modelled in levels and the ADL(1, 1) model becomes a stationary regression model with Auto-Regressive residuals of order one (AR1). The Fernandez model is a special case in which the dependent variable is a differenced series. Finally in the case of Litterman [1983], the dependent variable is a differenced series with non-stationary ARI(1, 1) residuals.

In Chow and Lin [1971], the authors were trying to solve a problem identified in Friedman [1962] on the related problems of distribution and extrapolation in the context of interpolating a time series by a related series. The problem of distribution relates to the conversion of low-frequency data such as quarterly GDP into high frequency data, say monthly GDP. To this end, related monthly series can be used to disaggregate the low frequency data. The problem of extrapolation is in finding future values of the lower frequency data by utilising higher frequency data. Chow and Lin [1971] show that the these problems can be solved in a unified fashion through an application of the theory of best linear unbiased estimation (BLUE) to a linear regression model. For more on BLUE estimates, see Theil [1971]. According to Silva and Cardoso [2001], the Chow and Lin method is still the most widely used method for disaggregating time series.

Muller [20XX] provides an alternative to Chow and Lin [1971] by utilising a maximum likelihood estimation approach. An advantage of which is the ability to estimate the model at the aggregate level. This approach maintains the advantage of Chow and Lin, particularly its moderate data demands, but simplifies it by allowing a one-step estimation on the basis of the aggregated data.

While Chow and Lin remains popular, according to Litterman [1983], Fernandez [1981] performs significantly better. Litterman [1983] proposes a slight modification of the Fernandez procedure to better account for serial correlation. For more on serial correlation, see Gubner [2006]. Litterman [1983] explains that the Fernandez approach of allowing random drift in the error process often improves estimates relative to Chow-Lin estimators. However, the random drift assumption for the high frequency data defines a filter which will remove all serial correlation in the lower frequency data residuals only when the model is correct. The author observes that in several applications, the particular model in Fernandez did not remove all serial correlation. In such circumstances, Fernandez [1981] recommends pre-filtering the data before applying the Fernandez procedure. Litterman [1983] outlines an alternative approach which doesn't rely on an ad hoc search for a filter which won't leave serial correlation. To test the validity of the approach, quarterly data which is itself the average of observed monthly values was disaggregated. Accuracy of the procedure could then be calculated as the true values were known. This test was conducted for four methods including Chow and Lin and Fernandez using six data sets. The data sets chosen were from National Income Accounts and the Flow of Funds Accounts. Accuracy was measured in terms of the mean square error of distributed levels from actual levels and the mean square error of the changes in the distributed values from the actual changes. The results indicated that the Litterman procedure had the smallest mean square error by both measures for four of the six datasets. In the three cases which had positive Markov parameters, the average reduction in the mean square error for the level measure was 13 per cent compared to the next best model.

3.4 Direct approach

Where data is available, either partially or in full, Eurostat [2016] outline a number of models which could be used under a direct approach. An Auto-regressive Distributed Lag (ADL) model is discussed in which lagged values of the dependent variable and current and lagged values of one or more explanatory variables are included among the regressors. As discussed by the authors, most macroeconomic time series have a trend and are therefore non-stationary. See Investopedia [2023] for more on non-stationary time series. The issue with non-stationary time series is that it can lead to spurious regression [Granger and Newbold, 1974] in which significant relationships between variables are found even when in reality none exist. As a result, it is necessary to test for a type of long run relationship called cointegration, see CFI [2023]. If cointegration is found, the ADL model can be expressed in a re-parameterised form called the error-correction model (ECM). This form captures both the long run equilibrium towards which the model is heading but also the short run dynamics along the way.

A procedure for estimating the ECM was introduced by Engle and Granger [1987] which involves four steps. Firstly, the order of integration is established. This is the number of times a series must be differenced in order to induce stationarity. See Hamilton [1994] for more information. To test whether a series is stationary, a number of statistical tests can be used including an Augmented Dickey-Fuller (ADF) test [Dickey and Fuller, 1979], a Phillips-Peron test [Phillips and Perron, 1987] and the KPSS test [Kwiatkowski et al., 1992]. Secondly, if step one indicates that the variables are integrated of the same order, then the long run relationship should be estimated. The suitability of the model should be checked by testing residuals for serial correlation and heteroscedasticity. There are a number of tests for serial correlation including a Durbin-Watson test, see Durbin and Watson [1950] and Durbin and Watson [1951], a Breusch-Godfrey test [Breusch, 1978] and Godfrey [1978]. For heteroscedasticity, a White test could be implemented [White, 1980]. In the third step, the order of integration of the residuals must be determined. The residuals must be stationary for the variables to be cointegrated. The final step is to estimate the error correction model. One drawback of the Engle-Granger approach is that if there are multiple explanatory variables then there may be multiple cointegration relationships. A Johansen test can be used to determine the number of cointegrating relationships [Johansen, 1991].

Another approach identified is to use dynamic factor models. The advantage here is that high-dimensional datasets can be included because dimensionality reduction techniques such as principal components analysis are utilised. The first stage of developing a dynamic factor model is feature extraction, ie determining which factors are to be included within the model. Factor selection can be based on information criteria such as Bayesian Information Criteria (BIC), see Bhat and Kumar [2010]. For a review of information criterion rules, see Stoica and Selén [2004]. For factor based regression criteria, see Groen and Kapetanios [2009]. For datasets which have missing values, Josse and Husson [2012] present a procedure for selecting the optimal number of principal components. Once the factors have been identified, the model can be estimated such as in Chamberlain and Rothschild [1982]. It's also highlighted that more sophisticated models could be developed by combining aspects of dynamic factor models and error correction models.

For instances where no data is available, Eurostat [2016] highlight pure forecasting methods such as ARIMA models and structural time series (STS) models. ARIMA models are a popular class of time series model due to their simplicity and generalisability. An ARIMA model describes a time series as a function of autoregressive (AR) and moving average (MA) terms where the 'I' in ARIMA refers to the order of integration, ie the number of times the time series must be differenced in order to induce stationarity. The number of lags of the AR and MA processes are referred to as the order of the respective effects. So for example, an ARIMA (1, 1, 1) model contains one lagged autoregressive term, one lagged moving average term and it was necessary to the difference the data once to induce stationarity. An autoregressive process is a linear regression where the explanatory variable is simply lagged values of the dependent variable. Similarly, a moving average process is a linear regression where the explanatory variables are lagged shocks or error terms.

The ARIMA modelling process is encapsulated in the Box-Jenkins procedure [Box and Jenkins, 1976]. The procedure can be summarised in three steps; model identification, estimation & diagnostic checking and finally forecasting. Model identification includes determining the order of integration and the order of the AR and MA processes. The order of the AR and MA processes can be determined by inspection of the autocorrelation function (ACF) and the partial autocorrelation function (PACF). The ACF is the correlation between a time series and lagged values of itself. It starts with a zero lag which results in a correlation of 1, ie a series is perfectly correlated with itself. It then proceeds to calculate correlation with higher order lags. PACF is a similar concept which gives the partial correlation of a series with its own lagged values where partial here refers to the fact that other lags are being controlled for. See for example Duke [2023a] for a discussion of how the ACF and PACF can be used to identify model order.

In practice, the ACF and PACF may give a conflicting or ambiguous picture of what the best model order should be. In such circumstances, a hyperparameter tuning approach can be taken with an information criteria such as the Akaike Information Criterion (AIC) serving as the loss function. For more on the AIC, see Akaike [1973], Akaike [1974] and Akaike [1985]. For more on hyperparameter tuning, see Feurer and Hutter [2019], Claesen and Moor [2015] and Bergstra and Bengio [2012]. For more on loss functions, see Hastie et al. [2009]. For estimation a number of procedures are available including maximum likelihood estimation (MLE) and the Yule-Walker equations. For more on MLE, see Young [2019], Hendry and Nielsen [2007] and Ward and Ahlquist [2018]. For Yule-Walker equations, see Theodoridis [2015]. For diagnostic checking, the assumptions of ARIMA can be assessed such as residuals being independent and normally distributed. For a discussion in the context of linear regression, see Duke [2023b]. The statistical significance of included coefficients can also be checked. If the model passed diagnostic checks then it's ready for forecasting. Extensions of ARIMA models to include additional explanatory variables are referenced in passing in Eurostat [2016]. For a fuller discussion see Hyndman and Athanasopoulos [2021].

Structural time series (STS) models offer an alternative to ARIMA models in which a series is expressed in terms of a trend, cycle, seasonal and irregular term. See Scott and Varian [2013] and Scott and Varian [2015] for more details. STS models also have the benefit of modernising other popular forecasting methods such as exponential smoothing, see Holt [1957] and Holt [2004].

Multivariate extensions of these models, such as Vector Autoregression (VAR) are also discussed. VAR models are a multivariate extension of AR models proposed by Sims [1980]. VARs are similar to AR models in that a dependent variables is treated as a function of lagged values of itself except that in the case of VAR, the dependent variable is a vector of time series variables. Where cointegrating relationships exist, the dynamic adjustment towards the long run equilibrium should be accounted for and such models are typically referred to as Vector Error Correction Models (VECM). Limitations of VAR for forecasting purposes include the fact that the number of parameters increases rapidly as the number of time series expands or the lag length increases. This can result in inefficient estimates of the parameters and wide confidence intervals around forecasts. Bayesian

VAR (BVAR) models have been proposed in response to these issues, see Karlsson [2013]. Other extensions of VAR have been developed such as Factor-Augmented Vector Autoregression (FAVAR) which enables a large number of explanatory variables to be included, see Bernanke et al. [2005] and Bai et al. [2014].

While deep learning methods are not mentioned in Eurostat [2016], they're use in flash estimates has been explored elsewhere. Richardson et al. [2019] examined whether machine learning algorithms could improve nowcasts[10] of real GDP in New Zealand. A number of popular machine learning algorithms were trained on a large real-time dataset of around 550 indicators. Accuracy was compared to a suite of statistical models such as those outlined by Eurostat [2016]. The authors found that the machine learning algorithms outperformed the statistical benchmarks. However, the only deep learning method considered was an Artificial Neural Network (ANN) which is not state-of-the-art for time series forecasting. Longo et al. [2022] incorporated state-of-the-art methods such as Long-Short-Term-Memory (LSTM) in their approach to forecasting US GDP. The authors concluded that an ensemble approach combining LSTM with dynamic factor models worked best.

3.5 A Comparison of Deep Learning and Traditional Econometric Methods

Deep learning methods range from general purpose Artificial Neural Networks (ANN) to architectures designed with particular data in mind. In the case of ordered sequence data such as time series, state of the art models are classes of Recurrent Neural Networks (RNNs) which include Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models. A characteristic feature of RNNs is that output from a layer can be fed back as an input. This makes it possible for sequential data to be fed through the model in a feedback loop. However, RNNs can result in the exploding or vanishing gradient problem when being trained on long sequences of data. One solution to this problem is LSTM models. The key difference compared to traditional RNNs is that recurrent units are replaced with a more complex structure called an LSTM cell. These cells are composed of four gates, 'forget', 'input', 'cell' and 'output' gates. The key architectural difference as it relates to the exploding or vanishing gradient problem is that LSTM cells are designed to emphasize additive as opposed to multiplicative operations. In traditional RNNs, gradients are backpropagated through the use of the chain-rule which involves repeated multiplication. It's this process which can lead to exploding or vanishing gradients. In contrast, LSTM cells are updated in such a way as to avoid this repeated multiplication. For more on LSTM models, see Hochreiter and Schmidhuber [1997].

An alternative solution to the exploding or vanishing gradient problem is GRU models. GRU models were introduced by Cho et al. [2014]. GRU models are typically more efficient than LSTM in terms of training time but their performance on a task depends on the particulars of that task, see for example Cahuantzi et al. [2023].

There is an extensive literature comparing the performance of modern deep learning methods to traditional econometric approaches to time series forecasting. Yamak et al. [2019] compare ARIMA, LSTM and GRU models for forecasting Bitcoin prices. In terms of performance metrics, based on Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) , Yamak et al. [2019] find that the ARIMA model outperforms both deep learning approaches. One possible reason for this highlighted by the authors is that the models were tested on a relatively small dataset

[10] Nowcasts are a similar concept to flash GDP estimates.

whereas Recurrent Neural Networks (RNNs) typically perform well on larger datasets. The authors also recognise that the selected features might not be sufficient to predict Bitcoin prices accurately. While crypto-currency markets don't appear to be as efficient as stock markets in so far as they don't follow random walks [Aggarwal, 2019], they would none the less be very difficult to predict which limits the value of the Yamak et al. [2019] comparison.

The past underperformance of deep learning methods relative to traditional econometric methods in time series forecasting was also referenced in Hewamalage et al. [2021]. However, the authors suggest this is changing. With growing amounts of big data, not necessarily in time span but in terms of the number of related variables, RNNs are now competitive with traditional methods. However, competitive might not be good enough. As Hewamalage et al. [2021] suggest, the fitting of RNNs is not as straightforward and automatic as state-of-the-art univariate forecasting benchmarks such as ARIMA.

Almosova and Andresen [2019] used RNNs to forecast US CPI inflation. The results indicated that an LSTM outperformed a number of traditional models including a seasonal autoregressive model (SARIMA) based on the root mean squared forecast error. These results are consistent with other studies which considered inflation forecasting such as Chen et al. [2001], Nakamura [2005] and McAdam and McNelis [2005]. These results are also consistent with Siami-Namini et al. [2018] who showed that deep learning based algorithms such as LSTM outperformed traditional methods such as ARIMA with the average reduction in error rates between 84-87 percent. Alon et al. [2001] also compared deep learning approaches (specifically an ANN) to traditional methods including exponential smoothing, ARIMA and multivariate regression. The data considered was US aggregate retail sales which has strong trend and seasonal patterns, making this study of particular relevance in the context of flash GDP estimates. The authors conclude that on average the ANNs have superior performance.

The above results claiming superior performance of deep learning models for time series forecasting is not supported by Teräsvirta et al. [2003]. Teräsvirta et al. [2003] compared traditional and deep learning methods on 47 monthly macroeconomic variables of the G7 economies. The results were mixed with the authors concluding that no model dominated the others with the best model varying across countries, variables and forecast horizon. Stock and Watson [1998] also considered a wide range of economic variables in their forecasting comparisons. The data consisted of 215 U.S. monthly macroeconomic time series at three forecasting horizons over the period 1959-1996. There were four classes of models considered: autoregressions, exponential smoothing, ANNs and smooth transition autoregressions. The authors conclude that the best overall model in terms of accuracy was achieved by autoregressions where pre-processing included unit root tests, ie tests for stationarity. However, the authors note that this performance could be improved if the different methods were combined in an ensemble approach, similar to the findings of Longo et al. [2022].

Sharda and Patil [1992] look at the results of a forecasting competition between a neural network and a Box-Jenkins forecasting system. The data consisted of seventy five series with the results showing that a simple neural net could forecast about as well as the traditional Box-Jenkins approach. Similar results were found in Foster et al. [1992] and Kang [1992]. Hill et al. [1996] also conducted an experiment in which time series forecasts produced by neural networks were compared to traditional statistical time series methods. The data used was from a major forecasting competition [Makridakis et al., 1982] and the neural networks were estimated using the same ground rules as for the competition. Across both monthly and quarterly time series, neural networks did a significantly better job than traditional methods. However, the accuracy of neural network and traditional models were comparable on the annual data. The authors believe that a key factor in the out performance of deep learning methods was their ability to handle discontinuities. Ahmed et al. [2010] also consider data from the M3 time series competition data containing around a thousand time series. While the authors note that there have been many comparison studies which compare neural networks with traditional forecasting methodologies, these studies have typically been confined to a basic neural network architecture. Ahmed et al. [2010] considers novel machine learning algorithms including Bayesian neural networks and generalized regression neural networks. The authors find significant differences in accuracy between the different models, concluding that the multilayer perceptron and the gaussian process regression performed best. The authors also note that choice of pre-processing methods can have a significant impact on results.

3.6 Conclusions

Eurostat [2016] provided a very useful overview of the methodologies employed within the European Statistical System for producing flash GDP estimates. However, the authors remained largely agnostic as to which models work best. There is also no use made of deep learning models despite there being a considerable literature on the use of such methods for forecasting with time series data. Many of these studies include comparisons of forecast accuracy between deep learning methods and traditional time series econometrics methods. Overall the literature reviewed in this study suggested that deep learning methods are competitive with traditional econometric methods in the context of time series forecasting. The results range from traditional methods moderately outperforming deep learning algorithms on accuracy metrics to radically underperforming in Siami-Namini et al. [2018]. Some papers have shed light on when deep learning methods can be expected to outperform traditional methods. For example, Yamak et al. [2019] highlights that deep learning models perform better with larger datasets. Hill et al. [1996] show that deep learning models handle discontinuities better than traditional methods. Others such as Teräsvirta et al. [2003] highlight that the best model depends on the particular dataset and forecast horizon being considered. Some papers including Longo et al. [2022] and Stock and Watson [1998] emphasize that the best performance can be achieved through an ensemble approach which combines different methods. However, as highlighted by Hewamalage et al. [2021], it's not all about accuracy. Traditional methods have advantages around ease of use, making them suitable for non-expert users as they are robust, efficient and automatic.

A significant limitation of the studies reviewing comparisons of deep learning and traditional time series econometrics methods is that many were using a relatively simple ANN architecture which is not best in class. A better comparison would be between traditional methods and LSTM and GRU models, as some of the papers reviewed have done.Another significant limitation of the literature is that although economic time series is a popular domain area for studies comparing forecast accuracy of deep learning and traditional methods, most don't consider GDP estimates. This might in part be due to the fact that GDP is typically published only quarterly which limits the amount of data available. Many of the economic variables seen in the literature review were higher frequency variables such as inflation or retail sales which are typically published on a monthly basis.

While there has been some exploration of deep learning methods in the context of flash GDP estimates, these methods are not used at all in the European Statistical System. Future research could explore the effectiveness of deep learning algorithms in the context of flash GDP estimates within the EU. There is also a need for greater research on when and why deep learning algorithms outperform traditional methods on certain time series forecasting tasks.

4 Methodology

A number of models were trained to produce flash estimates of modified final domestic demand. What follows is a primer on these models and the related concepts necessary to understand how the parameters of the models are estimated.

4.1 Machine Learning

To develop a baseline model to which more specialised time series models could be compared, linear regression models were explored. Regression models seek to define a linear relationship between a dependent variable and one or more independent variables, see equation 3. There are a number of equivalent terms for the variables in a regression model, in a machine learning context it's common to refer to the dependent variables as the target variable and the independent variables as the features. A common approach to estimating the parameters in equation 3 is Ordinary Least Squares (OLS). OLS chooses the parameters so as to minimize the sum of the squared residuals where residuals are the difference between the observed and predicted values, see equation 4. As OLS is squaring the residuals, this has the effect of penalising particularly large errors.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \tag{3}$$

where:

- Y is the dependent variable
- $X_1, X_2, \ldots, X_p$ are the independent variables
- β_0 is the y-intercept
- $\beta_1, \beta_2, \ldots, \beta_p$ are the slope coefficients for the independent variables $X_1, X_2, \ldots, X_p$ respectively
- ϵ is the error term

$$\min_{\beta_0, \beta_1, \ldots, \beta_p} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}))^2 \tag{4}$$

A slightly more sophisticated regression model is Least Absolute Shrinkage and Selection Operator (LASSO). Unlike traditional least squares regression, LASSO performs feature selection and regularization to avoid over-fitting. Over-fitting is the term used to describe a model which fits the training data well but performs poorly on unseen test data. This phenomenon occurs when models are overly complex, such as containing too many parameters. As a result, they learn the detail and noise of the training data, leaving them unable to generalise to unseen data. LASSO models solve both the feature selection and over-fitting problem by adding a slope parameter penalty to the model. See equation 5 which is very similar to equation 4 but note the inclusion of the regularization parameter λ. Inclusion of this parameter penalises large slope coefficients, hence the word shrinkage in Least Absolute Shrinkage and Selection Operator. Slope parameters can be shrunk to zero and hence effectively drop out of the model. It is in this sense that LASSO can perform feature selection. Shrinking slope parameters or even dropping features from the model altogether has the effect of reducing model complexity and helps to avoid over-fitting.

$$\min_{\beta_0,\beta_1,\ldots,\beta_p}\left[\sum_{i=1}^{n}(y_i-(\beta_0+\beta_1x_{i1}+\beta_2x_{i2}+\cdots+\beta_px_{ip}))^2+\lambda\sum_{j=1}^{p}|\beta_j|\right] \tag{5}$$

The choice of λ is determined through cross validation. The most common approach is k-fold cross validation where data is split into k groups or folds. One of the folds is held out as a test set and the rest of the folds are used for training the model. The model can then be evaluated on the test set and scored. This process repeats until each of the folds has been used as a test set. The results can then be averaged. This process can be performed for various values of λ and the value which gives the best average score on the chosen metric can be used in the final model. The advantage of such an approach is that the model trains on multiple train-test splits which gives a more reliable indication of model performance. However, this approach is not suitable for time series data, where the temporal ordering of the data needs to be preserved. In this case, time series cross validation can be performed which maintains ordering because the kth split returns the first k-folds ordered in time with the $(k+1)th$ fold used as the test set. As such, successive training sets are supersets of those that came before them.

