Masterthesis

The Nonlinearity of Crises: Machine Learning Approaches to Economic Forecasting

December 31, 2019

Name	Julian Oliver Dörr
Street	Hornthalstraße 6
Address	96047 Bamberg
ID	1911533
Institution	University of Bamberg
Faculty	Social Sciences, Economics, and Business Administration
Field of study	European Economic Studies (EES)
Supervisor	Prof. Dr. Christian Aßmann

Abstract

Macroeconomic forecasters often show a poor track record in producing reliable predictions. Especially times of economic crises have been consistently missed by traditional forecasting models, which is why the global financial crisis of 2007 - 2009 caught many people by surprise. Theory-based models and pure time series formulations, both traditionally used in macroeconomic forecasting, suffer the curse of dimensionality when being confronted with a high dimensional space of possible predictors. In times of increasing data availability, this opens the way to rethink macroeconomic forecasting. This paper analyzes the application of machine learning methods in forecasting real GDP growth. Machine learning poses a natural extension to the more traditional models because it is designed to extract information from high dimensional feature spaces. Moreover, machine learning methods are nonlinear by construction which allows them to capture nonlinear relations typically encountered among macroeconomic variables in recessions. Given the failure of existing forecasting models and the advantages of machine learning, the goal of this paper is to assess to which extent machine learning algorithms may contribute to the challenge of macroeconomic forecasting.

Contents

List of Figures

List of Tables

List of Acronyms

A

ACF Autocorrelation Function
ADF-Test Augmented Dickey-Fuller (ADF) Test
AIC Akaike Information Criterion
AICc corrected Akaike Information Criterion
AR Autoregressive
ARIMA Autoregressive Integrated Moving Average
ARMA Autoregressive Moving Average

B

BEA Bureau of Economic Analysis
BIC Bayesian Information Criterion

D

DM-Test Diebold-Marino (DM) Test
DSGE Dynamic Stochastic General Equilibrium

E

ECB European Central Bank

F

FAVAR Factor-Augmented Vector Autoregressive
FED Federal Reserve System
FRED Federal Reserve Economic Data

G

GB Gradient Boosting
GDP Gross Domestic Product

I

IMF International Monetary Fund

K

KKT conditions Karush–Kuhn–Tucker (KKT) conditions
KPSS-Test Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test

L

LSTM Long Short-term Memory

M

MA Moving Average
MAPE Mean Absolute Percentage Error
MdRAE Median Relative Absolute Error
MSE Mean Squared Error

N

NBER National Bureau of Economic Research

O

OECD Organization for Economic Co-operation and Development
OLS Ordinary Least Squares

P

PACF Partial Autocorrelation Function
PCA Principal Component Analysis

R

RelRMSE Relative RMSE
RF Random Forest
RMSE Root Mean Squared Error
RNN Recurrent Neural Network
RSS Residual Sum of Squares
RW Random Walk

S

SVM Support Vector Machine
SVR Support Vector Regression

V

VAR Vector Autoregressive

List of Symbols

Y_t	seasonally adjusted real GDP
y_t	seasonally adjusted quarter-over-quarter growth rate of real GDP expressed in percent
$\hat{y}_t$	real GDP growth forecast
e_t	forecasting error
e_t^b	forecasting error of benchmark model
r_t	relative forecasting error
h	forecasting horizon
D	number of distinct time series in feature space
x_d	d-th time series in feature space
$\boldsymbol{x}$	collection of all time series in feature space
$\boldsymbol{X}$	feature space
Θ	set of learner- and hyperparameter
$f(\boldsymbol{x};\Theta)$	forecasting function
$\mathcal{I}_t$	information set
p	autoregressive lag order
θ	AR coefficient
ϵ_t	white noise
σ^2	variance of white noise
q	moving average lag order
ϕ	MA coefficient
K	number of distinct time series in VAR model
$\boldsymbol{y}_t$	vector of K selected time series including real GDP growth
A	AR coefficient matrix
$\boldsymbol{a}_0$	vector of intercept terms
$\boldsymbol{\epsilon}_t$	vector of white noise components
Σ_ϵ	contemporaneous covariance matrix of white noise
V	number of unobserved factors in FAVAR model
$\boldsymbol{f}_t$	vector of V unobserved factors
$\tilde{\boldsymbol{f}}_t$	vector of estimated factors
Λ	matrix of factor loadings
I	identity matrix
$\boldsymbol{v}_t$	vector of error terms
$\boldsymbol{c}_t$	vector of principal components
J	number of terminal leaf nodes in regression tree
R_j	j-th terminal leaf node
$\hat{y}_{R_j}$	GDP forecast of samples falling into the j-th terminal leaf node
s	splitting point in domain of splitting variable
$node_{min}$	minimum number of observations in each terminal leaf node
$T(\boldsymbol{x};\Theta)$	regression tree
$\mathbb{1}(\cdot)$	indicator function
M	number of trees
d_{try}	number of predictor variables randomly sampled at each node
$depth_{max}$	interaction depth in regression trees
ν	learning rate in Gradient Boosting
res	residual vector
g	gradient of loss function
$\boldsymbol{\beta}$	parameter vector in Support Vector Regression
b	intercept term
ε	defines margin in which deviations from the approximating function are allowed
C	regularization parameter
$\xi_i^{(*)}$	slack variable(s) allowing observations to be located outside the ε-margin
$\mathcal{L}$	Lagrange function
$\alpha_i^{(*)}$	Lagrangian multiplier(s)
$\eta_i^{(*)}$	Lagrangian multiplier(s)
$\mathbb{S}$	set of Support Vectors
$\Phi(\boldsymbol{x})$	mapping function
$\boldsymbol{\mathfrak{X}}$	higher order feature space

Q	number of features in higher order feature space
$K(\boldsymbol{x}_i, \boldsymbol{x})$	kernel function
w	power in polynomial kernel
γ	kernel parameter
H_0	null hypothesis

1 Introduction

1.1 Call for New Approaches in Economics

The prediction of future economic development is a controversial issue which has kept economists busy for years. The International Monetary Fund releases its global forecasting calculations biannually in the *World Economic Outlook*. The Organization for Economic Co-operation and Development (OECD) publishes its *Economic Outlook* which comprises Gross Domestic Product (GDP) forecasts of its member states twice a year. The World Bank, too, conducts economic projections published in the *Global Economic Prospects* and the European Central Bank (ECB) issues its own *Survey of Professional Forecasters*. The list of renowned economic institutions is long which highlights the importance of the topic. In fact, many decision-makers rely on economic forecasts and closely watch what economists believe to happen in the near and distant future. Businesses, for example, base their spending and hiring plans on the future outlook of the economy. Also, financial investors show in their investment decisions a high degree of sensitivity to economic forecasts. Last but not least, government officials rely on macroeconomic projections in their policy-making, aiming at cushioning or even preventing economic downturns. Given the importance and the resources spent on economic forecasting, a natural question arises: How well can economists predict future economic growth?

As it turns out, the answer to this question is rather disillusioning, especially when it comes to economists' capability of forecasting economic crises. This has become a humbling truth in the global financial crisis of 2007 - 2009. Both extent and severeness of the financial crisis have been largely unforeseen by most economists. In fact, as Andrew G Haldane, Chief Economist at the Bank of England points out, none of the economic forecasters had anticipated a recession and the majority had not even expected a slight economic slowdown prior to 2007. It turned out that the 2008 one-year ahead GDP growth forecast error of 27 reputable economic forecasting institutions amounted to 8 percentage points on average (Bank of England, 2016). The failure of not even being able to predict roughly what in 2009 climaxed as the worst financial crisis after the Great Depression in 1930 pushed macroeconomics as a scientific field of research into its own crisis. In the UK, for instance, the Queen questioned economic researchers during a visit at the London School of Economics how nobody had foreseen the crisis, especially given its ferocity. In response to this critical question, a group of leading academics and practitioners from the field of economics drafted a letter which was sent to Buckingham Palace on July 22, 2009. In this letter, the authors mention among other things that economic forecasting was based on 'financial and economic models that were good at predicting the short-term and small risks, but few were equipped to say what would happen when things went wrong as they have' in 2008 (Besley & Hennessy, 2009, p. 9). Many renowned macroeconomists followed this self-critical assessment and started to question their own discipline in the aftermath of the global financial crisis. Much of their criticism challenges the methods and models predominantly used in economic research.