A similar model to LASSO is Ridge regression. Ridge regression also includes a regularization parameter but instead of the L1 norm used in LASSO, the L2 norm is used, see equation 6. As a consequence of the choice of norm, Ridge regression can't shrink coefficients to zero so will not perform feature selection like LASSO. However values can be shrunk close to zero and thus those features will have little impact on the model, even if they're still included. As with LASSO, Ridge regression can reduce overfitting by increasing bias in the model which can reduce variance when the model is applied to unseen data.

$$\min_{\beta_0,\beta_1,\ldots,\beta_p}\left[\sum_{i=1}^{n}(y_i-(\beta_0+\beta_1x_{i1}+\beta_2x_{i2}+\cdots+\beta_px_{ip}))^2+\lambda\sum_{j=1}^{p}\beta_j^2\right] \tag{6}$$

4.2 Time Series Econometrics

AutoRegressive Integrated Moving Average (ARIMA) models are very popular for time series forecasting. In an ARIMA model, a dependent variable is modelled as a function of lagged values of itself and lagged values of error terms. An ARIMA model can be characterised by three terms: p, d, q.

- p is the order of the autoregressive process, ie the number of lagged values of the dependent variable included in the model.

- d is the order of differencing, ie the number of times the data has had past values subtracted so as to induce stationarity. Stationarity is an important assumption of the model and requires that the mean, variance and covariance of a series be constant across time.

- q is the order of the moving average process, ie the number of lagged forecast errors.

See equation 7 for a generic ARIMA model. Values for p and q can be determined by examining Autocorrelation Functions (ACF) and Partial Autocorrelation Functions (PACF). An ACF plot shows the correlation between a series and its lags. A PACF plot shows the correlation between a series and its lags that is not explained by correlations at all lower-order lags. In practice though, time series data will not exactly be generated from an ARMA process and interpretation of the ACF and PACF plots will not be so straightforward. In such instances, a time series cross validation approach can be taken where many models are estimated and the model eventually chosen will be that which performs best on average at predicting values in the test set. The value d can be chosen through experimentation and formal statistical tests such as Augmented Dickey-Fuller (ADF). ADF tests the null hypothesis that a series is stationary. If the null is rejected, the series can be differenced and ADF again performed to assess stationarity of the differenced series. This process can be continued until the series is stationary.

$$\Delta^d y_t = \phi_1 \Delta^d y_{t-1} + \ldots + \phi_p \Delta^d y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} \tag{7}$$

where: $\Delta^d y_t$ is the differenced series (d-th difference of y_t),
$\phi_1, \ldots, \phi_p$ are the parameters for the AR (AutoRegressive) model,
$\theta_1, \ldots, \theta_q$ are the parameters for the MA (Moving Average) model,
ϵ_t is the error term at time t,
p is the order of the AR model,
d is the order of differencing,
q is the order of the MA model.

The parameters of an ARIMA model are chosen through Maximum Likelihood Estimation (MLE). In MLE, the likelihood of observing a given set of data conditional on a chosen set of parameters is calculated. In the case of ARIMA, the likelihood function is based on the likelihood the errors from the model are white noise. Optimal values for the parameters, ie the coefficients of the AR and MA terms can then be found by maximising the likelihood function. Calculating likelihoods involves the product of several probabilities. As probabilities are numbers between 0 and 1, their product can quickly become very close to 0 which can lead to issues of numerical stability. Specifically it can result in underflow which is a situation where numerical precision is lost because the values are closer to zero than the smallest possible floating point representation of the value possible on a computer. To avoid this problem, typically the log of the likelihood function is maximised. Taking the log doesn't change the location of the maximum values and as logs can convert multiplication to addition, it avoids the underflow problem. The actual process of maximising the log-likelihood function can be achieved with an optimisation algorithm such as gradient descent.

In gradient descent, parameters are initialised by guesses or in practice random numbers which are then iteratively updated. The updating occurs through calculating the gradient of the log-likelihood function which gives the direction in the parameter space where the function increases most rapidly. The initial guesses can be updated by taking a step in the direction given by the gradient, the size of the step is determined by a learning rate which is itself a hyper-parameter, ie chosen by the modeller.

ARIMA models can be easily extended to capture series which contain seasonality. This is achieved by combining an ARIMA model with another ARIMA model which uses seasonal differencing and seasonal lags based on the number of periods per season. For instance, in the case of quarterly economic data which exhibits a seasonal pattern, a first order seasonal difference would require subtracting from a given observation the value for the same quarter a year earlier. Similarly, a seasonal AR lag of order 1 involves using the value of the series from the same quarter last year to make a prediction.

Both ARIMA and SARIMA models can be further extended to include exogenous explanatory variables. Such models are called ARIMAX or SARIMAX. An ARIMAX model is estimated in much the same way as an ARIMA model but now there are additional parameters for each of the exogenous explanatory variables included in the model.

An alternative approach to ARIMAX models are Vector-AutoRegression (VAR) models. VAR models capture the linear interdependencies between multiple time series by allowing for each variable to be a function of past values of itself and past values of the other variables. As with ARIMA models, the data is required to be stationary, otherwise the regressions will likely be spurious. See equation 8 for a VAR model of order p. There are a number of advantages to VAR models including their simplicity and reduced form nature. VAR models don't make assumptions around which variables are endogenous and which are exogenous. All variables are treated the same way which simplifies the model. Similarly, VAR is reduced form in that it is a-theoretical, it doesn't explicitly incorporate economic theory in the structure of the model. VAR models can be estimated with OLS with the particular lag structure of the model determined through time series cross validation.

$$\mathbf{y}_t = \mathbf{c} + \Phi_1 \mathbf{y}_{t-1} + \Phi_2 \mathbf{y}_{t-2} + \ldots + \Phi_p \mathbf{y}_{t-p} + \mathbf{u}_t \tag{8}$$

where:

- $\mathbf{y}_t$ is the k-dimensional vector of the time-series variables at time t.
- $\mathbf{c}$ is the k-dimensional vector of constants.
- Φ_i are $k \times k$ coefficient matrices for $i = 1, \ldots, p$.
- $\mathbf{u}_t$ is the k-dimensional vector of white noise error terms.
- p is the number of lags.

4.3 Deep Learning

Artificial Neural Networks (ANNs) which can handle the sequential nature of time series data have been developed. The basic approach is a Recurrent Neural Network (RNNs). The architectural innovation in RNNs which allows them to handle sequential data is a feedback loop. In essence, a simple RNN works as follows:

1. An input (x_1) is fed into the network.

2. This input has a weight (w_1) and bias (b_1) term applied to it and is passed to an activation function. Let's call the output of the activation function y_1.

3. When the next input (x_2) is fed into the network, it has the same weight (w_1) and bias (b_1) term applied to it. However, before this number is passed to the activation function, we add the output of step 2 (y_1) with a weight (w_2) applied to y_1. This is the feedback loop which allows RNNs to handle sequential data of any length. It's known as the hidden state of the network.

4. Data is fed into the network until all the data in the training sequence is utilised. At which point the output of the activation function, rather than being looped back into the processing of the next input, has a new weight (w_3) and bias (b_2) term applied and gives the prediction for the next value in the sequence.

Note that the weights and biases for all the inputs in the training sequence are shared. RNNs are not often used in practice because for long sequences of data they can lead to a vanishing or exploding gradient problem. To see this, consider $w_2 = 2$ in the above simple example. The output of each hidden state gets doubled by w_2. In a long sequence of data, this constant doubling of the output of the hidden state can lead it to increase rapidly. As this number will feature in the calculation of gradients in the backpropagation process, it can lead to an exploding gradient problem. As a gradient becomes too large, it becomes difficult to find the parameters which optimise the loss function as the step size is too large. If $w_2 < 1$, a long sequence of data would result in a vanishing gradient problem. Here the issue is that it becomes difficult to find the parameters which optimise the loss function because the step size is too small. As the step size becomes smaller and smaller, training times get longer and longer. Fortunately methods have been developed which avoid the exploding/vanishing gradient problem. The most popular of these is the Long-Short-Term-Memory (LSTM) model.

In essence, a simple LSTM model works as follows.

- **Forget Gate:** The first stage of an LSTM model is the forget gate which determines how much of a long term memory should be forgotten. This is achieved by passing a short term memory from the so called hidden state (s_0) and an input (x_1) with respective weights w_1 and w_2 which are then summed. A bias term (b_1) is then added before the results are passed to a sigmoid activation function, see equation 9. A sigmoid activation function takes any real number and returns a number between 0 and 1 and can thus be interpreted as a proportion. This proportion is applied to the long-term memory, ie a particular value stored in the so called cell state, with the result being the updated long-term memory.

- **Input Gate:** The input gate determines how the long-term memory should be updated with new memories. A potential new memory is created by first taking s_0 and x_1 with weights w_3 and w_4 applied and then summing the result. A bias term (b_2) is then added before the result is passed to a $\tanh$ activation function, see equation 10. A $\tanh$ activation function returns a value between -1 and 1. Just as there was a forget gate to determine how much of the existing long term memory to keep, the potential long term memory also has a forget gate which determines how much of the potential memory to keep. The output of the input gate and the forget gate is then added to the existing long term memory to get a new long-term memory.

- **Output Gate:** The final stage of the LSTM unit is the output gate. The output gate determines how the short term-memory is updated. A new potential short-term memory is created by passing the long term memory to a $\tanh$ activation function. Once again, this potential short-term memory has a forget gate which determines what proportion of the new short term memory to keep. The output becomes the new short term memory.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{9}$$

$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{10}$$

The cell state, hidden state, input and output gates together with their corresponding forget gates define an LSTM unit. There is then an LSTM unit for each input in a given sequence of data. Each sequence of data uses an LSTM unit with the same weights and biases. The cell state of an LSTM, which represent long-term memories, lacks weights and biases which is how LSTM models avoid the exploding/vanishing gradient problem for long sequences of data.

4.4 Backpropagation and Gradient Descent

In the discussion of RNNs and LSTMs, the concept of weights and biases was central. These are parameters which need to be learned from the data. The particular weights and bias values learned are done so in order to optimise a particular loss function. A loss function describes quantitatively how well a particular neural network is predicting the target variable. Popular choices for regression problems include mean squared error and mean absolute percentage error, see equation 11 and equation 12 where y is the value and $\hat{y}$ is the predicted value. A key concept in the optimisation of a loss function is backpropagation. Backpropogation consists of:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{11}$$

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{12}$$

- **Forward Pass:** In the forward pass, each training example is fed through the network.

- **Calculate Error:** The error from the loss function for each training example is calculated.

- **Backward Pass:** To update the weights, it is necessary to know how much each particular weight is contributing to the error from the loss function. This can be achieved by using the chain-rule from calculus to calculate the partial derivatives of the loss function with respect to each of the weights. The partial derivatives are calculated starting with the most recent hidden layer and working backwards through the network. It's in this sense that the method propagates the error backward through the network and hence the name.

After the backward pass, the next step in finding optimal parameters depends on the particular optimisation algorithm being used to minimise the loss function. A simple and popular algorithm is Stochastic Gradient Descent (SGD). In this algorithm, an initial set of weights would have been initialised for the first forward pass. Then for each epoch, ie each complete pass through of the training data, the data should be shuffled. Then the three steps identified for backpropagation would be completed. The SGD algorithm would then continue on with the next step being:

- **Update weights:** Weights are updated based on equation 13 where α is the learning rate, ie a hyperparameter that controls the size of the step in the direction of the gradient.

$$w_{\text{new}} = w_{\text{old}} - \alpha \nabla L(w_{\text{old}}, x_i, y_i) \tag{13}$$

where:

- w_{new} and w_{old} are the updated and current weights, respectively.
- α is the learning rate.
- $\nabla L(w_{\text{old}}, x_i, y_i)$ is the gradient of the loss function with respect to the weights, evaluated at a single training example (x_i, y_i).

This process of updating weights would continue for a set number of epochs or until convergence. As weights are being updated based on individual training examples, the path to convergence can be noisy. For this reason, pure SDG is rarely used in practice. Instead, Mini-Batch Gradient Descent (MBGD) is more likely to be utilised. In MBGD, the dataset is split into batches of n training examples. The steps outlined above for SGD are largely the same for MBGD, except that instead of updating the weights after each training example, the gradient would be averaged over all training examples in each batch. The weights are then only updated for each batch rather than each training example. This process can lead to faster convergence.

There are many more sophisticated variants of SGD and MBGD which are all based on utilising information about the gradient of a loss function with respect to the weights of a network. One of the most popular such methods is Adaptive Moment Estimation (Adam). The Adam algorithm is a mixture of two optimisation algorithms methods known as momentum and RMSprop, it consists of the following steps:

- **Momentum:** Adam maintains an exponentially decaying average of past gradients. See equation 14.
- **RMSprop:** Adam also maintains an exponentially decaying average of past squared gradients. See equation 15.
- **Bias Correction:** Since m_t and v_t are initialised at zero, they are biased towards zero. Adam corrects these biases. See equations 16 and 17.
- **Weight Update:** Adam updates the weights as in equation 18 where α is the learning rate and ϵ is a small constant to prevent division by zero.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{14}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{15}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{16}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{17}$$

$$w_{\text{new}} = w_{\text{old}} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{18}$$

where:

- m_t and v_t are estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradients, respectively.
- β_1 and β_2 are exponential decay rates for moment estimates.
- α is the learning rate.
- ϵ is a small constant to prevent division by zero.
- g_t is the gradient at time step t.

5 Implementation & Results

5.1 Data

The variables used for this study were outlined in the secondary data section of the Research Design chapter. These variables were all available in the Central Statistics Office data dissemination tool, PxStat. Application Programming Interface (API) queries to PxStat return complicated nested JSON files. To simplify obtaining the data, a module was developed which contained a function which accepted one argument, the name of a table in the PxStat database, and returned a pandas dataframe.

While the response variable of interest is a quarterly series, the possible explanatory variables are monthly. For the purposes of flash estimates which have to be produced within 30 days of the end of the quarter, only data with higher than quarterly frequency would be of use. The potential features were up sampled to a quarterly frequency by averaging the months of the quarter.

5.2 Exploratory Data Analysis

While modified domestic demand and many potential features are available back to 1995 or close to it, some of the explanatory variables had many fewer observations. For example, the series on dwelling completions was only available to 2011. The series on the Monthly Services Index (MSI) and departures from the country either by air or sea were of similar length. The number of observations for the study was already limited so it was decided to drop the three aforementioned shorter series.

As part of exploratory data analysis, the series were all plotted against time. See figure 2 for a plot of the dependent variable. From visual inspection, the series exhibited heteroscedasticity. Heteroscedasticity is where the variance of a dataset changes as the level of the series changes. Appropriate transformations of a dataset which can induce homoscedasticity can contribute to making relationships with explanatory variables more linear which can be useful when linear regression models are being utilised.

Histograms of the data were also produced with a kernel density estimate overlaid, see figure 3. Many of the series had long tails which reflects the series trending upwards over time. While the unemployment data had a long tail reflecting the fact that most of the time the economy is at or close to full employment[11].

To correct for the heteroscedasticity in the data, a box-cox transformation was utilised. Box-cox also serves to bring data closer to a normal distribution, see figure 4. However, normality of data is not an assumption of regression based models. Box-cox is a generalisation of power transformations based on a parameter λ, see equation 19. In the case of modified final domestic demand, the estimated value of λ was 0.9. A value of 1 would mean no transformation was applied. From visual inspection of the data post transformation, it is difficult to see a change. However, more significant transformations were applied to some of the other series in the dataset.

[11] Full employment is an economics concept which doesn't suggest everyone who wants a job, has one. Instead it refers to the lowest rate of unemployment consistent with a non-accelerating rate of inflation.

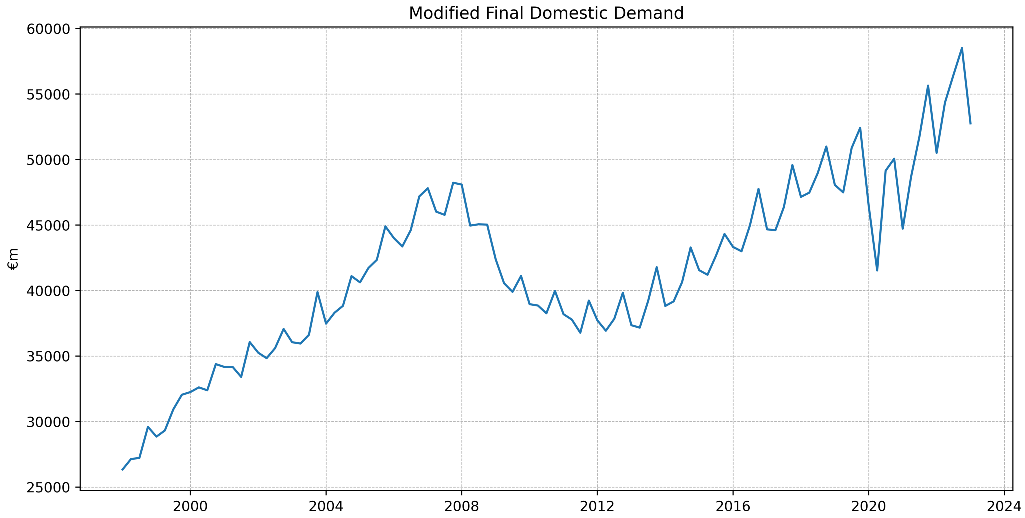


Figure 2: Modified Final Domestic Demand

$$y' = \begin{cases} \frac{y^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(y) & \text{if } \lambda = 0. \end{cases} \tag{19}$$

where:

- y' is the transformed value.
- y is the original value.
- λ is the Box-Cox parameter.