Blanchard and Romer, for example, have criticized the predominant use of Dynamic Stochastic General Equilibrium (DSGE) models which are founded on often flawed economic theory. Blanchard (2014) strongly challenges the linear view of mainstream macroeconomics on how economic aggregates relate to each other. He highlights that the global financial crisis has taught that small shocks in one economic sector, e.g. the U.S. housing market, can cause major disruptions in the global economy. Such extreme nonlinear relations between markets and economies cannot be captured by 'techniques [that] were best suited to a worldview in which economic fluctuations occurred but were regular, and essentially self correcting' (Blanchard, 2014, p. 28). It seems little surprising that linear models produce poor forecasts if the reality is characterized by strong degrees of nonlinearity. Chauvet and Potter (2013) support the view that during recessions linear relations break down, causing linear forecasting models to produce large forecasting errors.

Romer's (2016) criticism strongly contests the doctrine of exogenous shocks in DSGE forecasting models. He argues quite sharply that 'post-real macro models [and their] predictions were wildly incorrect, and [...] the doctrine on which they were based is fundamentally flawed' (Romer, 2016, p. 19). Romer refers to the assumption of exogenous shocks in theory-based macro models which 'explain' a significant fraction of forecast variance in GDP. In fact, if much of the forecasting accuracy in DSGE models is determined by exogenous events that the model itself cannot explain, the question arises which value such models add to the task of macroeconomic forecasting. Romer even goes as far as questioning whether the current state of macroeconomic modeling still qualifies as scientific research.

In a similar vein, Haldane strongly criticizes the 'methodological mono-culture' in macroeconomic fore-

casting which 'spectacularly' failed to even come close to predicting a recession in 2008 (Bank of England, 2016, p. 7). On the one hand, he blames the uniform structure of economic forecasting models which were not designed to foresee such events. On the other hand, he criticizes the lack of interdisciplinary in economics which hinders the discipline to come up with new models to forecast economic fluctuations. Haldane concludes that 'even if some of the post-crisis criticism of workhorse macro-economic models is overdone, it still raises the question of whether new modeling approaches might be explored [...] which better match real-world dynamics.' (Bank of England, 2016, p. 8). He continues that 'there could be considerable scope for disciplinary cross-pollination of ideas and models' (Bank of England, 2016, p. 8). Similarly, Blanchard (2014, p. 31) concludes his criticism by stressing the need to 'explore [...] all sorts of models [which] are more aware of nonlinearities and the dangers they pose'. Generally, the methodological malaise that macroeconomic forecasters experienced in the aftermath of the last crisis opens the gates to rethink macroeconomic forecasting and to explore new techniques of improving and developing existing methods. Renowned economists themselves have called for a renewal of modeling techniques in economic forecasting.

Following this call for new approaches in macroeconomic forecasting, this paper pursues an atheoretical approach to the task of forecasting GDP growth. It refrains from incorporating macroeconomic theory or micro-founded behavioral science (such as in DSGE models) into the forecasting models. Rather, it focuses on the efficient exploitation of information entailed in past data for the prediction of future realizations. The fundamental belief in this information-theoretic approach is that past and current realizations of macroeconomic variables allow to make projections about the future state of the economy. A natural starting point for such atheoretical models is the use of machine learning methods which are designed to algorithmically extract information encrypted in past data.

1.2 Machine Learning: A new Approach in Economic Forecasting?

Similar to the above criticism, Reis (2018) admits that macroeconomic forecasters have performed poorly in the past but questions whether the task of macroeconomists is to provide precise forecasts of future economic activity or if their duty is to provide policy-making guidance in a forward looking manner. Reis' concern is fundamental to this paper and requires some further insight into the difference between forecasting models based on economic theory and atheoretical machine learning methods. The former group of models - with DSGE as the workhorse in macroeconomics - consists of structural models based on economic rationale. They aim at describing the interrelation of macroeconomic variables with economic theory. Founded on behavioral microeconomics such as the Euler equation for intertemporal consumption smoothing and policy rules such as the Taylor principle for Central Bank's interest rate setting, theory-based models allow to 'tell a story' (Giacomini, 2015, p. 24). They are designed to illustrate causal channels about the relationships and dynamic forces that drive the economy. However, this approach produces valid forecasts only if the model describes the real world mechanisms correctly. Machine learning methods, in contrast, follow a data-driven approach free of any theory-based constraints which usually compresses the data in a tight (and often linear) corset. It is the approach to let the data speak which, in the context of a forecasting task, means that today's information about the economy has something to tell about how the economy looks tomorrow. In this respect, machine learning models focus on minimizing the forecasting error with the objective to produce the most accurate projection about the future state of the economy.

Most Central Banks nowadays use DSGE models for both analyzing policy effects and forecasting (Giacomini, 2015). However, there is typically a trade-off to make between a model's forecasting capability and its theoretical foundation desirable for policy evaluations (Giacomini, 2015; Pagan, 2003). Neglecting this trade-off is another reason why macroeconomic forecasters have performed so poorly in the past. They have misused theory-based policy models for the purpose of forecasting. In light of this trade-off and in response to Reis' criticism, the author of this paper stresses macroeconomists' necessity of being capable of handling both - policy evaluation and forecasting - but questions whether both tasks can be accomplished by the same class of models in a satisfying way. In fact, the objective of sketching future economic growth as precisely as possible necessitates techniques which efficiently process complex and rich sets of information. This contrasts with theory-based methods where researchers pick one model and a selection of variables based on principles and therefore essentially restrict the set of information entering their forecasting model.

Many economists acknowledge that the amount of information available for economic research has been grown rapidly and that macroeconomics increasingly acts in a data-rich environment (Bernanke & Boivin,

2003; Clements & Hendry, 2011). This has encouraged economists to consider general applications of data-driven machine learning methods in the field of economics in recent years (see for example Athey and Imbens (2017), Mullainathan and Spiess (2017), Varian (2014)). The scope of this paper is to analyze whether machine learning algorithms can contribute to the specific issue of forecasting GDP growth. To answer this question three specific supervised learning algorithms, Random Forest (RF), Gradient Boosting (GB) and Support Vector Regression (SVR), are applied in forecasting U.S. GDP growth. The performance of these models is benchmarked against more traditional econometric time series models which have been extensively used in literature before. These techniques comprise Autoregressive Integrated Moving Average (ARIMA) models, Vector Autoregressive (VAR) models and Factor-Augmented Vector Autoregressive (FAVAR) models.

Literature on the specific application of machine learning algorithms in GDP forecasting is very sparse. Biau and D'Elia (2010) have pioneered the use of RF to forecast European Union short-term GDP growth. They find that RF outperforms linear Autoregressive (AR) models and recommend the use of RF if forecasters are confronted with a large set of potential predictors. Buchen and Wohlrabe (2011) benchmark GB forecasts of U.S. industrial production against factor models. Similar to Biau and D'Elia (2010), they find that GB is a serious competitor especially if researchers face a large set of potential predictor variables. Gogas, Papadimitriou, and Takli (2013) use a SVR framework to forecast U.S. real GDP. However, their analysis is limited to a small set of monetary aggregates as predictor variables and they refrain from benchmarking against other modeling approaches. Tiffin (2016) uses RF to nowcast GDP growth rates of Lebanon, a country which publishes its official GDP figures with a time lag of up to two years. Their model indicates a Root Mean Squared Error of 1.17 in an out-of-sample nowcast which does not include the global financial crisis. Compared to the results of this paper, this is a relatively poor performance. Lehmann and Wohlrabe (2016, 2017) use boosting to forecast German economic activity both on the national and the regional level. They use Ordinary Least Squares (OLS) as base learners and find that the boosting model outperforms a boosted ARIMA model. Jung, Patnam, and Ter-Martirosyan (2018) forecast real GDP growth of seven industrial and emerging countries using an ensemble of different machine learning methods. The ensemble comprises RF and SVR among other methods. In their analysis, machine learning has a higher forecasting accuracy than AR and VAR models. Gogas, Papadimitriou, Matthaiou, and Chrysanthidou (2015) reformulate the forecasting issue into a classification task. They use Support Vector Machine (SVM) to predict recessions by means of information from the yield curve. In an out-of-sample assessment which includes the global financial crisis, the SVM classifier outperforms econometric probit and logit models. In a similar vein, Döpke, Fritsche, and Pierdzioch (2017), Ng (2014) use GB with regression trees as base learners to forecast recession probabilities for the Canadian and German economy, respectively. Their analysis focuses on the relative importance of economic indicators in forecasting the likelihood of future recessions. There are some more articles using neural networks to forecast GDP growth (see Tkacz (2001) for example). However, while neural networks certainly belong to the class of algorithmic machine learning methods, neural network forecasting is beyond the scope if this paper. Hassania and Silva (2015) review the challenges forecasters face when being confronted with 'big data'. Part of their paper surveys 'big data' forecasting of GDP by means of advanced statistical methods.