Stationarity is a key assumption in many time series modelling approaches. It's also important in the context of regression modelling so as to avoid spurious regression which is where two or more non-stationary time series are falsely seen as related due to a shared trend. Stationarity is a statistical property of time series data which requires the mean, variance and autocorrelation of a time series to be constant over time. From visual inspection, most of the time series are not stationary which is typical for economic time series. However, a formal statistical test was also conducted to test for stationarity. Augmented Dickey Fuller tests the null hypothesis that a unit root is present in a time series where a unit root indicates a trend in the series. The key concept is to run a regression to assess whether the change in a dependent variable can be explained by its lagged values, see equation 20. The null hypothesis is that $\gamma = 0$. The alternative hypothesis is that $\gamma < 0$. If $\gamma < 0$ it implies that the larger the value of the series, the greater the likelihood that a change in the series will be negative, ie there is some reversion to the mean. Suggesting that the series is stationary. On the other hand, if $\gamma = 0$, this reversion to the mean is not happening and therefore the series is likely non-stationary. Using a p-value of 0.05, augmented dickey fuller was run for each variable in the dataset. All variables were found to be non-stationary. Note that a p-value of 0.05 means that under the assumption that the null hypothesis is true, there is a 5 per cent chance that

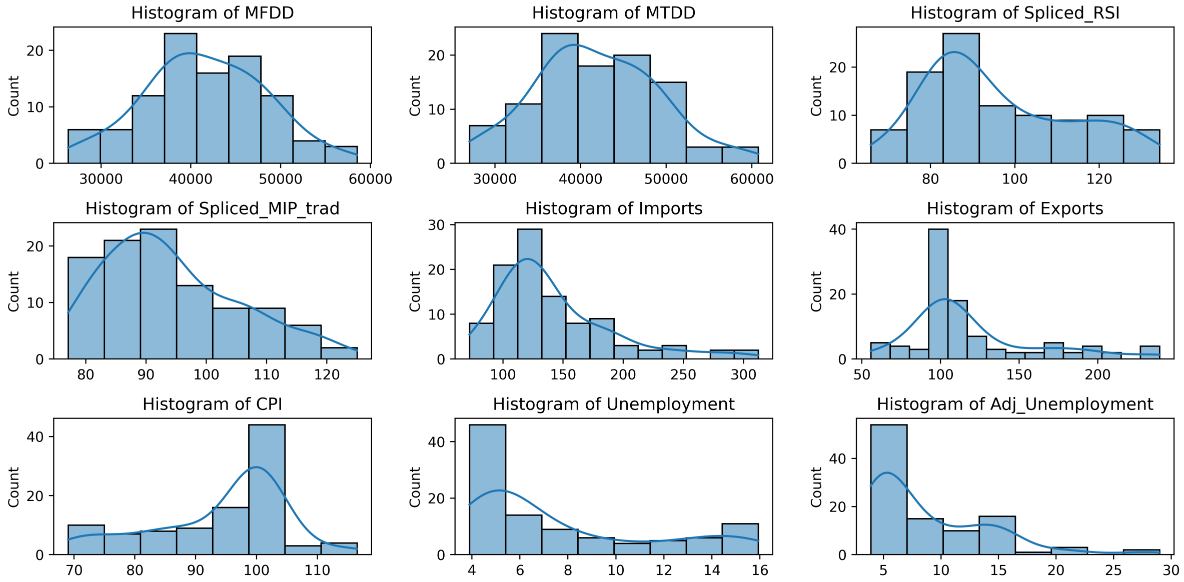


Figure 3: Histograms of Data

we would observe results at least as extreme as those observed. In this case, all p-values were above 0.05 and thus the null hypothesis was not rejected.

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \delta_2 \Delta y_{t-2} + \cdots + \delta_p \Delta y_{t-p} + \varepsilon_t \tag{20}$$

where:

- $\Delta y_t = y_t - y_{t-1}$ is the first-difference of the time series.
- α is the constant term.
- βt is the trend term.
- γ is the coefficient on the lagged level of the time series.
- δ_i are the coefficients on the lagged first-differences of the time series.
- ε_t is the error term.
- p is the number of lagged first-differences included.

To induce stationarity, first and fourth order differencing was implemented along with year-on-year growth rates. First order differencing is a natural choice while fourth order differencing represents seasonal differencing seen as the data has a quarterly frequency. For each of the three transformed datasets, the augmented dickey fuller test was again run. First order differencing was most effective at inducing stationarity although modified final domestic demand and monthly industrial production were still deemed to be non-stationary by the ADF test. From visual inspection, the first order differenced modified final domestic demand doesn't have a trend, however the variance of the series does increase as time progresses, particularly during the Covid-19 pandemic. Perhaps this is why the null hypothesis of ADF fails to be rejected for this series, see figure 5. To explore

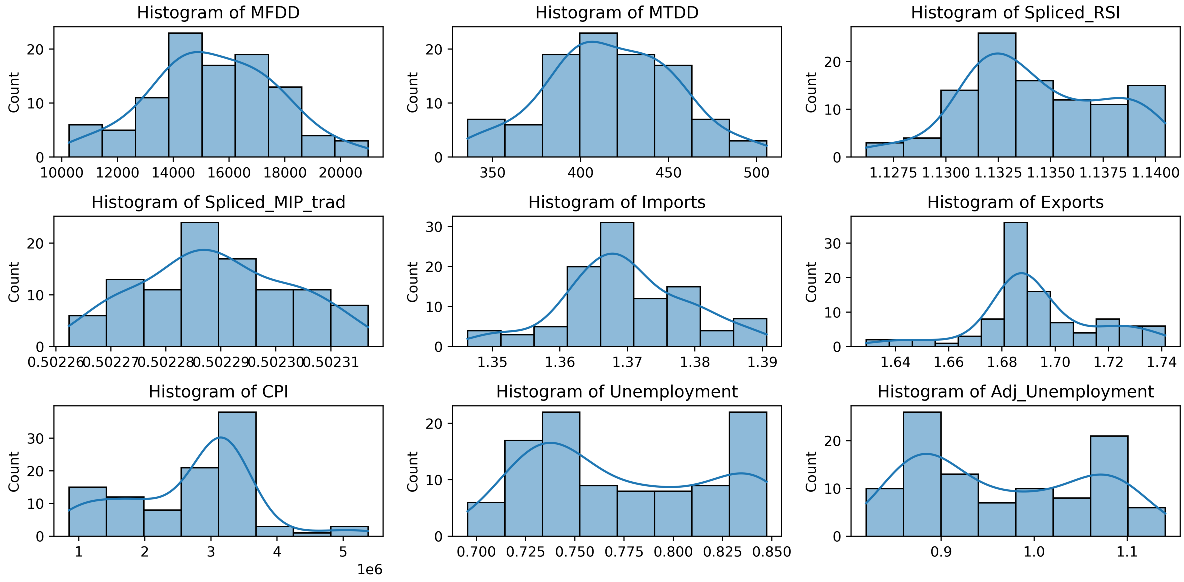


Figure 4: Histograms of Transformed Data

features which could be useful in modelling modified final domestic demand, which is the first research objective of this study, correlation heatmaps were produced for both the first and fourth order differenced data sets. See figure 6. The main diagonal is deep red reflecting the trivial fact that series are perfectly correlated with themselves. In the case of the first order differenced dataset, the correlations between modified final domestic demand and the potential features are mostly negative and close to zero. This is somewhat surprising as these series are capturing economic activity which would be classed as domestic demand. The correlations are much stronger for the fourth order differenced series, however perhaps this reflects the fact that fourth order differencing was less effective at inducing stationarity.

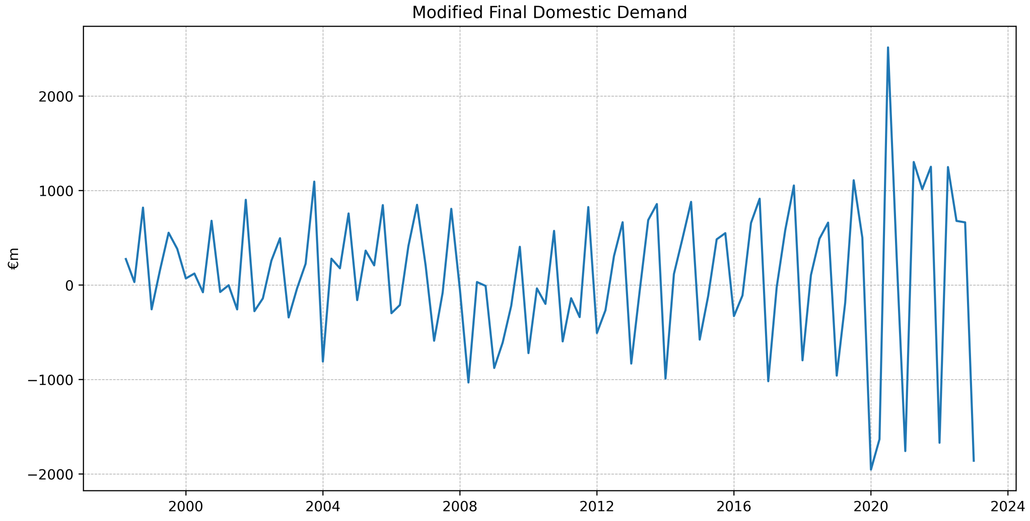


Figure 5: Modified Final Domestic Demand: First Order Difference

As well as performing a boxcox transformation and differencing, the data was also scaled to

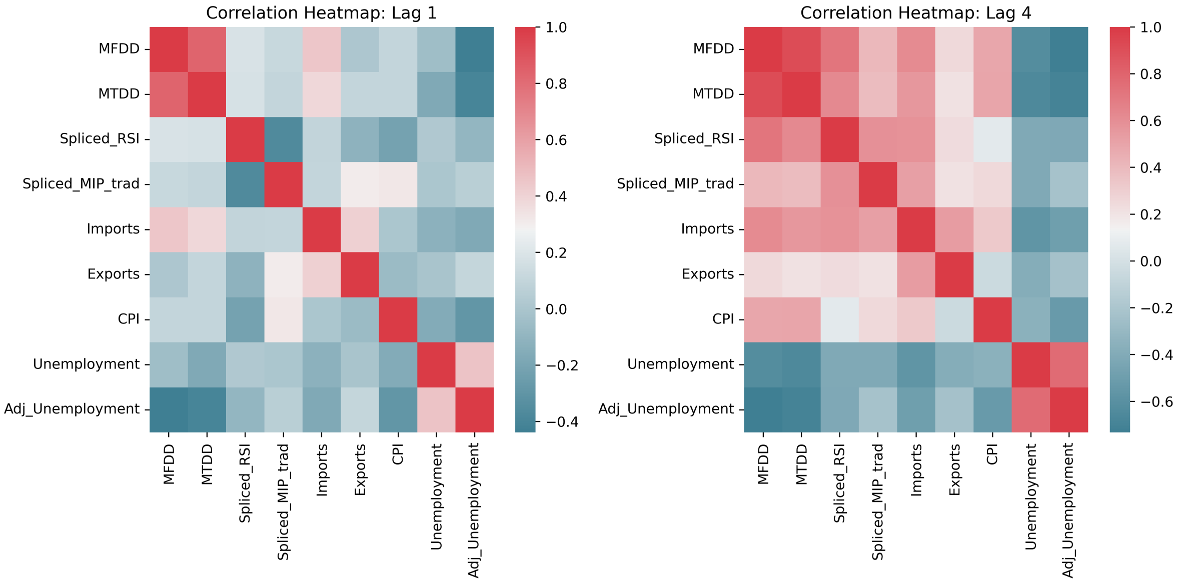


Figure 6: Correlation Heatmaps

have mean 0 and standard deviation of 1. This was performed as the series in the dataset were on very different scales. Some of the series were in billions of euro, others were indices with a base year set equal to 100 while the unemployment data was a percentage. Scaling can be an important processing step because regression models can be sensitive to the scale of the data. For example, in lasso and ridge regression, a penalty is added to the regression based on the size of the coefficients. Features with larger scales will naturally have larger coefficients. As a result these features would be penalised more, simply because of their scale rather than the underlying relationship between the target variable and the features. Scaling also makes interpretation of regression model parameters easier because the coefficients will be on the same scale. This makes it easier to see the relative importance of different features within the model. Finally, scaling can also help optimization techniques such as gradient descent to converge faster and more reliably.

5.3 Machine Learning

The first models run on the data were lasso and ridge regression. These models have the advantage of being relatively simply while lasso also performs feature selection. To fit the models, training and testing splits were created with 20 per cent of the data used for the test set. To preserve the temporal ordering of the data, shuffling was not used. Instead, time series cross validation was used with five splits of the data in order to find the optimal value of α which determines the slope penalty. The model did not perform well on the test set when used with the data which had a first order differencing transformation. See table 2. The lasso model performed much better with the fourth order differenced dataset. For this model, a mean absolute percentage error of 61 per cent was obtained on the test set. See figure 7 for a graph of the actual vs predicted values on the test set. Note that the inverse transformations were performed to produce this plot. The models were run again with the year-on-year growth rate transformation applied but these models performed even worse than the first order differencing based on the MAPE, see table 2.

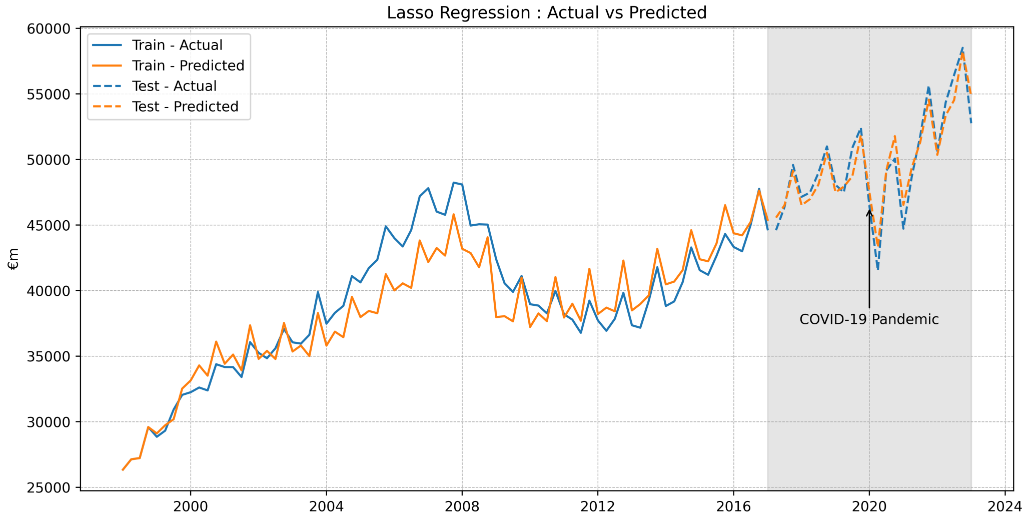


Figure 7: Lasso Regression

5.4 Time Series Econometrics

With the regression models having provided a baseline level of performance, the modelling switched to dedicated time series econometrics approaches, specifically Autoregressive Integrated Moving Average (ARIMA) models and Vector Auto Regression (VAR) models. The dataset with first order differencing was initially used. The data was also scaled to have mean 0 and standard deviation 1, though this is less important in the context of ARIMA models.

Before ARIMA could be applied, the serial correlation of modified final domestic demand was examined. See figure 8 for plots of the autocorrelation and partial autocorrelation functions. The ACF plot shows the correlations gradually reducing as lag length increases while the PACF has significant lags up to the fourth order. This suggests an ARMA(4, 0, 0) model might capture the dynamics of the data. As with the regression models, the data was again split into train and test sets with the test split set as 20 per cent of the data. However, given that the model is entirely based on lags of the dependent variable, walk forward validation was implemented. This is appropriate because in the context of flash GDP estimates, we're only ever interested in one step ahead forecasts. After the parameters of the model were estimated on the training set, a one step ahead prediction was made. Then the actual observation was appended to the training data and the model was re-estimated with the additional data point. This process continued until forecasts for each of the datapoints in the test set were made. Based on the MAPE for the test set, the model performed better than all the regression models apart from the best one, ie the lasso regression on the dataset with fourth order differencing.

This simple ARIMA model provides a baseline time series econometric model. To improve on this, an ARIMA model with additional explanatory variables was explored. To find the best features to use, an ARIMAX model was run for each possible combination of explanatory variables. The ARIMA order used was the same as for the baseline model, ie ARIMA(4, 0, 0). As with the baseline model, walk forward validation was used for evaluation. Based on the MAPE, the best combination of features was the retail sales index, exports and the Covid-19 adjusted unemployment

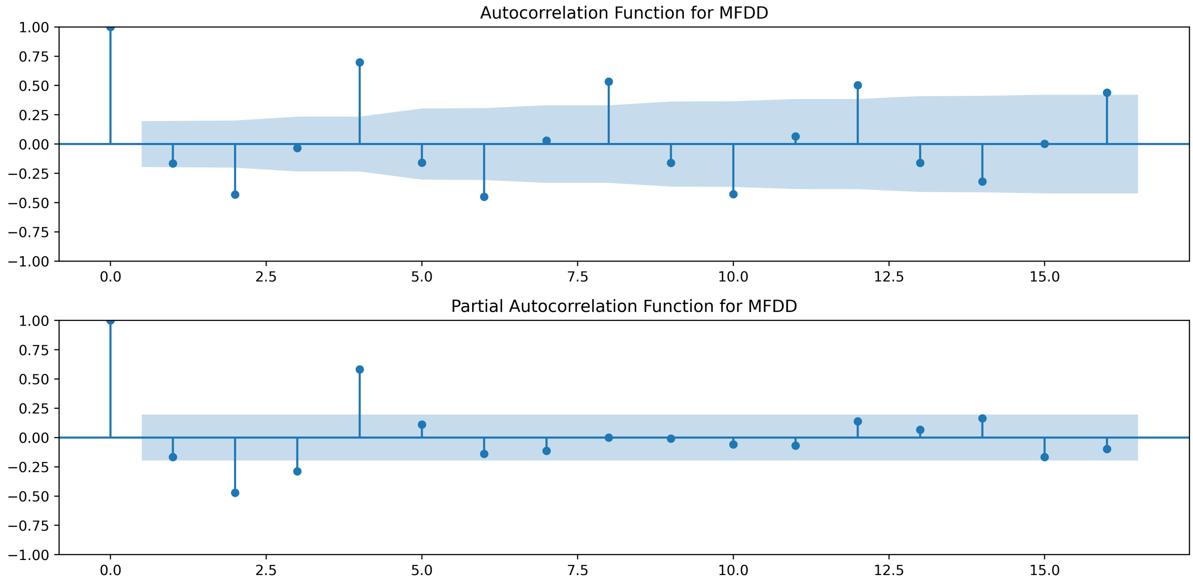


Figure 8: ACF & PACF Plots

rate. However, this combination of features did not have the lowest AIC or RMSE. See table 2. With good features identified, the next step was to find the optimal model order for an ARIMAX model with these features. To achieve this, all possible models with AR and MA lags up to and including order four were considered. Four seemed an appropriate value given that the data is quarterly. Based on the MAPE, an ARIMA(4, 0, 4) model was optimal. However, the AIC for the model was not the lowest, in part because the AIC favours parsimonious models. The MAPE on the test set for the optimal model was 30.5 per cent which is significantly better than either the baseline ARIMA model or the best regression model. See figure 9 for a plot of actual vs predicted.

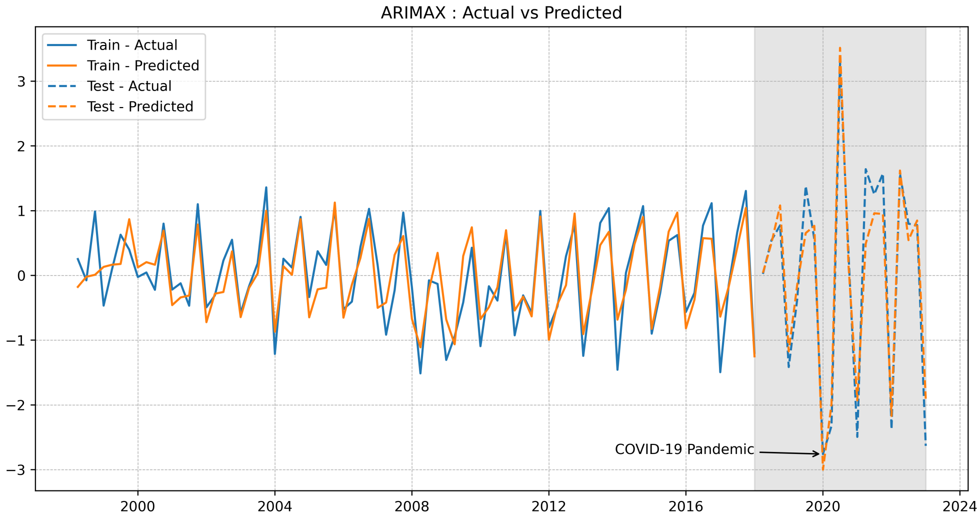


Figure 9: ARIMAX - Optimal Order & Exogenous Variables

Seen as the fourth order differenced data provided the best fitting regression model, the same analysis outlined above was conducted for the fourth order differenced data. However, the MAPE on the test set was much higher than for the first order differenced data. This is perhaps due to the data being closer to stationary in the case of the first order differenced data, as this is an important assumption in ARIMA modelling. Seaonal ARIMAX models were also explored with the first order differenced dataset. Hyperparameter tuning was implemented to find the optimal model order and combination of variables but the model underperformed all ARIMAX models based on MAPE on the test set. See table 2.