This paper aims at validating the results of previous research. Moreover, the analysis focuses on two specific aspects. First, it is examined how algorithmic models perform if they are confronted with different sets of information. Since machine learning models are designed to cope with rich information sets - nowadays often labeled as 'big data' -, it is assumed that forecasting errors get smaller the more variables are included in the forecasting tasks. To the author's best knowledge, GDP forecasts produced by machine learning methods have not been analyzed before in this specific context. Second, it is analyzed how well machine learning forecasts perform in times of recessions relative to the forecasts produced by more traditional time series models. Therefore, the general research scope is to analyze to what extent machine learning poses a suitable tool in macroeconomic forecasting and how it may complement traditional forecasting tools.

The paper proceeds as follows. Section 2 introduces the data sources and the predictor variables used in forecasting U.S. GDP growth. Section 3 explains the forecasting strategy within the machine learning framework. Furthermore, it introduces the methodological foundations of the time series models and the machine learning methods used in this paper. It also outlines how RF, GB and SVR can be used for forecasting. Section 4 presents the estimated forecasting models and discusses the final forecasting performance. Section 5 concludes.

2 Data

2.1 Gross Domestic Product (Target)

The U.S. Bureau of Economic Analysis (BEA) is used as the primary source for U.S. Gross Domestic Product data as it provides detailed information on how GDP figures are calculated. They track all U.S. economic accounts and related statistics which together form the final GDP figures. Consequently, BEA's data source allows to gain a deep understanding of the derivation and composition of one of the most closely watched aggregated statistic (see Bureau of Economic Analysis (2016) for more details). This paper aims at forecasting real changes in GDP attributable to changes in the amount of final goods and services produced in the economy excluding effects related to price changes. Since different price indices are used as predictor variables in this paper, it is important to work with a deflated GDP figure as target variable. For this purpose, BEA provides a measure of real GDP expressed in chained 2012 dollars.[1] This paper uses the level of quarterly real GDP figures in order to calculate discrete quarter-over-quarter growth rates of real GDP

$$y_t = 100 \left[\left(\frac{Y_t}{Y_{t-1}} \right) - 1 \right] \tag{1}$$

with Y_t as GDP level at time t. The quarter-over-quarter growth rate of real GDP expressed in percent, y_t, forms the target variable in all forecasting models.

Figure 1 shows the respective time series of real GDP in levels on the right axis and real GDP growth realizations on the left axis. The time horizon used in this paper comprises realizations from the first quarter of 1959 to the second quarter of 2019 on a quarterly basis. This is equivalent to a time series with 242 observations. The time series are split into training and test sets in order to test the out-of-sample performance of different forecasting models. The test set used for this purpose starts in 2007-Q2 and thus includes the global financial crisis. Moreover, the figure highlights periods of recession in gray according to the recession definition of the National Bureau of Economic Research (NBER). NBER defines a recession as 'significant decline in economic activity spread across the economy, lasting more than a few months, normally visible in real GDP, real income, employment, industrial production, and wholesale-retail sales' (National Bureau of Economic Research, 2012). With this definition, the NBER describes the business cycle as continuous succession of expansions and recessions.[2] Following this logic, the different forecasting models developed in this paper are also tested in their capacity to foresee contractions in the business cycle at an early stage by focusing on the predicted sign in the real GDP forecast.

2.2 Predictor Variables (Features)

The research division of the Federal Reserve Bank of St. Louis offers with its data service Federal Reserve Economic Data (FRED) a comprehensive resource for macroeconomic research. FRED grants access to more than 500,000 financial and economic variables from various public and private sources for the U.S. economy. This paper uses FRED-QD, a collection of 248 quarterly macroeconomic time series mainly retrieved from Federal Reserve Economic Data and enriched by variables from other public sources such as stock indices from NASDAQ and S&P. The quarterly data comprises a period from 1959-Q1 to 2019-Q2. FRED-QD is proposed as starting point for 'big data' research in macroeconomics (McCracken & Ng, 2016) and is therefore highly suitable for time series machine learning models. In fact, with 248 explanatory variables and 242 observations most conventional econometric models are likely to run out off degrees of freedom without some form of variable preselection. Machine learning methods do not face this curse of dimensionality which is why they are appealing candidates for forecasts based on data with so many potential predictors. Besides the wealth of information entailed in FRED-QD, the dataset offers further advantages. It is updated and published regularly and it covers all data revisions that often complicate macroeconomic research.[3]

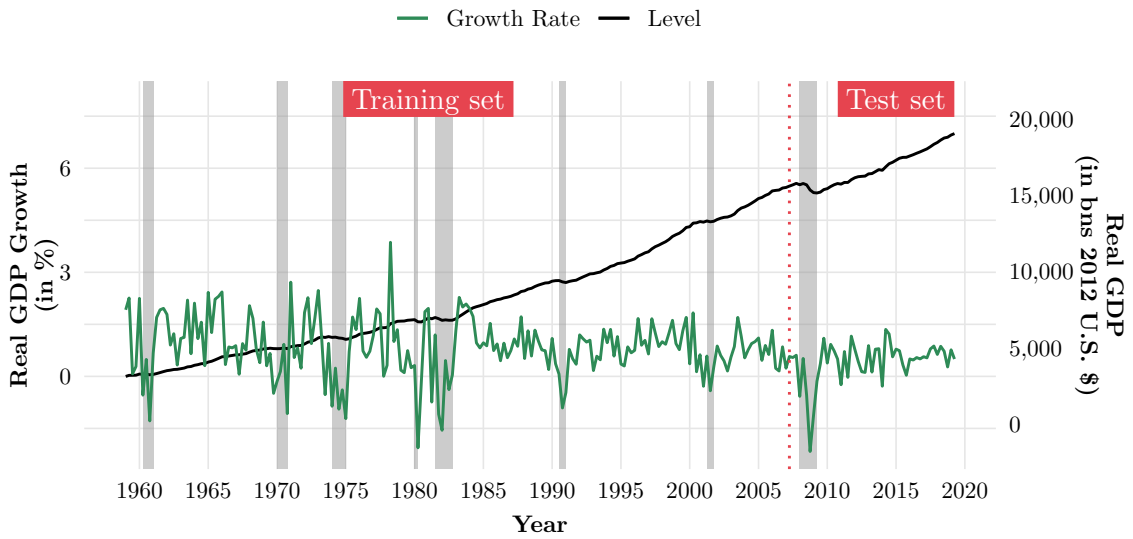
FRED-QD comprises components of GDP as well as variables from industrial production and employment, time series from the housing market, inventory, sales and orders information, various price indices, earnings and productivity measures, interest and exchange rates, stock market information, macroeconomic money and credit measures as well as business, household and public balance sheet information. For the majority of the series, recording starts in the first quarter of 1959. Unfortunately, 38 of the

[1] See Landefeld, Moulton, and Vojtech (2003) for more details on the methodology of chain-type indices.

[2] The NBER determines beginning and end of a recession and publishes the dates on its homepage. In the past, this has often happened with a time lag of up to 21 months, clearly showing the ex-post nature of their assessment.

[3] See section 2.3 for more details on the topic of data revisions.

Figure 1: U.S. Real GDP



Note: Figure shows BEA GDP figures from 1959-Q1 to 2019-Q2. GDP figures are seasonally adjusted by removing variations occurring in the same quarter every year. Seasonal adjustments follow BEA's methodology (for more information, see Cowan, Smith, and Thompson (2018)). Moreover, figures are expressed in real terms as chained 2012 U.S. dollars. More information on inflation adjustments by chaining techniques can be found in Landefeld, Moulton, and Vojtech (2003).

time series have only a limited history with recordings starting later than 1959. Imputation methods for these variables make little sense since the missing values only occur at the beginning of the series. As a consequence, these series have to be dropped for the further analysis. Table 9 in appendix A provides a detailed overview of all final predictor variables that enter the forecasting models.

In a forecasting setup, it is particularly important to work with macroeconomic variables which entail early signals of major disruptions in economic performance. Fortunately, the dataset contains almost all variables which the OECD defines as leading indicators of large economic adjustments for the U.S. economy (OECD, 2019). According to the OECD (2019), the following variables exhibit a leading relationship with U.S. Gross Domestic Product: total number of housing starts (HOUST in appendix A), manufacturer's new orders of durable goods in dollars (AMDMN_OX), share prices such as the S&P 500 stock price index (S_P_500), the consumer sentiment index from the University of Michigan (UMCSEN_TX), weekly hours worked in manufacturing (AWHMAN) and the interest rate spread between 3-Month treasury constant maturity and federal funds rate (TB3SMFFM). The only early indicator suggested by OECD that is not included in the dataset is a business confidence indicator of the manufacturing sector.[4] Appendix B provides a visual inspection of the above mentioned indicators. The visualizations reveal a leading relationship between the indicators' cyclicality and the starting dates of U.S. recessions. This means that the above variables, indeed, serve as early predictors of economic downturns in the U.S. They tend to have similar cyclical fluctuations as the business cycle but with the crucial difference that they precede fundamental movements in GDP growth. This paper estimates a vector autoregressive forecasting model built exclusively on U.S. leading indicators. Section 4.2.1 analyzes the forecasting performance of this model as well as the performance of machine learning methods which incorporate only the leading indicators.