VAR models were also explored. To identify the features to include in the VAR model, all possible combinations were tried with the lag order of the VAR determined automatically. As with the ARIMA models, walk forward validation was utilised to give a better indication of how the models would perform in a flash GDP estimate context. Based on the MAPE on the test set, the optimal combination of variables to include with modified final domestic demand was imports, exports and the consumer price index. See table 2. Having determined the optimal features to include, the optimal order of the lags was determined by trying all lag orders up to and including 8. Again walk forward validation was utilised with the optimal lag order of 1 being determined by the MAPE on the test set. Interestingly, this optimal model has a lower MAPE than when the lag order was determined automatically in the context of trying different combinations of variables. This was not explored further as both VAR model results were much poorer than either the best regression model or the best ARIMA model. Finally, the same process was conducted on the dataset which contained fourth order differencing, however the results were similar.

5.5 Deep Learning

So far in this analysis, an ARIMAX model has provided the most accurate estimates of modified final domestic demand based on MAPE on the test set. To assess whether deep learning methods could improve on this, Long Short Term Memory models were explored which are considered state of the art models for time series forecasting.

While stationarity is not an assumption of LSTM models, the dataset utilised in the first instance was the dataset which had been transformed by first order differencing. The data was split into training, validation and test sets. The training data was set at 80 per cent of the data with the validation and test sets both containing 10 per cent of the data.

A function was created for performing hyperparameter tuning which accepts three arguments, the data, an argument which determines the length of the input sequences and a directory name to store the results of the hyperparameter tuning. A sequential model was developed utilising the keras library. The first layer of the LSTM model is added with a variable number of units ranging from 32 to 512 in steps of 32. While this is fundamentally arbitrary, there can be computational efficiencies to be had in choosing units which are powers of 2. Up to six additional layers are tried in the tuning process. As with the first layer, the number of units ranges from 32 to 512 in steps of 32. A final layer is added with the same range of units as before but this layer does not return sequences. Regularization is performed with a dropout layer to help avoid overfitting. The dropout layer randomly sets a fraction of input units to 0 at each update during training. The values searched through the hyperparameter tuning range from 0 to 0.5 in increments of 0.1. Two dense layers are added at the end with the intermediate layer ranging from 10 to 100 units in steps of 10 with a rectified linear unit activation function. The last dense layer has a single output, representing the estimate we're interested in. The model is compiled with the Adam optimizer with a tuneable learning rate. The chosen loss function is MAPE.

To perform the hyperparameter tuning, random search is used with the maximum number of trials set equal to 90 with 3 executions per trial. The maximum number of trials determines the number of different sets of hyperparameters the tuner should try. As the name suggests, random search randomly searches the hyperparameter space, ie it is not trying all possible combinations. Executions per trial determines the number of times the model will be trained from scratch for a given set of hyperparameters with the results averaged. This is done to account for the random variation in training the model. This random variation comes from a number of places including initialisation of model weights. The choice of random search is necessary to keep training times tolerable. While the hyperparameter tuning was run on an NVIDIA GeForce RTX 3050 Ti Laptop GPU, it still took a considerable amount of time to run. The tuning for each dataset, which also included the number of lags as a hyperparameter, took approximately 160 minutes to run. When the hyperparameter tuning is complete, the function takes the best model and performs walk forward validation on the test set.

This function is run for values of the input sequences ranging from 2 to 12 in increments of 2. This is determining how many previous values are considered when making a prediction. It's analogous to how many lags to include in a VAR model. The best model identified had an input sequence length of 4, ie four previous values are considered when predicting the next value. The optimal model had 4 LSTM layers with 2.4 million parameters. The other optimal hyperparameters were a learning rate of 0.01 and a dropout rate of 0. The optimal values for the units in each layer can be see in a graph of the model architecture in figure 12. The MAPE on the test set was 90.9 per cent which is considerably less accurate than the best ARIMA model identified. See figure 10 for a graph of the model loss on the train and validation sets and figure 11 for actual vs predicted values.

The same analysis was performed on the dataset with fourth order differencing. Here the optimal number of lags was again four while the optimal model again had four LSTM layers with 2 million parameters. See figure 13 for a graph of the model architecture. The other optimal hyperparameters were a learning rate of 0.01 and a dropout rate of 0.4. The MAPE on the test set was 91.2 per cent which is only marginally worse than the model run on the first order differenced dataset. See figure 30 for a graph of the model loss on the train and validation sets and figure 31 for actual vs predicted values.

The same analysis was again performed but this time on the dataset where the transformation was to calculate y-o-y growth rates. Here the optimal number of lags was 12 while the optimal model again had four LSTM layers with just over 1.1 million parameters. See figure 15 for a graph of the model architecture. The other optimal hyperparameters were a learning rate of 0.01 and a dropout rate of 0.3. The MAPE on the test set was 103.2 per cent which is less accurate than the other LSTM models trained. See figure 32 for a graph of the model loss on the train and validation sets and figure 33 for actual vs predicted values.

LSTMs are robust to the statistical properties of the data, for example they can directly handle non-stationary or seasonal data. For this reason, an LSTM model was again trained but this time the dataset was simply scaled to have mean 0 and standard deviation 1. No other transformation was applied to the data. In addition, unlike the models above, aggregate imports were replaced with detailed import data intended to capture elements of gross capital formation, specifically machinery

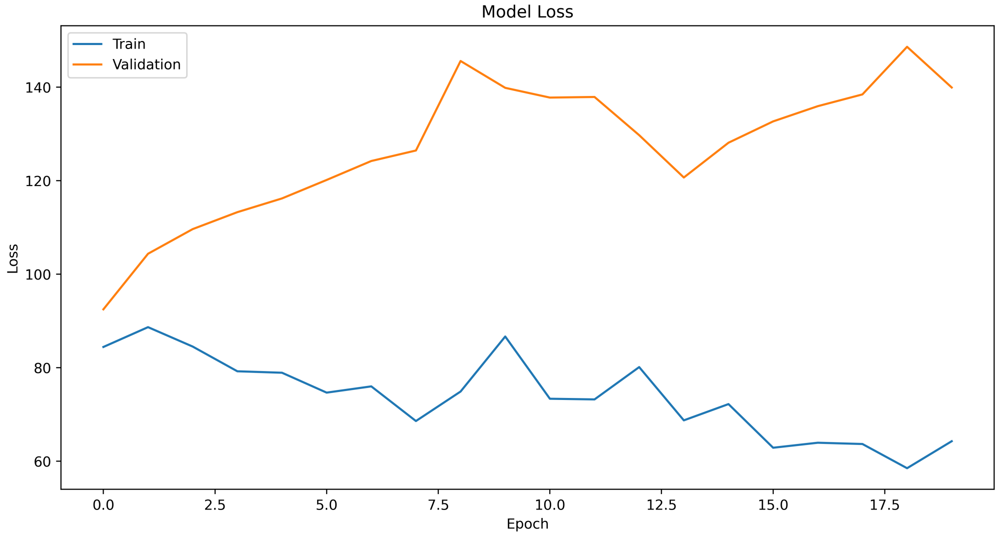


Figure 10: LSTM Loss

& equipment and building & construction. The optimal number of lags was 8 while the optimal model had three LSTM layers with just over 1.2 million parameters. See figure 11 for a graph of the model architecture. The other optimal hyperparameters were a learning rate of 0.001 and a dropout rate of 0.2. The MAPE on the test set was 97.8 per cent which is similar to the other LSTM models trained, suggesting that the accuracy is not sensitive to the particular transformations performed on the data. See figure 34 for a graph of the model loss on the train and validation sets and figure 35 for actual vs predicted values. The results are disappointing with the predicted values not so different from a simple naive forecast, where the prediction is set as the mean of the series.

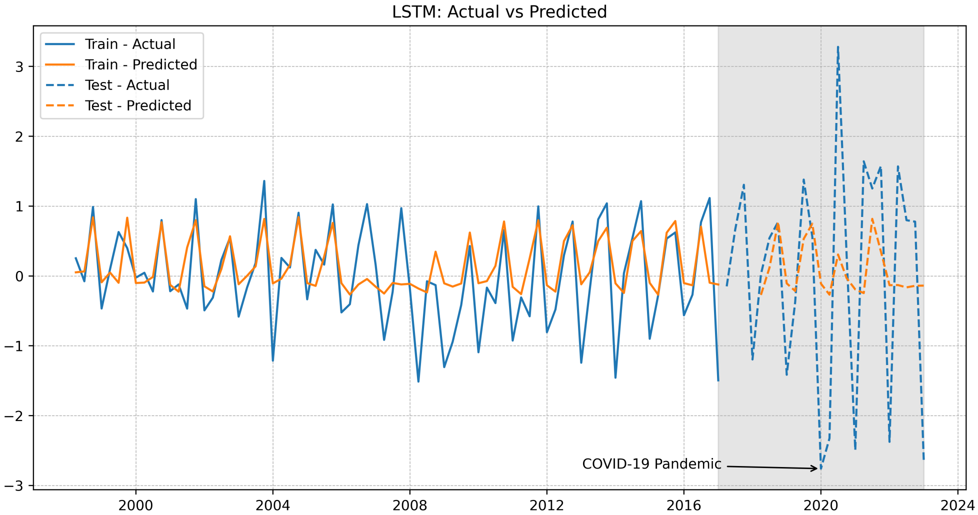


Figure 11: LSTM Actual vs Predicted

5.6 Table of Results

The results of the modelling are summarised in table 2 which displays the type of model, some cursory information on the chosen hyperparameters, the transformation applied to the data, the features included in the model and a couple of accuracy metrics with a column specifying whether the metric relates to the training or the test set.

ARIMA models were most accurate at predicting on a test set based on MAPE, with four of the top 5 models being ARIMA based. The most accurate model was estimated on the dataset with first order differencing. Note that all the datasets have a boxcox transformation and standard scaling applied so it's not referenced in the table. The most accurate ARIMA model had the exogenous variables retail sales, exports and adjusted unemployment included in the model. The best Lasso model had the third highest accuracy overall. This result is more impressive when considering that walk forward validation with 1 step ahead forecasts was not implemented in the case of the Lasso and Ridge regression models.

There were two LSTM models in the top 10 most accurate models. However the LSTM which was trained on the dataset with fourth order differencing, is not much different than a naive forecast where the predicted value is simply the mean of the series, see figure 31. While the LSTM model trained on the first order differenced dataset does a much better job capturing the dynamics of the series, it only has a marginally lower MAPE. See figure 11.

Table 2 shows that the top 10 most accurate models includes almost all of the different model types tried with ARIMA, Lasso, LSTM, and VAR represented. The results also indicate that no model type dominates all others with the accuracy depending on the particular transformations applied to the dataset. For example, while the most accurate model is an ARIMA model, the best Lasso, LSTM and VAR models are more accurate than the worst ARIMA model.

Table 2: Table of Results for Various Models and Datasets

Model Type	Hyperparameters	Transformation	Data	Variables	RMSE	MAPE
ARIMA	(4, 0, 4)	First Order Lag	Test	RSI, X, Adj_U	0.44	30.46
ARIMA	(4, 0, 0)	First Order Lag	Test	RSI, X, Adj_U	0.52	33.84
Lasso	alpha=0.03	Fourth Order Lag	Test	RSI, MIP, M, X, U, Adj_U	0.62	61
ARIMA	(0, 0, 2)	Fourth Order Lag	Test	RSI, MIP, M, X, Adj_U	0.43	62.38
ARIMA	(1, 0, 0)	Fourth Order Lag	Test	RSI, MIP, M, X, Adj_U	0.48	72.4
ARIMA	(4, 0, 0)	First Order Lag	Test	None	1.22	83.23
LSTM	Lag Order: 4	First Order Lag	Test	RSI, MIP, M, X, CPI, Adj_U	NA	90.91
LSTM	Lag Order: 4	Fourth Order Lag	Test	RSI, MIP, M, X, CPI, Adj_U	NA	91.20
VAR	Lag Order: 1	First Order Lag	Test	M, X, CPI	1.62	93.54
VAR	Lag Order: 6	Fourth Order Lag	Test	RSI, X	1.28	93.76
SARIMA	(0, 0, 0)(1, 0, 2, 4)	First Order Lag	Test	RSI, MIP, M, X, CPI, Adj_U	0.51	95.79
LSTM	Lag Order: 8	None	Test	RSI, MIP, X, CPI, Adj_U, Detailed M	NA	97.78
ARIMA	(1, 0, 0)	Fourth Order Lag	Test	None	1.15	102.92
LSTM	Lag Order: 12	Y-o-Y % change	Test	RSI, MIP, M, X, CPI, Adj_U	NA	103.15
Lasso	alpha=0.065	First Order Lag	Train	RSI, MIP, M, X, U, Adj_U	0.63	115.94
Ridge	alpha=28.94	First Order Lag	Train	RSI, MIP, M, X, U, Adj_U	0.63	117.75
Lasso	alpha=0.065	First Order Lag	Test	RSI, MIP, M, X, U, Adj_U	1.59	125.04
Ridge	alpha=28.94	First Order Lag	Test	RSI, MIP, M, X, U, Adj_U	1.57	134.8
Lasso	alpha=0.00	Y-o-Y % change	Test	RSI, MIP, M, X, U, Adj_U	1.84	243.55
Ridge	alpha=0.03	Y-o-Y % change	Test	RSI, MIP, M, X, U, Adj_U	2.14	263.02
Lasso	alpha=0.00	Y-o-Y % change	Train	RSI, MIP, M, X, U, Adj_U	0.39	296.9
Ridge	alpha=0.03	Y-o-Y % change	Train	RSI, MIP, M, X, U, Adj_U	0.39	296.9
Ridge	alpha=0.055	None	Train	RSI, MIP, X, U, Adj_U, Detailed M	0.2	339.84
Lasso	alpha=0.00	None	Train	RSI, MIP, X, U, Adj_U, Detailed M	0.2	343.9
Ridge	alpha=0.02	Fourth Order Lag	Test	RSI, MIP, M, X, U, Adj_U	5.76	357.04
Ridge	alpha=0.055	None	Test	RSI, MIP, X, U, Adj_U, Detailed M	0.43	801.63
Ridge	alpha=0.02	Fourth Order Lag	Train	RSI, MIP, M, X, U, Adj_U	0.39	817.5
Lasso	alpha=0.03	Fourth Order Lag	Train	RSI, MIP, M, X, U, Adj_U	0.44	999
Lasso	alpha=0.00	None	Test	RSI, MIP, X, U, Adj_U, Detailed Imports	0.49	1036.32

6 Discussion

This study sought to compare traditional econometric methods to deep learning approaches in the context of flash GDP estimates. The hope was that deep learning methods could improve the accuracy of flash estimates. However, that doesn't appear to be the case, at least as regards modified final domestic demand. Based on comparing mean absolute percentage errors between predicted values and actual values on a test set, ARIMA models with explanatory variables were considerably more accurate. In the literature review, we saw mixed results as to whether ARIMA models or deep learning methods perform best for time series forecasting. For example, Yamak et al. [2019] suggests the superiority of ARIMA models over deep learning models while Siami-Namini et al. [2018] provides evidence for the opposite conclusion. In the context of one step ahead forecasts of relatively short economic time series, this study lends weight to the view that ARIMA models are superior. It also helps justify the continued dominance of ARIMA type models in fields such as economics and official statistics. While deep learning models were not in the purview of the experts interviewed, based on the finding of this study, there's no great reason RNN type models should be in their toolkit. It's likely that were deep learning models significantly better at forecasting economic time series, there would be greater diffusion of these techniques outside of the computing field.

It remains possible of course that LSTM models could provide superior forecasting on the task considered here if more resources were devoted to hyperparameter tuning. However, this doesn't seem likely, as the model accuracy seemed to plateau and was not sensitive to additional tuning. Of course that doesn't preclude the possibility that there may be non-linearities here and that significantly improved accuracy could be achieved (without over-fitting) if only a large enough space of hyperparameters were searched. However, as discussed in the literature review, this is in part the weakness of deep learning approaches [Hewamalage et al., 2021]; they're not ready to use 'out of the box'. There's very little in the way of theory for choosing a particular network architecture, instead considerable computing resources are required to perform hyperparameter tuning along with some form of cross validation. This is in stark contrast to ARIMA models where there exists a simple theoretical framework for choosing the model structure. If hyperparameter tuning is to be performed, the space of hyperparameters is comparatively very small and little is needed in the way of computing resources.

While this study can't recommend deep learning models in order to improve flash GDP estimates in Ireland, there are lessons to be learned. In the primary research for this study, interviews were conducted with experts in the Central Statistics Office who work or have worked in the area of flash GDP estimation. That research identified that in some aspects of the flash GDP estimation process, early company data is available to inform estimates. This is the ideal situation. The purpose of flash GDP estimation is to produce early estimates of GDP compared to the non-flash estimate produced 60 days after the quarter ends. In so far as the data which feeds into the T+60 estimate is available at T+30 for the flash estimate, no models are required at all. However, this early company data needs to be complemented by ARIMA models and judgement. The primary research has revealed that only standard ARIMA models are being used, ie no additional explanatory variables are being incorporated into the modelling process. This is perhaps not too surprising as Eurostat [2016] provides only passing reference to the inclusion of exogenous explantory variables within ARIMA models. This study has shown that in the case of modified final domestic demand, there are large improvements in forecast accuracy to be had from incorporating additional explanatory variables in ARIMAX models.

There are a number of limitations to this study. Firstly, only a single series, modified final domestic demand, was considered. In reality, one-step ahead forecasts in the context of flash GDP estimation would include forecasting many series. As highlighted in Teräsvirta et al. [2003], the best model will depend in part on the particular time series of interest among other factors. Perhaps in a realistic flash GDP context where early estimates were being produced for many more series, deep learning models would perform better. Future research could explore this. In addition, this study made use of only publicly available datasets. In practice, national statistical institutes would have access to many more sources of data than were available for this study. As highlighted in Yamak et al. [2019], deep learning approaches perform better with a larger volume of data. Perhaps deep learning approaches would perform better in an actual flash GDP estimation context where more data would be available.

7 Conclusion

This study sought to investigate whether deep learning methods could be used to improve the accuracy of flash GDP estimates. Specifically, it examined whether LSTM models could produce more accurate one step ahead forecasts of modified final domestic demand when compared to the methods currently used in national statistical institutes. To this end, a literature review was conducted which explored the methods currently in use in the national statistical institutes within the European Union. The literature comparing the relative accuracy of different approaches to time series forecasting was also explored. It was shown in the literature review that the results are mixed as to whether traditional econometric models such as ARIMA are more or less accurate than modern deep learning approaches.

Regarding the data used for this study, the primary data collection included in-depth interviews with experts in the area of flash GDP estimation in the Central Statistics Office (CSO). This information provided detailed insight into the methods and approaches currently taken by the CSO to produce flash estimates. The secondary data sources consisted of a number of high frequency datasets which could be useful for producing early estimates of quarterly modified final domestic demand within 30 days of the end of a quarter.

Exploratory data analysis was performed which consisted of visual inspections of the data and formal statistical tests such as Augmented Dickey Fuller to establish the relevant statistical properties of the dataset. This analysis informed the choice of transformations of the data including a box-cox transformation to assist in variance stabilisation, transformations to induce stationarity and scaling of the data so each series had a mean of 0 and a standard deviation of 1. These were all important steps to prepare the data for modelling. To assess which variables would be most useful for predicting modified final domestic demand, correlation heatmaps and pairwise plots were examined. However, this analysis was simply suggestive as more formal feature selection was performed in the modelling phase.

A number of models were explored to produce flash estimates of modified final domestic demand. The relatively simple regression based models LASSO and RIDGE were used with cross validation to establish a baseline of predictive accuracy. To improve on this, dedicated time series econometrics methods were implemented including ARIMA, ARIMAX and VAR models. Hyperparameter tuning was utilised to find the optimal combinations of features and model structure. To improve forecast accuracy and also to provide a more realistic comparison to the one step ahead forecasts used in a flash GDP context, walk forward validation was utilised.

The central question of this study is whether deep learning approaches, specifically LSTM models, could be used to improve accuracy in the context of flash GDP estimation. To explore this, a number of LSTM models were trained. As with the models referenced above, the modelling was done on various datasets which differed based on the particular transformations applied. For each of these models, extensive hyperparameter tuning was implemented which explored a hyperparameter space which consisted of the particular architecture of the LSTM, the number of lags to include, the dropout rate and the learning rate to be used for the Adam optimiser. Once an optimal model was identified, walk forward validation was implemented on a test set as in the case of the ARIMA modelling. The conclusions of this study are that at least for the case of one step ahead forecasts of modified final domestic demand, ARIMAX models provide more accurate forecasts than

LSTM models as measured by the MAPE. In fact, most of the LSTM models offer little more than a naive forecast where one simply predicts the mean of the series. Although Teräsvirta et al. [2003] emphasises that the best model can depend on the particular dataset of interest.