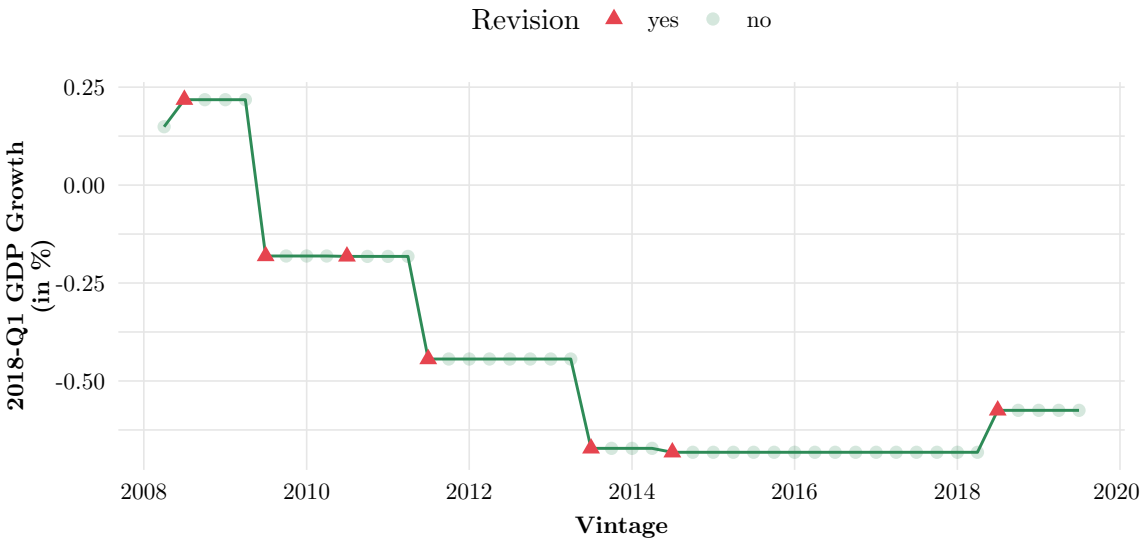
2.3 Real-time Data versus Revised Data

Macroeconomic research is often confronted with a peculiarity in terms of data reporting which is usually not encountered in other fields of research. It is common that macroeconomic variables are revised after their initial publication, implying that today's realization of a variable can differ from the realization at a later point in time. There are many reasons why macroeconomic variables experience, often substantial, revisions. When agencies publish data for the first time, they are confronted with only a limited set of information. As time passes, more information becomes available, allowing for a more accurate measurement of macroeconomic indicators. Another reason are changes in accounting standards or methodology (Croushore & Stark, 2000). For example, variables measured in real terms must be revised if the base year is updated. These factors explain why one and the same variable may have different values de-

[4] See OECD (2012) for more details on the methodology of leading indicators.

pending on the vintage in which the variable is reported. This becomes obvious in figure 2 where the value of the 2008-Q1 GDP growth figure is displayed for different reporting vintages. In the first five quarters after the initial publication, 2008-Q1 GDP growth has been reported as positive number. Only after the second revision of the figure in 2009-Q3, six quarters after the first release, agencies have had enough information available to update to a negative growth figure. In later revisions the value has been corrected further downward.

Figure 2: Actual 2008-Q1 Real GDP Growth by Vintages



Note: Real-time data is retrieved from the Real-Time Data Research Center of the Federal Reserve Bank of Philadelphia. Red triangles indicate dates when the 2018-Q1 growth figure has been revised. Green points represent periods of no revision.

The Real-Time Data Research Center of the Federal Reserve Bank of Philadelphia provides a dataset of real-time quarter-over-quarter GDP growth rates. This dataset comprises historical vintages for the level of GDP starting with the fourth quarter of 1965 as first available vintage. Unfortunately, there exist no real-time data collections for the predictor variables. Therefore, this paper refrains from real-time forecasts but produces forecasts based on the latest reporting vintage which is the second quarter of 2019. The consequences of not working with real-time data need to be weighed accordingly.

The major consequence of not working with real-time data in forecasting tasks is related to the comparison of the performance of a forecasting model which has been used at different points in time. Due to data revisions, a researcher may claim that a certain model would have produced better results in the past that it actually did back then (Croushore & Stark, 2000). However, this may only be the case because she uses revised and therefore richer data than available in the past. In other words, vintages matter if one wants to compare models which have been produced at different points in time. Clearly, one way to approach this issue is to work with real-time data when training a forecasting model. Croushore and Stark (2000) show how forecasts are affected by the choice of data vintages. They compare forecasts based on the most recently available vintage with forecasts based on real-time data and find that for simple models such as autoregressive models the choice of vintage affects the forecasting results substantially. One important contribution of this paper is to compare different forecasting models which are all based on the same data source and the same vintage of data revision. Whether one takes real-time data or the latest available revised data is therefore in this context of minor concern.

Nonetheless, in the context of machine learning frameworks, another remark is needed on real-time data. Using training and test data sets, a perfectly accurate approach would look as follows: The training data, used to estimate the model and strictly preceding the test data, needs to be based on the latest vintage of data which has been available at the last date that is part of the training data set. Using a more recent vintage implies that training observations entail information that only became available at a time after the end of the training period. For the most honest assessment of the generalization performance of a time series model, this is problematic as the strict time-based demarcation between training and test data vanishes to some extent.

3 Methodology

3.1 Forecasting Strategy

The general methodology in this paper is to predict U.S. GDP growth in $t+h$ with h as forecasting horizon by the set of D distinct macroeconomic time series observed today in t as well as by their realizations in the past. U.S. GDP growth is defined as target variable $y \subset \mathbb{R}^1$. The term response variable is used as synonym. The set of D macroeconomic time series is referred to as feature space $\boldsymbol{x} = (x_1, \ldots, x_D) \in \boldsymbol{X} \subset \mathbb{R}^D$. The single components of the feature space x_d are termed as features. Predictors or explanatory variables are used synonymously. The final forecasts are generated through a set of approximating functions or models $f(\boldsymbol{x}; \Theta)$ which are parameterized by Θ

$$\hat{y}_{t+h} = f(\boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t-p}; \hat{\Theta}). \tag{2}$$

The scope of this paper is to introduce machine learning methods to the issue of macroeconomic forecasting. In order to evaluate how these methods perform in predicting U.S. GDP growth, it is paramount to benchmark the forecasting results against more commonly applied forecasting tools from the field of econometrics. Only in this way it is possible to judge whether machine learning can contribute or even outperform more standard methods.

Generally, econometric time series models represent stochastic processes while machine learning methods are more of an algorithmic nature. Comparing the probabilistic data modeling approach common in econometrics with the algorithmic approach common in machine learning, Breiman (2001b) speaks of two different 'cultures' which distinguish one from another in several dimensions. Outlining how these two cultures differ is important to understand the reason why machine learning can be useful for macroeconomic forecasting. Some of these dimensions in which traditional econometric models (as described in the following section) differ from machine learning methods (as described in section 3.3) are uncertainty, structure and objective (Harrell, 2019).

1. Uncertainty: Econometric models explicitly take uncertainty into account by assuming some form of probabilistic distribution for the residual term. This results in stochastic data models that allow to conduct both point forecasts as well interval forecasts, enabling the researcher to draw conclusions about the uncertainty of a forecasted value. Machine learning models, in contrast, are non-probabilistic and do not model uncertainty explicitly.

2. Structure: Econometricians select a parametric model which they believe describes the data generating process of the underlying data best. Typically, they assume additivity of predictor effects when specifying the model. Machine learning practitioners do not impose any preconceived structure on data but rather implement an algorithm which learns the relation between target variable and predictors in a data-driven manner (in contrast to a model-driven approach).

3. Objective: Machine learning algorithms are designed to achieve a high predictive accuracy in the first place. The focus in econometric models lies more on drawing conclusions from estimated model parameters. Certainly, it depends on the exact application which of the two, predictive accuracy or interpretability, is more important, but the former has usually an edge over the latter when it comes to forecasting.

The above mentioned differences between the two cultures of statistical modeling have two important consequences. First, if the econometric model imposed by the researcher is a poor description of how the data is generated, the conclusion drawn from the model estimates are inherently wrong. This problem is not faced by machine learning methods as they do not impose any structural model on the data. Second, machine learning techniques are often labeled as 'black box' (Breiman, 2001b, p. 199) methods which allow little to no room of interpretation with regards to the relation between the target variable and its predictors.[5] Econometric models are designed in a way that allows structural interpretation of the relationship between target and features *given* the data model is specified correctly.