The choice of producing modified final domestic demand was motivated in part by limiting the breadth of this study. A more realistic approach, as discussed in the problem clarification, would involve separately producing early estimates of the components of the modified final domestic demand and summing these to produce an estimate for the aggregate. Future research could explore whether the superiority of ARIMA models would also apply in this more realistic case. The lower accuracy of deep learning methods compared to traditional econometric methods could possibly be due to the relatively small datasets available in the context of economic time series. This possibility was suggested in Yamak et al. [2019] as discussed in the literature review. However, further research is required in this area to understand when and why deep learning methods are useful for time series forecasting.

Finally, the success of generative AI models such as ChatGPT has brought much attention to a particular kind of deep learning architecture called a transformer. Generative AI chatbots need architectures which can handle the sequential nature of language. As time series data is an ordered sequence of values in time, future research could explore whether transformers could be used to improve the accuracy of early estimates in a flash GDP context.

While deep learning approaches can't be recommended to the CSO as a means to improve flash GDP estimation based on this study, there are still important findings resulting from this work. As discovered in the primary research, where the CSO incorporates modelling in its estimates, simple ARIMA models with no additional explanatory variables are utilised. This study has shown that incorporating features in an ARIMAX model can greatly improve the accuracy of estimates.

A LSTM Model Architecture

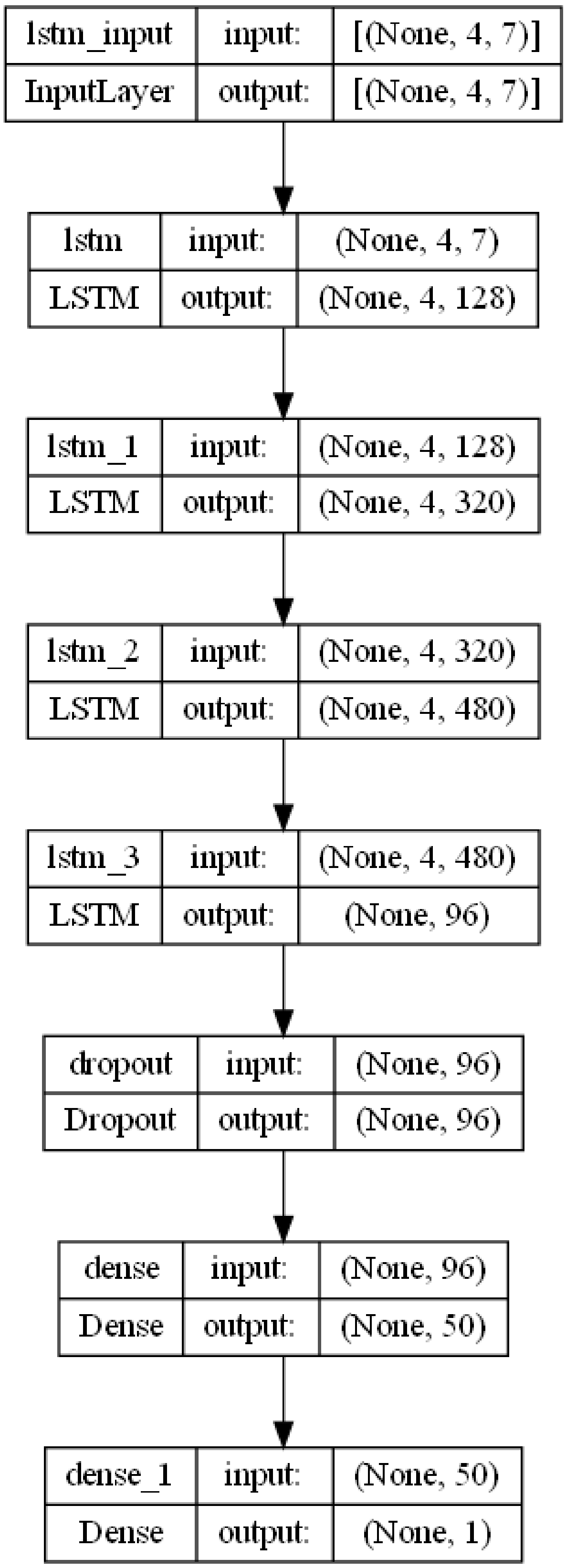

lstm_input | input: | [(None, 4, 7)]
InputLayer | output: | [(None, 4, 7)]
lstm | input: | (None, 4, 7)
LSTM | output: | (None, 4, 128)
lstm_1 | input: | (None, 4, 128)
LSTM | output: | (None, 4, 320)
lstm_2 | input: | (None, 4, 320)
LSTM | output: | (None, 4, 480)
lstm_3 | input: | (None, 4, 480)
LSTM | output: | (None, 96)
dropout | input: | (None, 96)
Dropout | output: | (None, 96)
dense | input: | (None, 96)
Dense | output: | (None, 50)
dense_1 | input: | (None, 50)
Dense | output: | (None, 1)

Figure 12: LSTM Architecuture: Lag 1

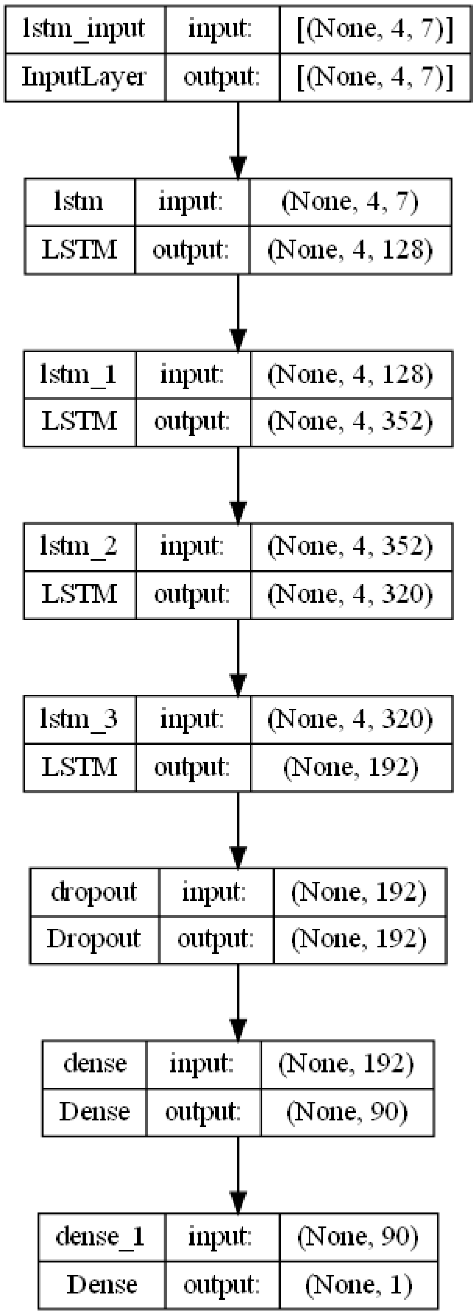

lstm_input | input: | [(None, 4, 7)]
InputLayer | output: | [(None, 4, 7)]
lstm | input: | (None, 4, 7)
LSTM | output: | (None, 4, 128)
lstm_1 | input: | (None, 4, 128)
LSTM | output: | (None, 4, 352)
lstm_2 | input: | (None, 4, 352)
LSTM | output: | (None, 4, 320)
lstm_3 | input: | (None, 4, 320)
LSTM | output: | (None, 192)
dropout | input: | (None, 192)
Dropout | output: | (None, 192)
dense | input: | (None, 192)
Dense | output: | (None, 90)
dense_1 | input: | (None, 90)
Dense | output: | (None, 1)

Figure 13: LSTM Architecture: Lag 4

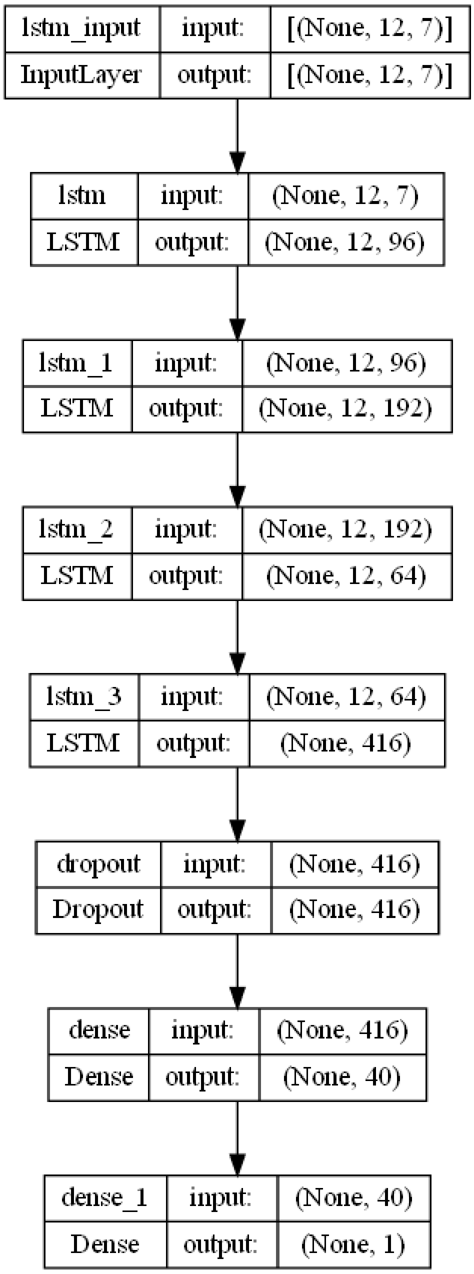
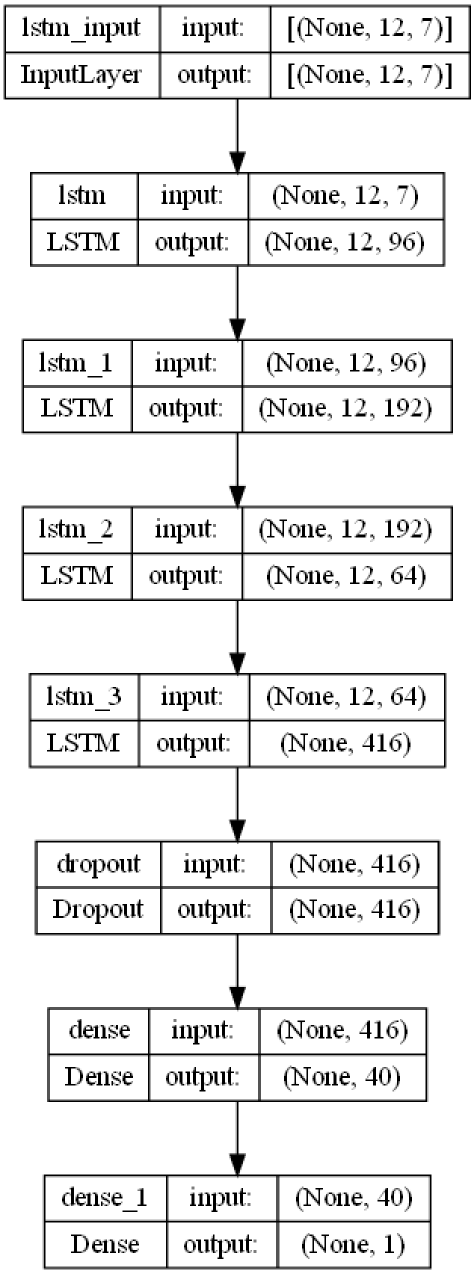

lstm_input | input: | [(None, 12, 7)]
InputLayer | output: | [(None, 12, 7)]
lstm | input: | (None, 12, 7)
LSTM | output: | (None, 12, 96)
lstm_1 | input: | (None, 12, 96)
LSTM | output: | (None, 12, 192)
lstm_2 | input: | (None, 12, 192)
LSTM | output: | (None, 12, 64)
lstm_3 | input: | (None, 12, 64)
LSTM | output: | (None, 416)
dropout | input: | (None, 416)
Dropout | output: | (None, 416)
dense | input: | (None, 416)
Dense | output: | (None, 40)
dense_1 | input: | (None, 40)
Dense | output: | (None, 1)

Figure 14: LSTM Architecture: y-o-y



lstm_input | input: | [(None, 12, 7)]
InputLayer | output: | [(None, 12, 7)]

lstm | input: | (None, 12, 7)
LSTM | output: | (None, 12, 96)

lstm_1 | input: | (None, 12, 96)
LSTM | output: | (None, 12, 192)

lstm_2 | input: | (None, 12, 192)
LSTM | output: | (None, 12, 64)

lstm_3 | input: | (None, 12, 64)
LSTM | output: | (None, 416)

dropout | input: | (None, 416)
Dropout | output: | (None, 416)

dense | input: | (None, 416)
Dense | output: | (None, 40)

dense_1 | input: | (None, 40)
Dense | output: | (None, 1)