It is important to have these characteristics in mind when applying different forecasting methods and especially when comparing their performance as done in this paper. The following highlights specific aspects of the forecasting strategy in more detail.

[5] Breiman (2001b), for instance, admits the black box nature of machine learning methods, but at the same time argues that accurate information and not interpretability matters most. This is a controversial statement as it strongly depends on the context and the research question how important the interpretation of the underlying methodology is.

3.1.1 Stationarity

Stationarity is an important concept in the context of producing forecasts based on time series models. Stationarity requires that the statistical properties of the process generating the time series do not change over time. This does not mean that the time series itself does not vary over time (otherwise there would be no reason to produce forecasts), but that the *way* how future realizations are generated does not change in the course of time (Palachy, 2019). Intuitively, it is not possible to forecast a time series if the data generating process according to which the series is distributed is time-dependent. In its weak form, stationary imposes that mean, variance and covariance of the time series are finite and time-independent. This assumption implies sufficient stability in the statistical properties of the time series which allows to model the data generating process. Extrapolating such a time series model then allows to produce multi-step ahead forecasts.

In contrast to time series models, machine learning methods do not model the data generating process (Harrell, 2019), but rather try to learn patterns encrypted in the data it is trained on. The primary goal of machine learning is to develop a model that generalizes beyond the data it has been trained on. In a time series context, this imposes restrictions on the data which are closely related to the concept of stationarity. While machine learning methods do not aim at modeling the data generating process itself, they clearly demand that the data generating process in the training set does not differ from the one in the test set. In a time series task, the training set needs to strictly precede the test set requiring some degree of stability in the way the series are generated. Only then machine learning methods can learn patterns on the training data which are useful for predictions on the test set. The easiest way to think about the necessity of stationarity-related restrictions in a machine learning context is to imagine a non-stationary trending time series. If realizations beyond the training data take on values which are far off the values the algorithm has seen during training, the machine learning method will clearly be unable to produce accurate predictions. A trending series which is non-stationary due to its time-dependent mean will exactly cause such a scenario. In this respect, stationarity is also an important concept in machine learning setups as it ensures that the data generating process does not fundamentally change beyond training data. This becomes particularly problematic in terms of trending series when the level of data changes systematically.

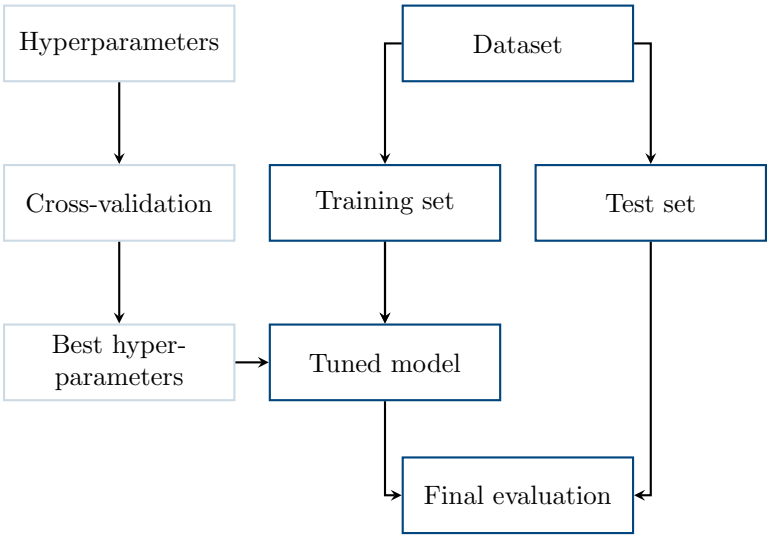
Given these considerations, it is crucial to turn non-stationary time series into stationary ones before feeding them into the models. The maintainers of FRED-QD suggest transformation codes applicable to the data to obtain stationary series (McCracken & Ng, 2016; J. Stock & Watson, 2012). The transformations can be roughly categorized. Real activity variables are recommended to be converted to quarterly growth rates (first differences of logs), prices and wages to quarterly changes of quarterly inflation (second differences of logs), interest rates to simple changes (first differences), and interest rate spreads are suggested to be kept in levels (J. Stock & Watson, 2012). The transformation recommendations strongly rely on differencing of the time series which is a common tool to turn non-stationary series into stationary ones. This paper follows the recommended transformations as first step to induce stationarity. After transforming the series, versions of the Augmented Dickey-Fuller (ADF) Test and of the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test are conducted to test whether the transformations have led to stationary series. In a first ADF-Test, the existence of a unit root is tested against a stationary Autoregressive process without intercept. A second specification of the ADF-Test incorporates an intercept term and tests again the existence of a Random Walk against a stationary AR process with constant. The third version of the ADF-Test incorporates both intercept term and a linear trend term. It then tests a stochastic trend, i.e. a Random Walk with drift, against a deterministic trend inducing trend stationarity. Since ADF-Tests are known to suffer low power (Verbeek, 2004), two different KPSS-Tests, one with a null hypothesis of level stationarity and the other one with a H_0 of trend stationarity, are conducted to validate the testing results. The number of lags in the stationarity tests is determined by the Bayesian Information Criterion (BIC). If any of the tests indicates that the respective series, despite its transformation, is still non-stationary at the 5% level, further visual inspections are conducted and alternative transformations are tested. The final transformations can be found in the third column of table 9 in the appendix.

3.1.2 Resampling Strategy

In this paper, nested cross-validation as model selection and model validation strategy is used. Basically, nested cross-validation is a combination of two loops of cross-validation. In an inner loop, cross-validation is used for calibrating the model's hyperparameters and features. In other words, the inner loop is responsible for tuning the machine learning models. Given the best set of hyperparameters and features

obtained from the inner loop, the outer loop uses cross-validation as a tool for evaluating the model's generalization performance on unseen data. Consequently, the inner loop aims at model building by means of parameter tuning and feature selection while the outer loop assesses the model's generalization performance (Varma & Simon, 2006). Generally, nested cross-validation requires three-way partitioning of the data into three independent sets: training, validation and test set. Figure 3 illustrates this typical machine learning workflow.

Figure 3: Machine Learning Framework



Note: Flowchart displays typical workflow in setting up a machine learning model. Nodes on the left highlighted in light blue indicate the tuning procedure on the training data (inner loop). Given the tuned model, the final performance of the model is assessed on the out-of-sample test set as displayed by the nodes in dark blue (outer loop). Flowchart is adapted from Pedregosa et al. (2019).

The outer loop first splits the data into a training set on the one hand and a test set on the other hand. The test set is strictly withheld from the model building process and only serves for assessing the model's generalization performance. Therefore, the test set is also referred to as out-of-sample-set. The assessment of a model's generalization capability using out-of-sample data is vital in every machine learning setting since most machine learning techniques tend to overfit the data they are trained on (James, Witten, Hastie, & Tibshirani, 2013). This means that fitting a model on training data tend to result in too complex models which are not capable of differentiating signal and noise in the underlying data.[6] As a consequence, their performance on training data tends to be very good but once applied on unseen data, they perform poorly because of their incapability of recognizing the signal entailed in new data.

The inner loop, responsible for model building, iteratively splits the overall training set into two further subsets - one called training subset and the other one called validation set. Similar to the test set in the outer loop, the validation set in the inner loop is kept aside when training the model. Using different hypeparameter constellations the model is then trained on the training subset and the performance of all different constellations is assessed on the left-out validation set. Clearly, this is computationally expensive as the set of possible hyperparameter cofigurations is likely to be infinite. Different strategies exist to approximate the best set of hyperparameters. Section 3.1.3 explains the exact tuning strategy used for this purpose. Once the best hyperparameter configuration is found in the inner loop, the model is trained on the full training set, and the tuned model's errors on the test set are recorded in the outer loop. The errors from the outer loop are then aggregated by specific error measures which are discussed in more detail in section 3.1.4.

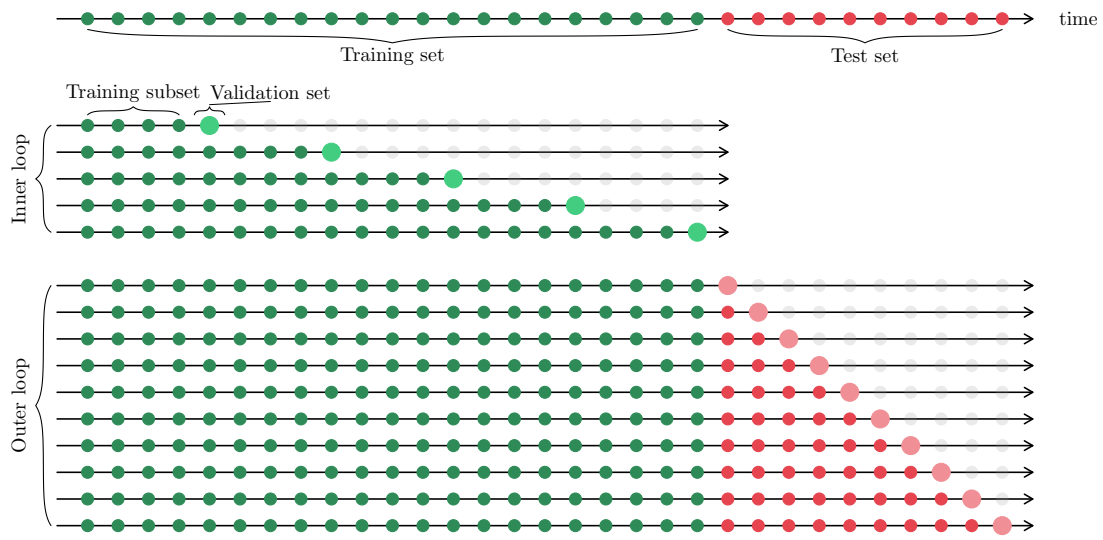
Generally, the models in this paper are used to make predictions in a time series context. This means that predictions must be understood as forecasts where the time ordering of data plays a crucial role. Particularly, the partitioning of data during cross-validation requires special attention. K-fold cross-validation as one of the most widely used partitioning strategies (Bergmeir, Hyndman, & Koo, 2018) splits the data randomly into k independent train and test sets (train and validation subsets in the inner loop). However,

[6] The same can be true for econometric time series models. Allowing too many lags in the model may yield good in-sample results but poor out-of-sample forecasts. For time series models, instead of cross-validation, information theoretic criteria are used in order to find the best lag structure. See also table 1 for more information.

for time series data this is no valid approach as the randomization does not preserve the time ordering of the underlying data.[7] Dividing data randomly into a training and test set bears the risk that realizations in the test set are older than the observations used for training the forecasting model. This is invalid from a forecasting perspective, since it implies that the past could be predicted by the future. For this reason, a special form of blocked cross-validation which splits the data in chronological blocks, strictly preserving the time structure, is applied in both loops. The partitioning strategy in blocked cross-validation for time series data respects that the test set (validation set) is always ahead of the training set (training subset). In other words, the test set needs to be always the last block in the cross-validation procedure. Unlike k-fold cross-validation which makes full use of the available data as each fold is used both for model training and model testing, in blocked cross-validation for time series data the latest block of data can never be used for training due to the time dependent logic of forecasting. This means that blocked cross-validation can never make use of the full data as efficiently as k-fold cross-validation does. However, it preserves the natural dependency of time series data, acknowledging that the future can only depend on the past and not vice versa (Bergmeir & Benítez, 2012).

The forecasting strategy in this paper conducts one-quarter ahead as well as one-year ahead forecasts following the rolling-origin-recalibration procedure described in Tashman (2000). Tashman (2000) defines the forecasting origin as the last available value from which on forecasting is performed. With a rolling origin strategy observations from the test set sequentially move to the training data. In each iteration step, the model is recalibrated according to the new (by one observation enlarged) training data.[8] One can think of this approach as an expanding-window strategy where the training data grows by one observation in each validation step. Alternatively, one could also use a rolling-window-recalibration strategy where, similar to the rolling-origin-recalibration procedure, data from the test set is sequentially transferred to the training set but the size of training data is kept constant by dropping one observation from the beginning of the series. This alternative recalibration strategy does not make a substantial difference in this paper's application. Figure 4 illustrates the cross-validation strategy applicable to time series data.

Figure 4: Time Series Resampling



Note: Figure shows the overall split into training and test data at the top. Training observations are shown as green dots, test observations as red dots. The split strictly obeys the time ordering and ensures that the training set precedes the test set. In the inner loop, training data is further split into blocks of training subsets and blocks of validation sets. Using an expanding-window strategy, model estimation with varying hyperparameter constellations is done on each of the training subsets. The performance of the forecasts (light green points) of all constellations is then compared with actual realization. The best performing hyperparameter combination is then used in the outer loop. Given the best hyperparameters, the tuned model's generalization performance is assessed on the training data. Using again an expanding-window strategy, the forecasting performance is assessed by comparing the forecasted value (light red dots) with the actual realization. Visualization is based on one-quarter ahead forecasts and is adapted from Hyndman (2016).

[7] See Bergmeir et al. (2018) for exceptions where k-fold cross-validation can still be applied to time series data.

[8] Note that in the inner loop cross-validation is not designed by single steps but by a larger step size in order to lower computational expense. Nonetheless, the time ordering is strictly preserved using blocked cross-validation.

3.1.3 Tuning Strategy

Part of the model selection process is the tuning of hyperparameters. In machine learning, hyperparameters, also referred to as free parameters, are not learned during training but fixed before the training process begins. While the optimal value of learner parameters results from the underlying optimization problem of the algorithm, the optimal set of hyperparameters needs to be found by testing the performance of different hyperparameter constellations on the independent validation sets. This process of testing different parameters to find the optimal hyperparameter combination is called hyperparameter tuning. The number and the nature of hyperparameters and learner parameters depend on the learning algorithm. For example, in Support Vector Regression using a sigmoid kernel there are four hyperparameters: ε which determines the width of the ε-tube within which deviations from the regression function are allowed, the cost parameter C which steers the trade off between model complexity and the degree to which deviations larger than ε are permitted as well as two further kernel parameters γ and c. The dual variables α_i and α_i^* as well as the intercept parameter b, in contrast, pose learner parameters in SVR which are learned during the training phase.[9]

When tuning an algorithm's hyperparameters, one needs to define a suitable search space S for each parameter first. The optimal value of a hyperparameter for a given learning algorithm highly depends on the underlying data set and the prediction task. Generally, it is little known about where to search for the optimal value of a parameter in advance (Bergstra & Bengio, 2012). Therefore, one usually needs to choose a relatively large search space to start the tuning procedure. Different tuning methods exist to find or approximate the optimal set of free parameters. A widely used method is grid search which exhaustively considers all possible hyperparameter combinations given the predefined search spaces. The advantage of grid search is that it always finds the optimal parameter constellation. However, the downside of grid search is that with large search spaces this becomes computationally highly expensive as grid search tests every single combination. In fact, the number of combinations grows overproportionally with the number of free parameters. In the above example, the number of trials conducted by grid search equals $(S_{size})^4$ with S_{size} as the size of the search space for each hyperparameter (here assumed to be equal for all four parameters). This over-proportional growth in the number of hyperparameters makes grid search suffering the curse of dimensionality (Bergstra & Bengio, 2012). Grid search is therefore highly unsuitable for searching large grids with many parameters since the computational expense becomes excessively large.

Given limited computational resources, this paper follows a two stage tuning strategy. In a first stage of tuning, it conducts a randomized search on the initially wide search spaces. Random search picks hyperparamter combinations randomly from the predefined search grid. The advantage of random search is that the user controls the number of trials by providing a maximum number of randomly chosen parameter constellations whose performance is tested. Consequently, the user can actively steer the computational expense. In contrast to grid search, random search usually does not find the optimal hyperparameter constellation but gets close enough with much lesser iterations (Bergstra & Bengio, 2012). In this sense, random search does not suffer the curse of dimensionality nor does it spend excessive time searching in ex ante unknown low interest areas where hyperparameter combinations perform poorly.[10] In this paper, the number of trials is limited to 100. With the random search results of first-stage tuning, the search space of each hyperparameter is narrowed in the following way. First, the top 30% parameter configurations are extracted from the random search results. From these high performing hyperparameter constellations, the (reasonably rounded) first quartile and third quartile of each hyperparameter form the basis of the new lower and upper bounds for the search spaces in the second stage of tuning.[11] This approach allows to narrow down the initially broad search spaces to regions where parameters perform well. The narrowed search spaces are searched again either by random search or by grid search. An overview of the initial search spaces and the narrowed search spaces in second-stage finetuning for each tuning parameter as well as information on the search methods can be found in table 1.

3.1.4 Forecast Accuracy Measurement

A further important aspect of a sound forecasting strategy lies in the principles of calculating the aggregate forecasting accuracy. Generally, accuracy measurements can be assigned to one of the following five groups: (i) scale-dependent measures, (ii) measures based on percentage errors, (iii) measures based on relative errors, (iv) relative measures and (v) measures based on scaled errors (Hyndman & Koehler,

[9] See section 3.3.3 for a detailed explanation of SVR.

[10] Bergstra and Bengio (2012) provide further details as to why random search is more efficient than grid search.

[11] It is ensured that the very best performing hyperparameter constellation from first-stage tuning is incorporated into the new bounds.