Figure 15: LSTM Architecture: Scaling Only

B ACF & PACF Plots

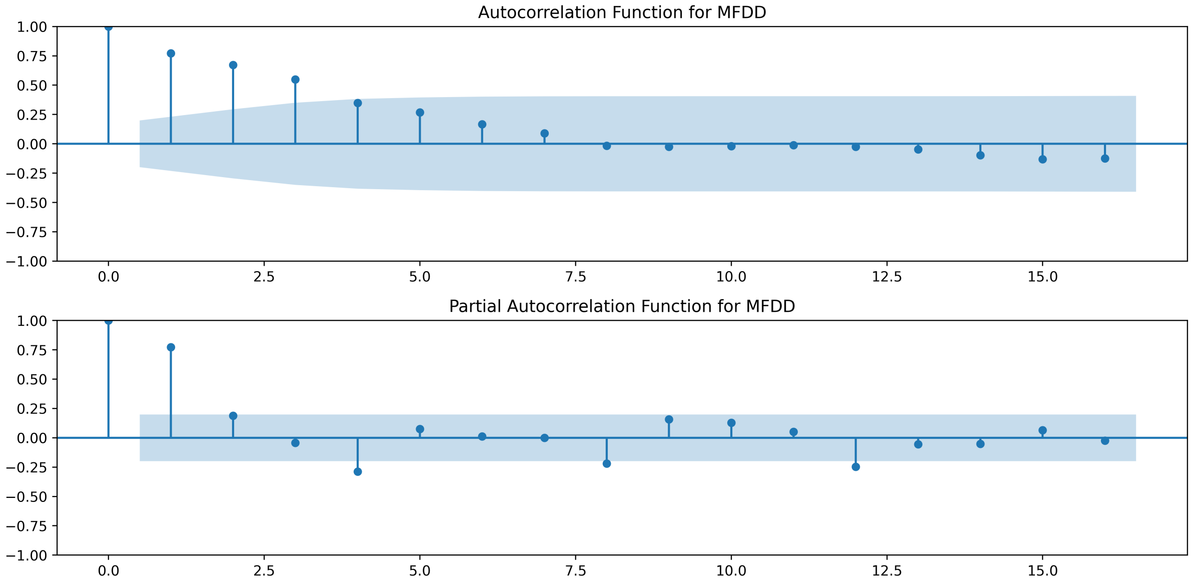


Figure 16: LSTM Loss

C Model Graphs

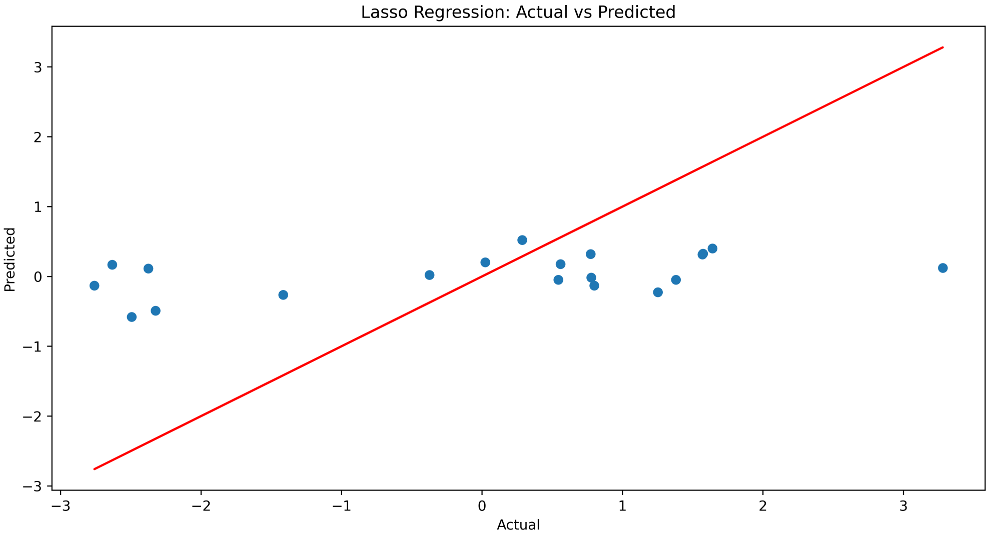


Figure 17: LASSO - Actual vs Predicted - Lag 1

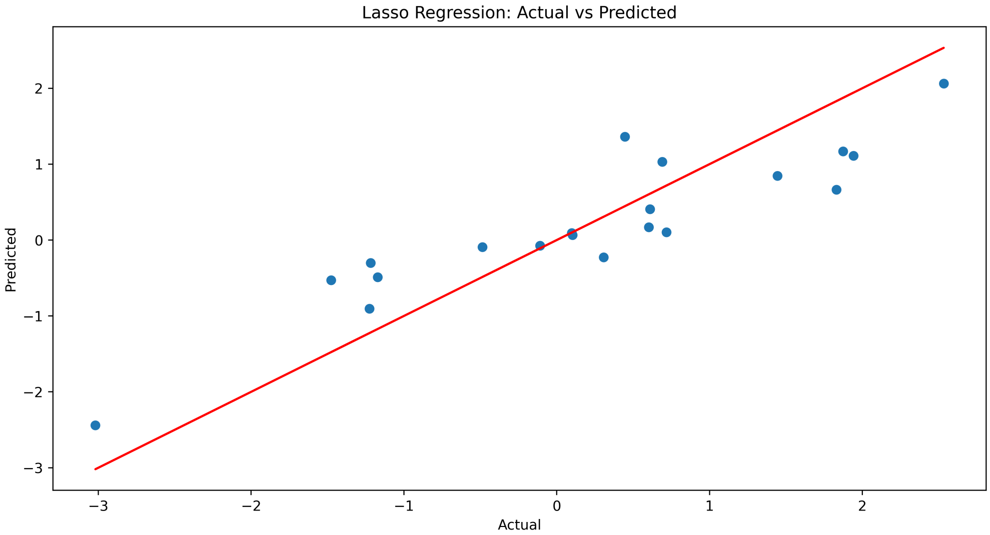


Figure 18: LASSO - Actual vs Predicted - Lag 4

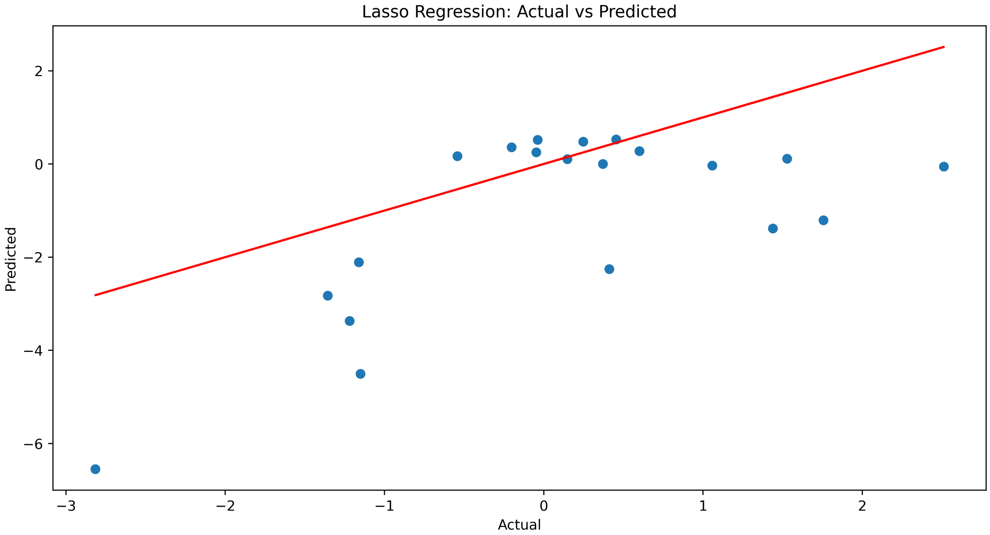


Figure 19: LASSO - Actual vs Predicted - y-o-y

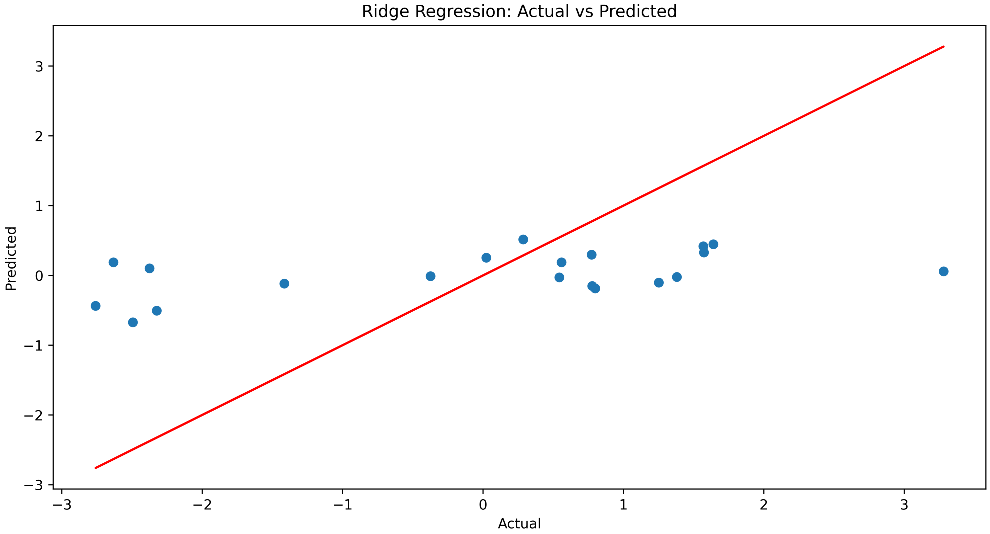


Figure 20: Ridge - Actual vs Predicted - Lag 1

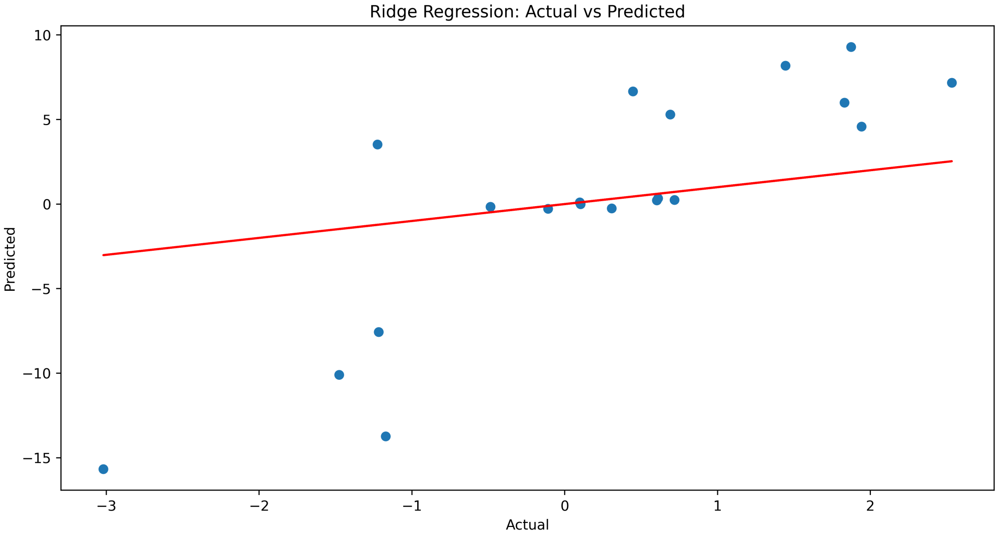


Figure 21: Ridge - Actual vs Predicted - Lag 4

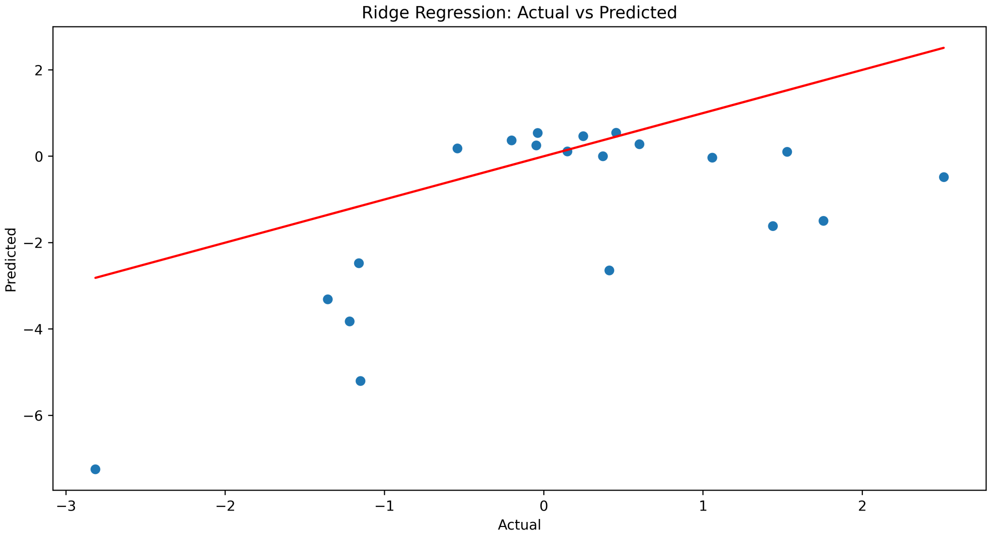


Figure 22: Ridge - Actual vs Predicted - y-o-y

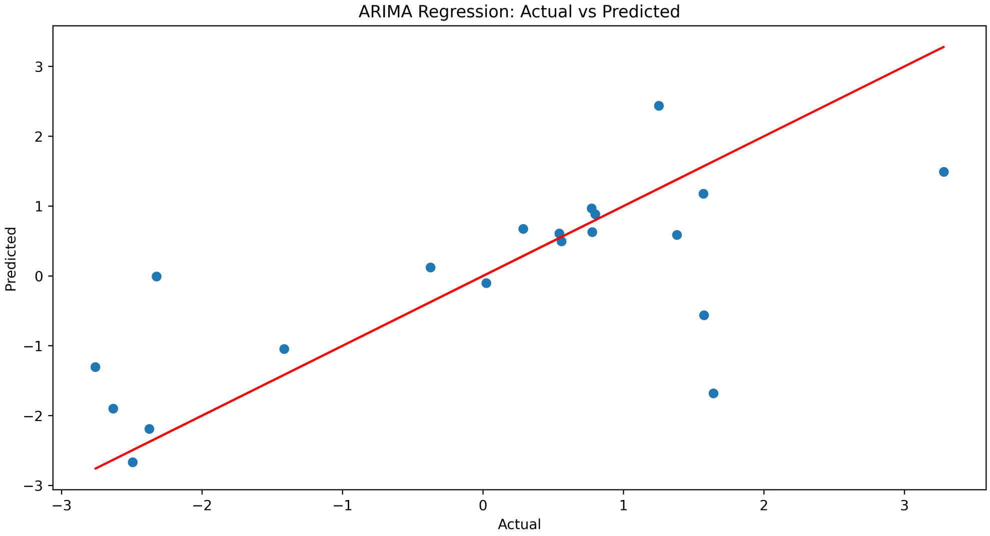


Figure 23: ARIMA - Actual vs Predicted - Lag 1

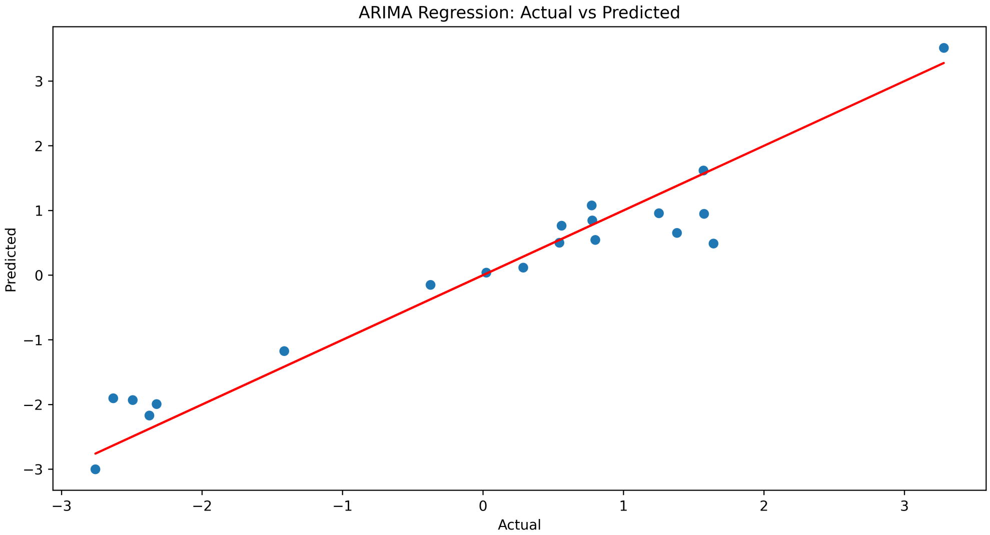


Figure 24: Optimal ARIMA - Actual vs Predicted - Lag 1

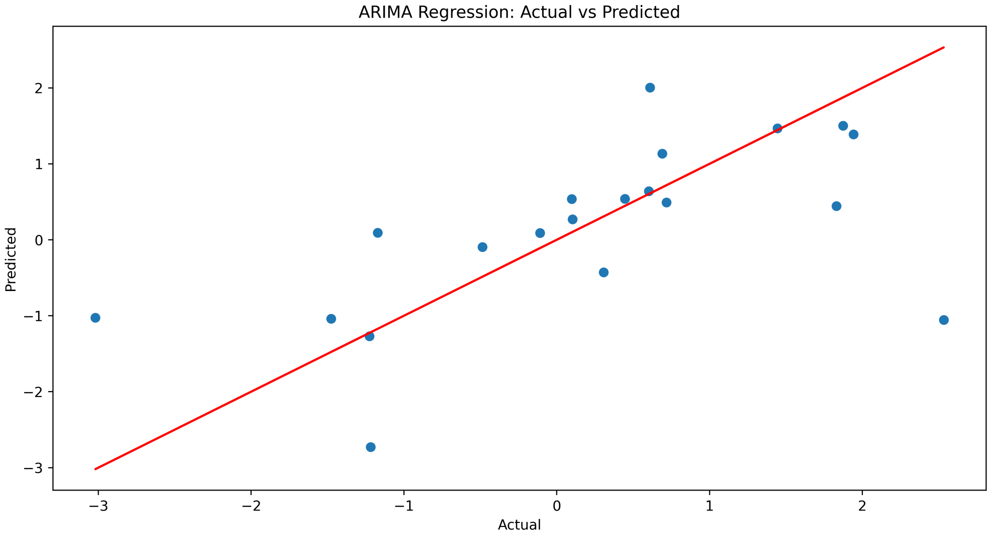


Figure 25: ARIMA - Actual vs Predicted - Lag 4

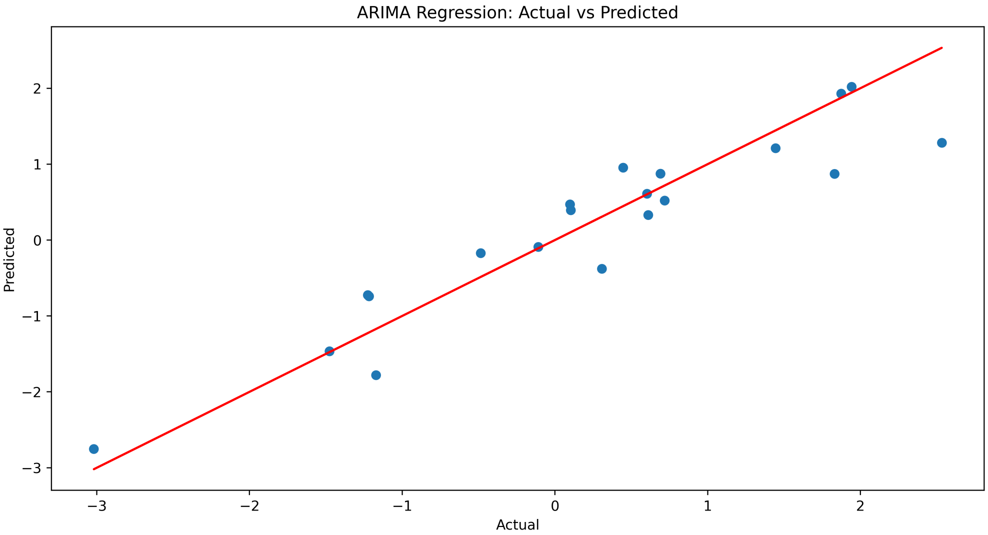


Figure 26: Optimal Exogenous Variables ARIMA - Actual vs Predicted - Lag 4

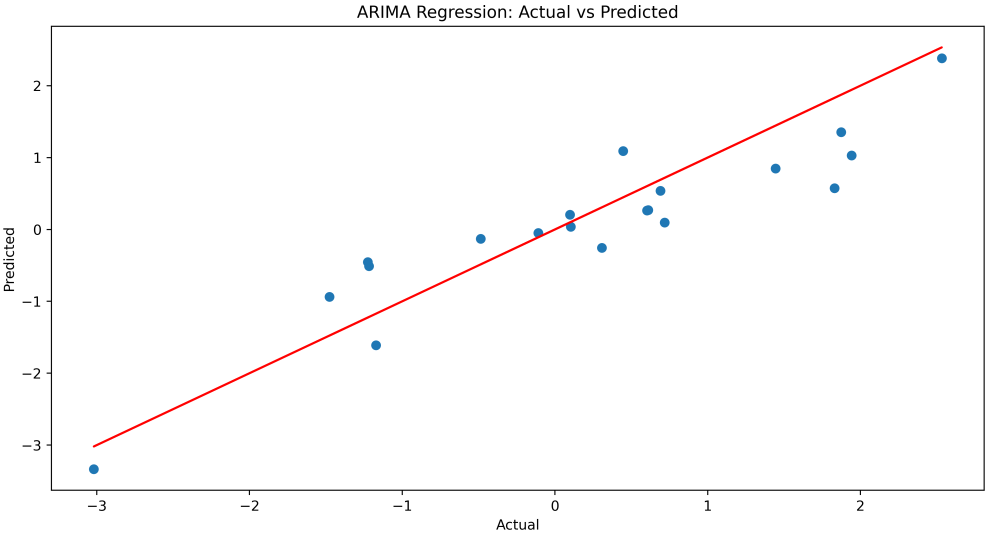


Figure 27: Optimal ARIMA - Actual vs Predicted - Lag 4

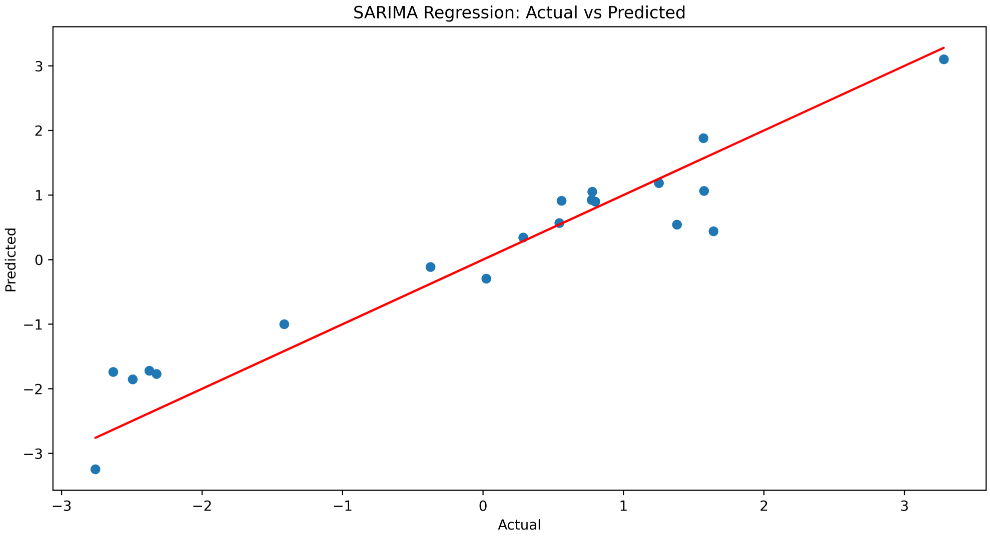


Figure 28: Optimal SARIMAX - Actual vs Predicted - Lag 1

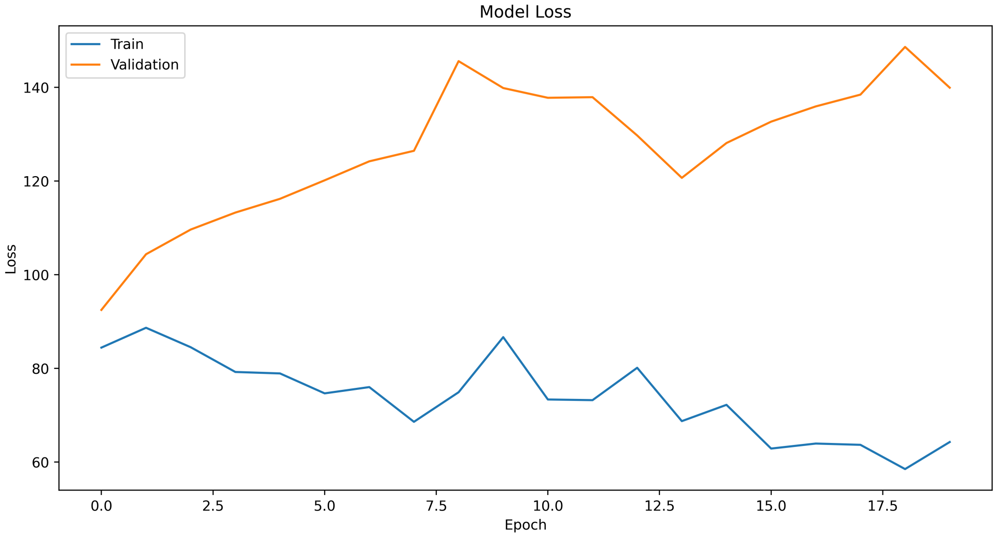


Figure 29: LSTM Model Loss: Lag 1

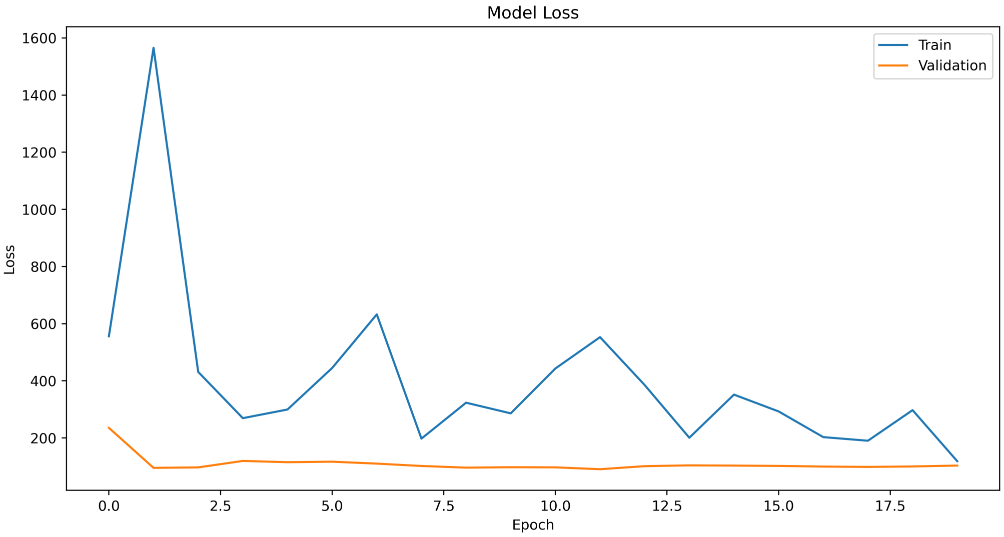


Figure 30: LSTM Model Loss: Lag 4

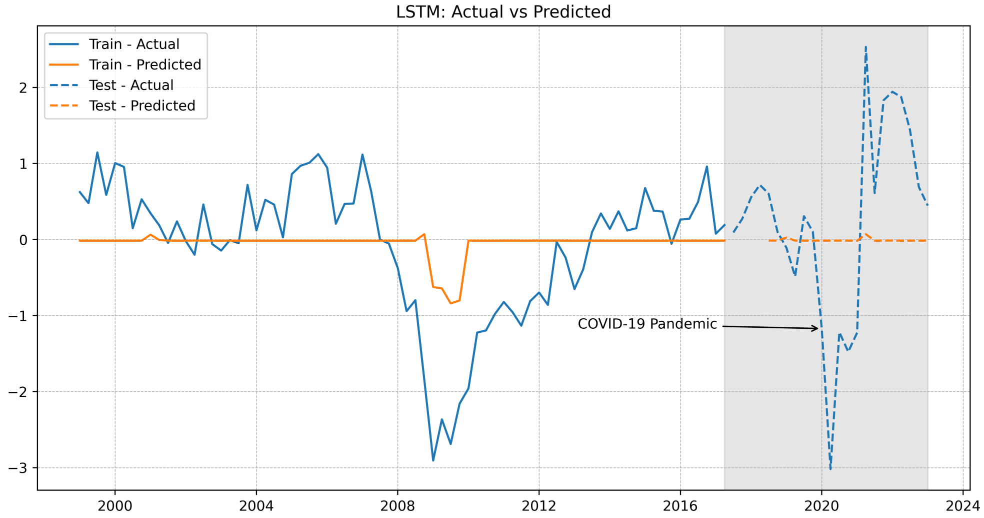


Figure 31: LSTM Lag 4: Actual vs Predicted

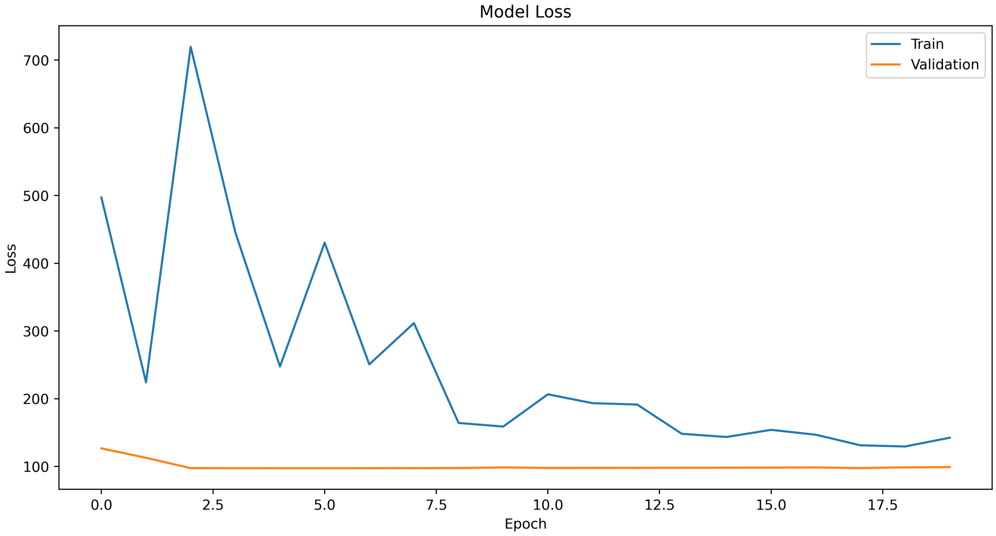


Figure 32: LSTM Model Loss: Y-o-Y

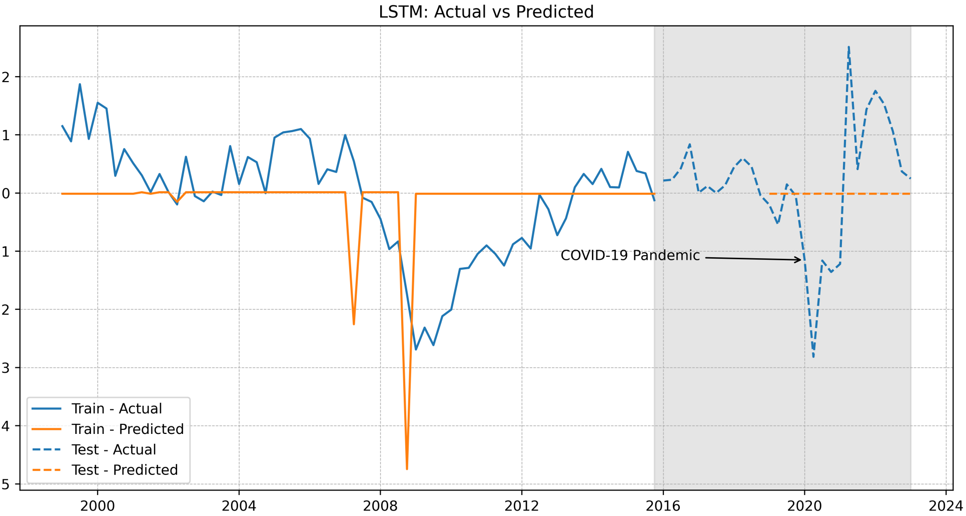


Figure 33: LSTM Y-o-Y: Actual vs Predicted

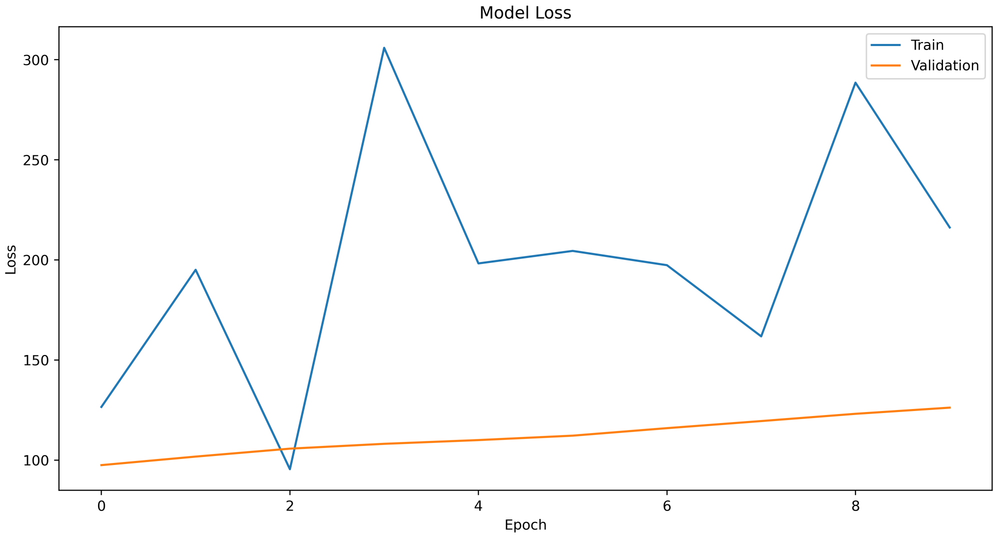


Figure 34: LSTM Model Loss: Standard Scaling Only

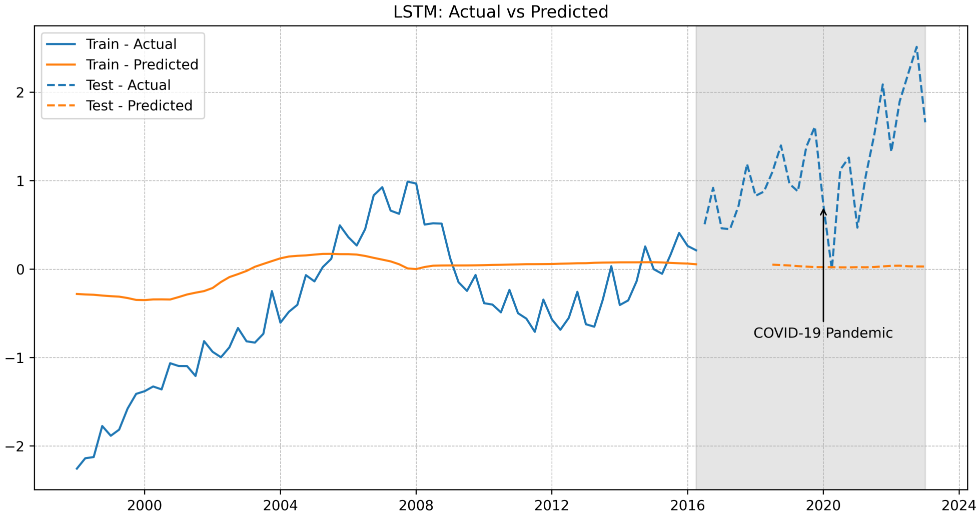


Figure 35: LSTM Std Scaling Only: Actual vs Predicted

D Transcripts

Transcript A

Speaker 1

So you might just kick things off by giving me some of the context as to why the CSO started producing flash estimates.

Speaker 2

Yeah, I think the CSO started producing flash estimates because most other European countries had been sending their estimate to eurostat and we were one of the few countries not providing a flash estimate. Eurostat were pushing us to provide one so that they would have an overall view of what the EU GDP would look like, especially with Ireland having GDP rates which are often out of line with the rest of the EU.

Speaker 1

Is this a legal requirement or is it just voluntary at this point?

Speaker 2

Not sure on the legal details but there are still some other countries who don't publish it.

Speaker 1

Do you want to give me an overview of the process for creating the flash estimates?

Speaker 2

Firstly we used to run forecasts of every single input into the process of calculating GDP using win X13. A second step would be getting early data and we use a combination of the forecasts and the early data. Once we had our unadjusted figures, we would have to seasonally adjust them.

Speaker 1

So you're interested in value, volume and price. So what exactly is it you're forecasting? Do you forecast value and then deflate it or do you separately forecast volume and price?

Speaker 2

We used to forecast all three but we only use the volume in the end.

Speaker 1

So which variables are forecast?

Speaker 1

It would be all the high-level inputs on the income and expenditure side.

Speaker 1

OK, so you originally started looking at expenditure too and decided to just go with income?

Speaker 1

So the expenditure and the income are both forecasted but expenditure wasn't accurate enough. It's very hard to forecast the expenditure side. On the income side, after we tested these for a few quarters, we could see that they were close to the T+60 estimates. So that's why we decided to go with the income. We're also interested in looking at the expenditure side, even though it's not used in the final calculation.

Speaker 1

What were the issues which made forecasting the expenditures side so challenging?

Speaker 2

With the exception of one or two components on the expenditure side like PCE, there wasn't a predictable trend.

Speaker 1

Was there more data available to forecast the income side or were the series themselves inherently more forecastable?

Speaker 2

Everything would have been the same when forecasting the income and expenditure side, because you would have been just using the previous T+60. So it's due to the income side being inherently more predictable.

Speaker 1

So which models were actually used then for the forecasting?

Speaker 2

We were using reg ARIMA in Winx 13. It performs automatic ARIMA modelling so the model structure was chosen automatically.

Speaker 1

Reg ARIMA sounds to me like you're including additional explanatory variables, but am I right in saying these are all just pure ARIMAs?

Speaker 2

That's correct.

Speaker 1

Were there any other models explored or why were the ARIMA models chosen?

Speaker 2

Methodology section had tested three different packages and they had looked into different ways of forecasting GDP. The advice was to use ARIMA models and Winx13 was used because of its ease of use.

Speaker 1

Do you know which other packages they tested?

Speaker 2

SAS and JDemetra were also tested. Overall the results were very similar, winx13 was favoured because of its ease of use. However, an issue with winx13 is that there's very few online resources.

Speaker 1

So what did you find were some of the main challenges of producing flash estimates for Ireland?

Speaker 2

One of the main challenges was we wanted early data, we didn't just want to rely on forecasts which are unlikely to capture turning points in the economy such as the pandemic. The challenge was trying to get the data in earlier because although it's a T+30 estimate, we need the data by T+25 at the latest. For the T+60 didn't, a lot of data didn't come in until T+50. We needed to encourage people to try and get their data ready earlier. Another challenge is that you can't actually rely on having the same data each quarter, sometimes certain data was ready in time, other times it wasn't.

Speaker 1

So what's some of this early data that you use?

Speaker 2

It was some LCU data and the Revenue PMOD data.

Speaker 1

Was that a big part of the motivation in doing the income method? Because I suppose they have the COE there every month and that's a big chunk of the income method right there.

Speaker 2

Not at the start, it was just easier to forecast the income side. There was more data than LCU and the PMOD. There was data around transport, hotels and retail.

Speaker 1

How did you measure the accuracy of your flash estimates? Were there particular metrics you used such as mean absolute error or mean absolute percentage error?

Speaker 2

We looked at various metrics including mean absolute error. We weren't happy with the accuracy of the estimates but Ireland is different to other European countries with the number of multinationals present. It was always going to be a challenge for accuracy when only the income method was being used. Even at T+60 there's a big difference sometimes between the income and expenditure side.

Speaker 1

What was eurostats involvement in developing the flash estimates?

Speaker 2

We met with them, took them through our process and they offered feedback.

Speaker 1

Did you get the impression from eurostat that it was maybe standard in other countries that if they use the income and expenditure method to produce their T+60 estimates, they were doing that for the flash as well?

Speaker 2

That seemed to be the case but then for a lot of countries, their T+30 result is the same for T+60 because they don't do T+60, they do a T+80. A lot of countries just do a T+30 now so when you're looking at their revisions between T+30 and T+60, there's none.

Speaker 1

Are there future plans for the direction you take the Flash estimates?

Speaker 2

We have plans to look more into the expenditure side and to see was there any other methods of forecasting that we might be able to use. So, we're going to look into incorporating both sides because we found that the income side might be really good but then at T+60 when it's combined with the expenditure side, you're getting a different view.

Transcript B

Speaker 1

Could you briefly describe the process you currently follow for producing flash GDP estimates?

Speaker 2

The current process is three fold. Firstly we employ univariate ARIMA models to predict the forecast quarter. This is done for 389 variables of interest. In addition we employ other approaches to estimation. This includes models that include an early view of real company data and data modelling of PCE by subject matter experts. The information is then assessed by a team of analysts where a final consensus is arrived.

Speaker 1

What data do you use in your models? How do you select these data?

Speaker 2

The data that we use for the ARIMA models will be the series at the last T60 QNA. This has the advantage of using data that has been quality assessed previously. In the case of models that use ‘early data’ we use available data from large companies in the Industry and IT service sectors.

Speaker 1

What type of forecasting models do you currently use for flash GDP estimates? What criteria did you use to select these models?

Speaker 2

The models used are standard ARIMA models. For the ARIMA models the parameters chosen are adopted based on past performance on the predictive capacity of the model. For the early view models we are using information from large companies so the more complete our early view the more accurate the estimate.

Speaker 1

What are the main challenges you encounter when producing flash GDP estimate? How do you manage them?

Speaker 2

The key challenge is the lack of available data and back series revision. Much of the data is volatile and can influenced by a small number of large companies. There is a tendency to believe that using more data will produce a better estimate. We found that when you have to produce an early estimate they key is to use a smaller set of data with less variability alongside real company data to produce a more reasonable estimate.

Speaker 1

How do you measure the accuracy of your flash estimates?

Speaker 2

The key element is predicting recent economic growth. We rely principally on an Income side estimate for the Flash T30 estimate. At the T60 it also has an expenditure side component incorporated. Thus when measuring accuracy we look to see how our estimate matched the Income side T60 estimate for 11 key components. Then at an overall GDP level we compare with the T60 result which incorporates the expenditure side as data is available at this time.

Speaker 1

How important is the timeliness of data in improving the accuracy of flash GDP estimates?

Speaker 2 This is absolutely key. If we had an earlier view on much of the data available at T60 particularly on the expenditure side we could produce enhanced ‘early model’ results.

Speaker 1 What role does automation play in your current GDP estimation process?

Speaker 2 Automation is key as we have a limited amount of time to produce the estimate, the deadlines are very tight. We spent much time designing a system that allows us to concentrate on quickly analysing data and results. We use a variety of software packages to achieve this.

Speaker 1

What tools or software packages do you currently use in your forecasting work?

Speaker 2

We use WinX13 for the ARIMA modelling, we use SAS for processing and reporting and we use JDemetra to seasonally adjust results. Excel is also used to analyse trends.

Speaker 1

Are there any plans to change or improve the current process of GDP flash estimates? If so, could you share some details about these plans?

Speaker 2

Our key aim is to have more early data available on the expenditure side to improve the forecasts.

Transcript C

Speaker 1

You might start off by giving me some context as to why we produce flash estimates.

Speaker 1

The background is eurostat publish an estimate of GDP in the euro area and EU at 30 days and again at 60 days. The main quarterly national accounts are published at around 60 days after the end of the quarter. I think eurostat started publishing flash estimates back in 2016 but Ireland wasn't in a position at that time to do it. Eurostat want as many countries as possible to produce estimates.

Speaker 1

Is this a legal requirement?

Speaker 2

I'm not totally sure. My understanding is that the full quarterly national accounts and annual accounts are a legal requirement. In the Eurostat publication, it's still talking about it being experimental and not everybody does it. I think maybe there's eleven countries that contribute to the 30 day estimate. So not everyone does it.

Speaker 1

Do you want to give me the overview of you know how we produce the flash estimates?

Speaker 2

The first thing to say is GDP is produced using two different methodologies. There's the output volumes side and the expenditure side. The output volumes side is based on 10 different sectors. At the T + 30 preliminary estimate, we focus on the output volume side and we don't really use an expenditure side method yet. The first step is that we carry out some ARIMA modelling. We make a first estimate which is purely based on the ARIMA models, modelling each of the 11 components separately and adding them together. The second thing we do then is to try and gather early data which is primarily data from the large cases unit here in the CSO on. It doesn't cover the entirety of the sector but it does cover some of the large cases. We also get some information on compensation of employees data that comes from Revenue. We start with a naive forecast that is only really using the back series and then we're adding in data for some of the different components and then we come up with an estimate of GDP and then the next thing we do is we convene a meeting that we call the Delphi meeting. The aim of that meeting is for us to present the forecast using the models plus how we've incorporated the early data and go through each of the 11 components in in turn and and give an overview of what that means in terms of the quarter on quarter and year on year change. The goal is to come to a consensus view of the preliminary GDP estimate. After that we would put that through the seasonal adjustment models. Finally we produce a short publication which we normally publish a day before Eurostat publish they're euro zone wide publication.

Speaker 1

Are you separately forecasting current prices and volume?

Speaker 2

Yes but the volume side takes precedence, that's the main forecast. We would also do the current prices forecast and look at the implied prices. We only publish the chain-linked constant prices ourselves. Although current and chain-linked data, both non-seasonally adjusted and seasonally adjusted, are transmitted to Eurostat.

Speaker 1

You've already answered another question, we're not directly forecasting the seasonally adjusted series?

Speaker 2

No, we're predicting each of the components and then seasonally adjusting each of the components separately and then adding them together, similar to the T+60 process.

Speaker 1

At the Delphi meeting, how many different pictures are presented? Is there a pure forecast, a reg ARIMA and there's the early data from LCU, how does that fit together?

Speaker 2

We present the pure ARIMA output and then we would present our Delphi model which for some of the different components we haven't enough early data and we will stick purely with the ARIMA model. For some components it's based off the early data and for others we would take the ARIMA model and tweak it based on what we've seen from the early data.

Speaker 1

The early data, that does feed directly into the reg ARIMA?

Speaker 2

No, it doesn't feed into the reg ARIMA at all. That doesn't take any data at all, that's purely based on the back series.

Speaker 1

Ok and the actual the outcome of the Delphi meeting, is that a mixture of the early data, the ARIMA model and some judgement?