Table 1: Hyperparameter Search Spaces

Model	Hyperparameter	First-stage tuning search method	First-stage tuning lower bound	First-stage tuning upper bound	Second-stage finetuning search method	Second-stage finetuning lower bound	Second-stage finetuning upper bound
ARIMA	p	IC	0	10			
	q	IC	0	10			
VAR	p	IC	0	10			
FAVAR	V	CV (g.s.)	1	10			
	p	IC	0	10			
RF	M	CV (r.s.)	10^1	10^3	CV (g.s.)	50	550
	d_{try}	CV (r.s.)	21	210	CV (g.s.)	85	185
	$node_{min}$	CV (r.s.)	1	95	CV (g.s.)	85	95
GB	M	CV (r.s.)	10^1	10^3	CV (g.s.)	500	1000
	ν	CV (r.s.)	10^{-3}	10^{-1}	CV (g.s.)	0.01	0.06
	$depth_{max}$	CV (r.s.)	1	10	CV (g.s.)	9	10
SVR	C	CV (r.s.)	10^{-2}	10^4	CV (r.s.)	0.06	0.36
	ε	CV (r.s.)	10^{-5}	10^0	CV (r.s.)	0.0002	0.068

Note: In the column search method, the following abbreviations are used (i) IC: information criterion based parameter selection, (ii) CV (g.s.): parameter selection via a cross validated grid search, (iii) CV (r.s.): parameter selection via a cross validated randomized search.
In first-stage tuning of RF, the upper bound of d_{try} equals the total number of features. Note that this corresponds to Bagging. The lower bound of d_{try} is calculated as $0.1 \cdot$ (number of features). The upper bound of $node_{min}$ is calculated as $0.5 \cdot$ (number of training observations). This corresponds to very small trees (possibly stumps with only one binary split). The lower bound of $node_{min}$ is fixed at 1. This results in large trees with some leaf nodes including only one observation.
In first-stage tuning of GB, search spaces mainly mimic ranges used in previous research and recommended by literature (see James, Witten, Hastie, and Tibshirani (2013) for example).
In SVR polynomial, radial and sigmoid kernels are considered for tuning. Kernel parameters are not tuned but predefined using reasonable values ($w = 3$ and $\gamma = (\text{number of features})^{-1}$; see kernel definitions in section 4.1). In finetuning, only the sigmoid kernel as best performing kernel in first-stage tuning is further considered.
More details on the meaning of the individual hyperparameters can be found in the methodology section 3.3.
Second-stage finetuning bounds refer to the models developed for one-quarter ahead forecasts.

2006). While these classes of accuracy measures have very different properties, they all have in common that they are based on the out-of-sample forecast error. The forecast error as the unpredictable part of an observation is defined as the difference between the observed true value and its forecast

$$e_t = y_t - \hat{y}_t. \tag{3}$$

In an extended empirical assessment of annual and quarterly economic time series data, Armstrong and Collopy (1992) analyzed the suitability of different error measures based on four criteria: (i) reliability, (ii) construct validity, (iii) sensitivity and (iv) relationship to decisions. Generally, they distinguish between error measures used for calibrating a model (i.e. tuning the model's parameters) and error measures to assess the model's generalization performance. In the following, the use of specific error measures in this paper will be justified based on the above mentioned criteria.

For the purpose of model calibration, Armstrong and Collopy (1992) suggest using an error measure that is primarily characterized by a high degree of sensitivity. Sensitivity in this context means that the error measure should clearly indicate the impact on accuracy if one of the model's hyperparameters is changed. Moreover, sensitivity is related to how susceptible a measure reacts to outliers. It is important to mention that this is not necessarily desirable for error measures in the context of parameter tuning (Armstrong, 2001). In fact, Armstrong (2001) argues that low sensitivity to outliers is not desirable if the focus of the forecasting task lies on predicting abnormal cases such as wars, floods or hurricanes. This paper focuses on how well forecasting methods perform in times of economic crises. In a statistical sense, the effects of a crisis on an error measure can be seen as similar to the effects caused by outliers. Crises can be placed into the same group of abnormal cases which is why the building process of the model should result in parameters that allow the model to capture such cases. In other words, it is reasonable to choose an error measure which is sensitive to anomalies in the underlying data when tuning its hyperparameters. Error measures characterized by a high degree of sensitivity are the scale-dependent Root Mean Squared Error

(RMSE)

$$RMSE = \sqrt{mean(e_t^2)} \tag{4}$$

and the Mean Absolute Percentage Error (MAPE) which is based on percentage errors

$$MAPE = mean(|p_t|) \tag{5}$$

with $p_t = 100\frac{e_t}{y_t}$ (Armstrong & Collopy, 1992).

RMSE has the advantage of being on the same scale as the target variable which makes it easy to understand and interpret. At the same time, this is the deficiency of the RMSE as it does not allow to compare forecast performances across series with different scales (Hyndman & Koehler, 2006). The solution to this scale dependency are error measures based on percentage errors, such as the MAPE. However, the problem of MAPE is that it puts stronger weight on positive errors than on negative errors with negative errors being bound to a maximum of 100%. Since for the tuning of parameters the scale dependency of RMSE is irrelevant (tuning does not involve comparing forecast errors across different series), model calibration on the training set is based on RMSE in this paper. The asymmetric penalty of positive and negative errors in case of MAPE makes it an unsuitable candidate for this purpose.

For the assessment of generalization performance and the related comparison among forecasting models, Armstrong and Collopy (1992) highlight the importance of reliability, construct validity and relationship to decision making as main criteria for a sound error measure selection. Reliability means that the accuracy measure should rank different forecasting methods similarly if applied to different subsamples of the data. Construct validity assesses whether an error measure is in line with rankings among different forecasting methods produced by other error measures. This means that a valid error measure would rank the performance of opposing forecasting methods not entirely different as compared to other error measures. Ultimately, the relationship to decision making criteria favors measures which can be easily interpreted and which allow to draw direct conclusions for practitioners. Usually, error measures on the same scale as the target variable can be interpreted most easily and are thus most suitable for decision making (Armstrong & Collopy, 1992). According to the empirical study of Armstrong and Collopy (1992), none of the conventional error measures fulfills all of these criteria simultaneously. Nonetheless, the authors recommend the use of Median Relative Absolute Error (MdRAE) which, despite of being at a different scale as the target variable, is reliable and highly correlates with the rankings produced by other measures - a strong hint for construct validity. MdRAE is a measure based on relative errors. It is calculated by dividing each error by the error produced by another benchmark model

$$MdRAE = median(|r_t|) \tag{6}$$

with $r_t = \frac{e_t}{e_t^b}$. Note that e_t^b is the forecast error obtained from the benchmark model.

Besides the MdRAE which is based on relative errors, one can also assess generalization performance by means of relative measures (Hyndman & Koehler, 2006). For instance, if RMSE^b is the Root Mean Squared Error obtained from a benchmark model such as the Random Walk model, then the Relative RMSE (RelRMSE) is defined as

$$RelRMSE = \frac{RMSE}{RMSE^b}. \tag{7}$$

The RelRMSE can be easily interpreted. It tells by how many percent the proposed model is better or worse compared to the benchmark model. If RelRMSE is greater than one, it performs worse than the benchmark model, if it is smaller than one, it performs better. For the assessment of generalization performance, both MdRAE and RelRMSE are reported in this paper.

Moreover, this paper uses an extension of the Diebold-Marino (DM) Test in order to assess whether the difference in predictive accuracy between two models is statistically significant (Diebold & Mariano, 2002; Harvey, Leybourne, & Newbold, 1997). The test calculates the error differential series of the forecasts from the benchmark model and the alternative model. Under the null hypothesis that the two models have the same level of accuracy, the DM-Test statistic is asymptotically normal. The alternative hypothesis is that the accuracy between both models is significantly different. This allows to evaluate whether differences in the error measurements resulting from two different models are statistically significant. Detailed results of the DM-Test can be found in figure 8 of section 4.2.1.

3.2 Benchmark Econometric Models

This section introduces the basic concept of three time series models commonly used in macroeconomic forecasting. The models comprise an univariate Autoregressive model, Vector Autoregressive models and Factor-Augmented Vector Autoregressive models. The three types of models act on very different information sets and serve as benchmarking models for the machine learning methods.