Speaker 2

Yes, it's been different every time. In practise there hasn't been a huge amount of additional judgement applied to what we were presenting. There is potential to challenge the models based on other data etc.

Speaker 1

So what exactly are we forecasting? In so far as we're forecasting the non-seasonally adjusted data, we're probably not doing it q-o-q. Are we forecasting a y-o-y percentage change? And then that's getting applied to our existing base or how does that work?

Speaker 2

Yeah, we're basically applying the changes that we see to the back series. We look at it separately for seasonally adjusted and non-seasonally adjusted.

Speaker 1

You mentioned that we're using reg Arima, so what are some of the variables that we're incorporating in that?

Speaker 1

We're not incorporating any additional variables, each variable is forecasted from itself. It's forecast purely from its own back series.

Speaker 1

Ok so it's just pure ARIMA forecast. Presumably you have a pure ARIMA forecast for every variable, early data for some other variables and then there's some judgement as to whether you go with the early estimates data instead of the ARIMA.

Speaker 2

Exactly, so for something like public administration, education and health, we'd look at the ARIMA forecast but also the early data on compensation of employees which is a substantial part of the GVA of the sector. Element of judgement as to which figure to use or whether to go somewhere between the two.

Speaker 1

ARIMAs need to be run on stationary data, do you know how the data was transformed? So were you doing q-o-q differences or y-o-y seasonal differences or were you looking at maybe growth rates?

Speaker 2

There's automatic model selection being utilised.

Speaker 1

Ok, so in that in that case I might jump to tools and software. Is this being used in SAS or what's being used?

Speaker 1

We're using quite a few different things, the ARIMA modelling is being done in winX13. Data wrangling work is being done in SAS and then the seasonal adjustment is being done in JDemetra and R is being used for data visualisation as well.

Speaker 1

In terms of running the auto-ARIMA is that run fresh every new quarter or is it like the seasonal adjustment where it gets run up to the first quarter and then the model is fixed for all the open quarters and it doesn't get reviewed again until the annual process. So how does that work?

Speaker 2

No, it's run completely fresh, we're not constraining it.

Speaker 1

So what are some of the main challenges of producing flash GDP estimates for Ireland?

Speaker 1

The main challenge really is that at 30 days, quite a bit of the information isn't available. We can't completely align the methodology with the quarterly national accounts methodology because we're just trying to utilise what what's there. The second big challenge is that we don't have enough data to run the expenditure side models. Occasionally results between expenditure and output sides can be quite different.

Speaker 1

Are there any plans at this point to start incorporating the expenditure side?

Speaker 2

We have an internal model that we're monitoring but it's not reliable enough to bring into the final estimate. The big issue on the expenditure side is capital formation which is quite erratic. There isn't enough data at 30 days after the end of the quarter to really get a good view of that. It's one of the things we're going to keep trying to look into to see where can we get some early information and consider how we utilise the information we have to see if there's any trends that are correlated with that.

Speaker 1

And what about the modified indicators like modified domestic demand? I suppose users are particularly interested in that as a kind of a measure of the domestic economy and it gets around some of your problems with the capform being volatile.

Speaker 2

We've started looking at that but we're certainly not at the point where we're comfortable going out with the with the 30 day estimate at that point. For the PCE data we have an internal model that has been developed to give a 30 day forecast for that. For PANTS we have some early data. Challenges still with modified capform. A lot of work being done on this.

Speaker 1

For the the Flash estimates, it's currently just the output method. Is there anything to exploit in terms of historical differences between the output and expenditure methods?

Speaker 2

We're looking at that as well. The statistical discrepancy is the difference between the income and expenditure methods. We're looking at patterns but in practise it does tend to move around quite a bit.

Speaker 1

How do we assess accuracy?

Speaker 2

I suppose what we're doing is a comparison of the T+30 estimate and the T+60 estimate. Soon as the QNA comes out, what we're doing is looking at each individual component and just going back and trying to see where are the differences.

Speaker 1

In terms of data timeliness, there isn't an issue for the ARIMA models because they're pure ARIMA models, but for the early data you do get in are there are there timelines concerns there?

Speaker 2

Yeah there are huge issues with timeliness. Theoretically, if all the data for the quarterly national accounts came in at 30 days then we wouldn't have a flash estimate we would just be running the quarterly accounts at 30 days. The lack of availability of some of that data means we're trying to find models of the data for predicting the QNA. Some of the timeliness issues are things like using the monthly services index where there will only ever be two months out of the three months available. Similarly with trade data. We have some data on the large industries from the large cases but a lot of that's coming in only at T+25 day while we're trying to publish T+28.

Speaker 1

In the case of the likes of the MSI and the trade data where you have maybe two months. Are we just going with the two months or do we forecast the third month or how does that work?

Speaker 2

In most cases we're using the two months of data and comparing to the same two months the previous year.

Speaker 1

For the LCU and the COE data at T+30, how close are we to having the full information that might be available for the T + 60 do you have?

Speaker 2

The COE data, that's in, we have it. For the LCU data there's sometimes some big companies missing. In addition to that, sometimes companies can revise their returns. So they give an initial estimate before 30 days and then they come back with their revised figure later on and obviously we wouldn't have that. Sometimes those revisions can be quite big as well.

References

Divyam Aggarwal. Do bitcoins follow a random walk model? *Research in Economics*, 2019. URL https://api.semanticscholar.org/CorpusID:159446753.

Nesreen Ahmed, Amir F. Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29:594 – 621, 2010. URL https://api.semanticscholar.org/CorpusID:15917396.

Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle. 1973. URL https://api.semanticscholar.org/CorpusID:64903870.

Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19(6):716–723, December 1974.

Hirotugu Akaike. Prediction and entropy. In *A Celebration of Statistics*, pages 1–24. Springer, 1985.

Anna Almosova and Niek Andresen. Nonlinear inflation forecasting with recurrent neural networks latest version is available here. 2019. URL https://api.semanticscholar.org/CorpusID:208534619.

Ilan Alon, Min Qi, and Robert J. Sadowski. Forecasting aggregate retail sales. *Journal of Retailing and Consumer Services*, 8:147–156, 2001. URL https://api.semanticscholar.org/CorpusID:11930318.

Jushan Bai, Kunpeng Li, and Lina Lu. Estimation and inference of favar models. Mpra paper, University Library of Munich, Germany, 2014. URL https://EconPapers.repec.org/RePEc:pra:mprapa:60960.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012. ISSN ISSN 1533-7928. URL http://www.jmlr.org/papers/v13/bergstra12a.html.

Ben Bernanke, Jean Boivin, and Piotr Eliasz. Measuring the effects of monetary policy: A factor-augmented vector autoregressive (favar) approach. *The Quarterly Journal of Economics*, 120:387–422, 02 2005. doi: 10.1162/qjec.2005.120.1.387.

Harish Bhat and Nitesh Kumar. On the derivation of the bayesian information criterion. 01 2010.

George.E.P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976.

T S Breusch. Testing for Autocorrelation in Dynamic Linear Models. *Australian Economic Papers*, 17(31):334–355, December 1978. URL https://ideas.repec.org/a/bla/ausecp/v17y1978i31p334-55.html.

Roberto Cahuantzi, Xinye Chen, and Stefan Güttel. A comparison of lstm and gru networks for learning symbolic sequences, 2023.

Eddie Casey. Analytical note: Modified investment, 2023. URL https://www.fiscalcouncil.ie/wp-content/uploads/2023/05/Analytical-Note-Modified-Investment-Eddie-Casey-May-2023.pdf.

CFI. Cointegration. https://corporatefinanceinstitute.com/resources/data-science/cointegration/, 2023. Accessed: August 30, 2023.

Gary Chamberlain and Michael L. Rothschild. Arbitrage, factor structure, and mean-variance analysis on large asset markets. *NBER Working Paper Series*, 1982. URL https://api.semanticscholar.org/CorpusID:155064610.

Xiaohong Chen, Jeffrey Racine, and Norman Swanson. Semiparametric arx neural-network models with an application to forecasting inflation. *Neural Networks, IEEE Transactions on*, 12:674 – 683, 08 2001. doi: 10.1109/72.935081.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

Gregory C Chow and An-loh Lin. Best Linear Unbiased Interpolation, Distribution, and Extrapolation of Time Series by Related Series. *The Review of Economics and Statistics*, 53(4):372–375, November 1971. URL https://ideas.repec.org/a/tpr/restat/v53y1971i4p372-75.html.

Marc Claesen and Bart De Moor. Hyperparameter search in machine learning. *ArXiv*, abs/1502.02127, 2015. URL https://api.semanticscholar.org/CorpusID:17147092.

CSO. Gross domestic product (gdp). https://www.cso.ie/en/interactivezone/statisticsexplained/nationalaccountsexplained/grossdomesticproductgdp/, 2023. Accessed: August 30, 2023.

D. Dickey and Wayne Fuller. Distribution of the estimators for autoregressive time series with a unit root. *JASA. Journal of the American Statistical Association*, 74, 06 1979. doi: 10.2307/2286348.

University Duke. Forecasting: Arima for time series forecasting. https://people.duke.edu/~rnau/411arim3.htm, 2023a. Accessed: August 30, 2023.

University Duke. Time series analysis. https://people.duke.edu/~rnau/411arim3.htm, 2023b. Accessed: August 30, 2023.

J. Durbin and G. S. Watson. TESTING FOR SERIAL CORRELATION IN LEAST SQUARES REGRESSION. I. *Biometrika*, 37(3-4):409–428, 12 1950. ISSN 0006-3444. doi: 10.1093/biomet/37.3-4.409. URL https://doi.org/10.1093/biomet/37.3-4.409.

J. Durbin and G. S. Watson. TESTING FOR SERIAL CORRELATION IN LEAST SQUARES REGRESSION. II. *Biometrika*, 38(1-2):159–178, 06 1951. ISSN 0006-3444. doi: 10.1093/biomet/38.1-2.159. URL https://doi.org/10.1093/biomet/38.1-2.159.

Robert F Engle and Clive W J Granger. Co-integration and Error Correction: Representation, Estimation, and Testing. *Econometrica*, 55(2):251–276, March 1987. URL `https://ideas.repec.org/a/ecm/emetrp/v55y1987i2p251-76.html`.

Eurostat. Euro area and european union gdp flash estimates at 30 days. *Statistical Working Papers*, 2016.

Roque Benjamin Fernandez. A methodological note on the estimation of time series. *The Review of Economics and Statistics*, 63:471–476, 1981. URL `https://api.semanticscholar.org/CorpusID:119449498`.

Matthias Feurer and Frank Hutter. *Hyperparameter Optimization*, pages 3–33. 05 2019. ISBN 978-3-030-05317-8. doi: 10.1007/978-3-030-05318-5_1.

John Fitzgerald. National Accounts for a Global Economy: the Case of Ireland. Trinity Economics Papers tep0418, Trinity College Dublin, Department of Economics, May 2018. URL `https://ideas.repec.org/p/tcd/tcduee/tep0418.html`.

William R. Foster, Fred L. Collopy, and Lyle H. Ungar. Neural network forecasting of short, noisy time series. *Computers & Chemical Engineering*, 16:293–297, 1992. URL `https://api.semanticscholar.org/CorpusID:121516607`.

Milton Friedman. The interpolation of time series by related series. *Journal of the American Statistical Association*, 57:729–757, 1962. URL `https://api.semanticscholar.org/CorpusID:116204247`.

Leslie Godfrey. Testing against general autoregressive and moving average error models when the regressors include lagged dependent variables. *Econometrica*, 46(6):1293–1301, 1978. URL `https://EconPapers.repec.org/RePEc:ecm:emetrp:v:46:y:1978:i:6:p:1293-1301`.

Clive Granger and P. Newbold. Spurious regressions in econometrics. *Journal of Econometrics*, 2 (2):111–120, 1974. URL `https://EconPapers.repec.org/RePEc:eee:econom:v:2:y:1974:i:2:p:111-120`.

Jan J. J. Groen and George Kapetanios. Model selection criteria for factor-augmented regressions. Staff Reports 363, Federal Reserve Bank of New York, 2009. URL `https://ideas.repec.org/p/fip/fednsr/363.html`.

John A Gubner. *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge University Press, 2006.

James Douglas Hamilton. *Time Series Analysis*. Princeton University Press, 1994. URL `https://www.worldcat.org/title/time-series-analysis/oclc/1194970663&referer=brief_results`.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009. URL `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`.

David F Hendry and Bent Nielsen. *Econometric Modeling: A Likelihood Approach*. Princeton University Press, 2007.

Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, jan 2021. doi: 10.1016/j.ijforecast.2020.06.008. URL https://doi.org/10.1016%2Fj.ijforecast.2020.06.008.

Timothy Hill, Marcus O'Connor, and William Remus. Neural network models for time series forecasts. *Management Science*, 42:1082–1092, 11 1996. doi: 10.1287/mnsc.42.7.1082.

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

Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *ONR Research Memorandum*, 52, 1957.

Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5–10, 2004. URL https://ideas.repec.org/a/eee/intfor/v20y2004i1p5-10.html.

Rob J Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, 2021.

IMF. Chapter 10: Time series analysis and forecasting. https://www.imf.org/external/pubs/ft/qna/pdf/2017/chapter10.pdf, 2017. Accessed: August 30, 2023.

Investopedia. Stationarity. https://www.investopedia.com/articles/trading/07/stationary.asp, 2023. Accessed: August 30, 2023.

Soren Johansen. Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models. *Econometrica*, 59(6):1551–80, 1991. URL https://EconPapers.repec.org/RePEc:ecm:emetrp:v:59:y:1991:i:6:p:1551-80.

Julie Josse and Francois Husson. Selecting the number of components in pca using cross-validation approximations. *Computational Statistics & Data Analysis*, 56, 06 2012. doi: 10.1016/j.csda.2011.11.012.

Suh Y. Kang. An investigation of the use of feedforward neural networks for forecasting. 1992. URL https://api.semanticscholar.org/CorpusID:59773474.

Sune Karlsson. Forecasting with bayesian vector autoregression. volume 2, chapter Chapter 15, pages 791–897. Elsevier, 2013. URL https://EconPapers.repec.org/RePEc:eee:ecofch:2-791.

John Maynard Keynes. *The General Theory of Employment, Interest and Money*. Palgrave Macmillan, 1936.

Simon Kuznets. *National Income, 1929-1932*. NBER, 1934.

Denis Kwiatkowski, Peter C. B. Phillips, Peter Schmidt, and Yongcheol Shin. Testing the null hypothesis of stationarity against the alternative of a unit root : How sure are we that economic time series have a unit root? *Journal of Econometrics*, 54(1-3):159–178, 1992. URL `https://ideas.repec.org/a/eee/econom/v54y1992i1-3p159-178.html`.

Robert Litterman. A random walk, markov model for the distribution of time series. *Journal of Business & Economic Statistics*, 1(2):169–73, 1983. URL `https://EconPapers.repec.org/RePEc:bes:jnlbes:v:1:y:1983:i:2:p:169-73`.

Luigi Longo, Massimo Riccaboni, and Armando Rungi. A neural network ensemble approach for GDP forecasting. *Journal of Economic Dynamics and Control*, 134(C), 2022. doi: 10.1016/j.jedc.2021.10427. URL `https://ideas.repec.org/a/eee/dyncon/v134y2022ics016518892100213x.html`.

Spyros Makridakis, A Andersen, R Carbone, R Fildes, M Hibon, R Lewandowski, J Newton, E Parzen, and R Winkler. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, 1(2):111–153, 1982.

Peter McAdam and Paul McNelis. Forecasting inflation with thick models and neural networks. *Economic Modelling*, 22(5):848–867, 2005. URL `https://EconPapers.repec.org/RePEc:eee:ecmode:v:22:y:2005:i:5:p:848-867`.

Christian Muller. A maximum likelihood short-cut to the chow-lin procedure. *Zurich University of Applied Sciences School of Management and Law*, 20XX.

Emi Nakamura. Inflation forecasting using a neural network. *Economics Letters*, 86(3):373–378, 2005. URL `https://EconPapers.repec.org/RePEc:eee:ecolet:v:86:y:2005:i:3:p:373-378`.

OECD. Title of the webpage, 2002. URL `https://web.archive.org/web/20210627164746/https://stats.oecd.org/glossary/detail.asp?ID=1163`.

Peter Phillips and Pierre Perron. Testing for a unit root in time series regression. Cowles Foundation Discussion Papers 795R, Cowles Foundation for Research in Economics, Yale University, 1987. URL `https://EconPapers.repec.org/RePEc:cwl:cwldpp:795r`.

Tommaso Proietti. Temporal Disaggregation by State Space Methods: Dynamic Regression Methods Revisited. Econometrics 0411011, University Library of Munich, Germany, November 2004. URL `https://ideas.repec.org/p/wpa/wuwpem/0411011.html`.

Adam Richardson, Thomas van Florenstein Mulder, and Tugrul Vehbi. Nowcasting GDP using machine learning algorithms: A real-time assessment. Reserve Bank of New Zealand Discussion Paper Series DP2019/03, Reserve Bank of New Zealand, November 2019. URL `https://ideas.repec.org/p/nzb/nzbdps/2019-3.html`.

Steven L. Scott and Hal Varian. Bayesian variable selection for nowcasting economic time series. In *Economic Analysis of the Digital Economy*, pages 119–135. National Bureau of Economic Research, Inc, 2015. URL `https://EconPapers.repec.org/RePEc:nbr:nberch:12995`.

Steven L. Scott and Hal R. Varian. Predicting the present with bayesian structural time series. *Econometrics: Econometric & Statistical Methods - General eJournal*, 2013. URL https://api.semanticscholar.org/CorpusID:9060868.

Ramesh Sharda and R. B. Patil. Connectionist approach to time series prediction: an empirical test. *Journal of Intelligent Manufacturing*, 3:317–323, 1992. URL https://api.semanticscholar.org/CorpusID:15332409.

Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394–1401, 2018. URL https://api.semanticscholar.org/CorpusID:58671842.

J.M.C. Santos Silva and Fátima Cardoso. The chow-lin method using dynamic models. *Economic Modelling*, 18:269–280, 2001. URL https://api.semanticscholar.org/CorpusID:121457187.

Christopher Sims. Macroeconomics and reality. *Econometrica*, 48(1):1–48, 1980. URL https://EconPapers.repec.org/RePEc:ecm:emetrp:v:48:y:1980:i:1:p:1-48.

James Stock and Mark Watson. A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series. NBER Working Papers 6607, National Bureau of Economic Research, Inc, 1998. URL https://EconPapers.repec.org/RePEc:nbr:nberwo:6607.

Petre Stoica and Yngve Selén. Model-order selection: a review of information criterion rules. *IEEE Signal Processing Magazine*, 21:36–47, 2004. URL https://api.semanticscholar.org/CorpusID:17338979.

John B Taylor. Discretion versus policy rules in practice. *Carnegie-Rochester conference series on public policy*, 39:195–214, 1993.

Timo Teräsvirta, Dick van Dijk, and M. C. Medeiros. Smooth transition autoregressions , neural networks , and linear models in forecasting macroeconomic time series : A re-examination. 2003. URL https://api.semanticscholar.org/CorpusID:14384993.

Henri Theil. *Principles of Econometrics*. John Wiley & Sons, 1971.

S. Theodoridis. *Machine Learning: A Bayesian and Optimization Perspective*. 05 2015. ISBN 978-0-12-801522-3.

Michael Ward and John Ahlquist. *Maximum Likelihood for Social Science: Strategies for Analysis*. 11 2018. ISBN 9781107185821. doi: 10.1017/9781316888544.

Halbert L. White. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48:817–838, 1980. URL https://api.semanticscholar.org/CorpusID:17463469.

Peter T. Yamak, Li Yujian, and Pius Kwao Gadosey. A comparison between arima, lstm, and gru for time series forecasting. *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, 2019. URL https://api.semanticscholar.org/CorpusID:211104801.

G. Alastair Young. Mathematical Statistics: An Introduction to Likelihood Based Inference Richard J. Rossi John Wiley & Sons, 2018, xv + 422 pages, £85.99, ebook ISBN: 978-1-118-77104-4, LCCN 2018010628 (ebook). *International Statistical Review*, 87(1):178–179, April 2019. doi: 10.1111/insr.12315. URL https://ideas.repec.org/a/bla/istatr/v87y2019i1p178-179.html.