3.2.1 Univariate Autoregressive Model

One of the simplest approaches in time series modeling is to exploit the target series' autocorrelative structure in order to make predictions of the series' future realizations. This means one analyzes how current realizations of the target series are related to its past realizations (Verbeek, 2004). The most prominent model of this univariate approach is known as Autoregressive Integrated Moving Average model. The formal introduction of ARIMA models goes back to Box and Jenkins (1970). Their influential work in the field of time series analysis, known as Box-Jenkins modeling, describes an iterative three stage procedure. Their framework originally comprises model selection, parameter estimation and model checking (Box, Jenkins, Reinsel, & Ljung, 2016). Hyndman and Athanasopoulos (2018) extend the Box-Jenkins framework to a seven-step approach including (i) data visualization and outlier detection, (ii) data transformation to stabilize variance, (iii) differencing to obtain stationary series, (iv) examination of Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF), (v) order selection using the corrected Akaike Information Criterion, (vi) testing residuals for white noise and (vii) the calculation of forecasts with the final model. This paper closely follows this seven-step approach.[12] While Autoregressive Moving Average (ARMA) models combine Autoregressive and Moving Average (MA) time series components, ARIMA additionally includes differencing of the time series as part of the model building process. Therefore, ARIMA models have, besides the AR order p and the MA order q, a third parameter which describes the order of differencing d. As described in section, 3.1.1 differencing is commonly used to stabilize the mean in order to obtain a stationary series (Hyndman & Athanasopoulos, 2018). In this paper, transformation of data with the goal to obtain stationary series is part of the data preparation process and is conducted before model building. Therefore, this paper focuses on ARMA models for linear univariate time series modeling. An ARMA(p, q) model is defined as follows

$$y_t = \theta_1 y_{t-1} + \ldots + \theta_p y_{t-p} + \epsilon_t + \phi_1 \epsilon_{t-1} + \ldots + \phi_q \epsilon_{t-q} \tag{8}$$

with ϵ_t as white noise component with zero mean and constant variance, i.e. $\epsilon_t \sim WN(0, \sigma^2)$.[13]

It becomes obvious that an ARMA(p, q) model uses both lagged values of the target variable and lagged variables of the white noise error term as predictors for the current realization of the target. Lagged values of the target variable form the AR part of the model; the error term and lagged realizations thereof build the MA part.

Alternatively, ARMA(p, q) models can be represented using the lag operator

$$\theta(L) y_t = \phi(L) \epsilon_t \tag{9}$$

with lag operater $L^p y_t = y_{t-p}$ and lag polynomial $\theta(L) = 1 - \theta_1 L - \ldots - \theta_p L^p$.

Given that the AR polynomial is invertible which it is if, and only if, the the target series is stationary, (Verbeek, 2004) equation (9) can be written as

$$y_t = \theta(L)^{-1} \phi(L) \epsilon_t. \tag{10}$$

Assuming normality of ϵ_t the $p + q$ parameters of the above ARMA specification are estimated using Maximum Likelihood (Hyndman & Khandakar, 2008). Given the estimated parameters and using all information available today, it is possible to produce forecasts by means of ARMA models. With the Mean Squared Error (MSE) as loss function, the expected value conditioned on today's set of information serves as optimal predictor of future realizations of y_t. A forecast of the target variable in $t + h$ can be expressed in the following way

$$\hat{y}_{t+h} = f(y_t, \ldots, y_{t-p+h}; \hat{\Theta}) = \mathrm{E}\left[y_{t+h} | \mathcal{I}_t\right] \tag{11}$$

[12] Note that steps (i) - (iii) are fundamental to all models and therefore part of the general data preparation as described in section 3.1.

[13] Note that in this section $y_t := y_t - \mu$ is the demeaned series of GDP growth with μ as the mean of the series y_t. This representation is chosen for notational convenience. The final model includes an explicit constant.

with information set $\mathcal{I}_t = \{y_1, \ldots, y_t, \epsilon_1, \ldots, \epsilon_t\}$.

The one-step ahead forecast for an ARMA(p, q) model is calculated as follows:

$$\hat{y}_{t+1} = \hat{\theta}_1 y_t + \ldots + \hat{\theta}_p y_{t-p+1} + \underbrace{\mathrm{E}\left[\epsilon_{t+1}\right]}_{=0} + \hat{\phi}_1 \epsilon_t + \ldots + \hat{\phi}_q \epsilon_{t-q+1}. \tag{12}$$

A point forecast in $t + h$ is calculated recursively using the forecasts from periods $t + 1, \ldots, t + h - 1$.

$$\hat{y}_{t+h} = \hat{\theta}_1 \hat{y}_{t+h-1} + \ldots + \hat{\theta}_{h-1} \hat{y}_{t+1} + \ldots + \hat{\theta}_p y_{t-p+h} + \underbrace{\mathrm{E}\left[\epsilon_{t+h}\right]}_{=0} + \ldots + \hat{\phi}_{h-1} \underbrace{\mathrm{E}\left[\epsilon_{t+1}\right]}_{=0} + \ldots + \hat{\phi}_q \epsilon_{t-q+h} \tag{13}$$

assuming that p and q are larger than h.

Despite the simplicity of ARMA models which only rely on the autocorrelation in the target variable and its error term, they tend to perform remarkably well in empirical economic studies compared to more elaborated structural models which incorporate the history of other economic variables as well (Verbeek, 2004). Therefore, ARMA models pose an important benchmark when comparing the forecasting accuracy of different methods. Another important benchmarking model in time series forecasts, the Random Walk model, can be derived from the above ARMA model.

Random Walk

From the ARMA family of models, one can obtain a special time series model which often serves as naïve benchmark in macroeconomic forecasting tasks (Fildes & Stekler, 2002). ARMA$(1, 0)$ with $\theta_1 = 1$ is defined as

$$y_t = y_{t-1} + \epsilon_t \tag{14}$$

and forms a Random Walk (RW) model. In a RW model, a forecast simply equals the current observation irrespective of the forecasting horizon:

$$\hat{y}_{t+h} = y_t \quad \forall\ h \geq 1. \tag{15}$$

According to former editor-in-chief of the *International Journal of Forecasting*, Rob J Hyndman, a policy of the journal is that every submitted method must be compared to standard benchmarks such as the RW model before the paper will even be considered for publication (Hyndman, 2010). In light of this minimum requirement, this paper compares forecasting performance among others against the forecasts obtained from a RW model.

3.2.2 Vector Autoregressive Model

A natural extension to univariate time series models that only depend on the history of the target variable are multivariate time series models which allow to model the relationship between several macroeconomic time series. Vector Autoregressive models pose a prominent multivariate extension of the univariate ARIMA model. From a forecasting perspective, the inclusion of other macroeconomic aggregates into the model means that the history of variables other than the target variable plays a role in producing forecasts for the target variable. Consequently, this approach operates on an extended set of information, possibly allowing to produce more accurate forecasts of the target variable (Verbeek, 2004). Moreover, VAR models allow to generate forecasts for all variables incorporated into the model. This is true because vector autoregressions are designed in a way that all variables are treated symmetrically. Changes in one variable affect all other variables in the system. So VAR models release the researcher from distinguishing between endogenous and exogenous variables since all variables are treated as endogenous (Verbeek, 2004). This is useful when the the interdependence of variables is a priori unknown as it is often the case in forecasting tasks with many possible interactions such as the one in this paper. A VAR(p) model with K endogenous variables and p lags is defined as follows

$$\boldsymbol{y}_t = \boldsymbol{a}_0 + A_1 \boldsymbol{y}_{t-1} + \ldots + A_p \boldsymbol{y}_{t-p} + \boldsymbol{\epsilon}_t \tag{16}$$

with $\boldsymbol{y}_t = (y_{1t}, \ldots, y_{kt}, \ldots, y_{Kt})$ as a vector of K preselected time series which enter the model, $\boldsymbol{a}_0$ as vector of intercept terms, A_j as $(K \times K)$ coefficient matrix for all $j = 1, \ldots, p$ lags and $\boldsymbol{\epsilon}_t$ as K-dimensional serially uncorrelated white noise process with $\mathrm{E}(\boldsymbol{\epsilon}_t) = \mathbf{0}$ and contemporaneous covariance

matrix $\mathrm{E}(\boldsymbol{\epsilon}_t\boldsymbol{\epsilon}_t') = \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$. Note that contemporaneous correlations among the white noise terms are allowed.

In lag operator notation, the VAR(p) can be written as

$$A(L)\boldsymbol{y}_t = \boldsymbol{a}_0 + \boldsymbol{\epsilon}_t \tag{17}$$

with $A(L) = I_K + A_1 L + \ldots + A_p L^p$.

Forecasting with VAR models works analogously to ARMA models. Forecasting the multivariate model in $t+h$ can be expressed in the following way

$$\hat{\boldsymbol{y}}_{t+h} = f(\boldsymbol{y}_t, \ldots, \boldsymbol{y}_{t-p+1}; \hat{\Theta}) = \mathrm{E}\left[\boldsymbol{y}_{t+h} | \mathcal{I}_t\right] \tag{18}$$

with information set $\mathcal{I}_t = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t, \boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_t\}$. Clearly, the information set in the VAR case is much richer as compared to the univariate ARIMA case. It comprises the complete history of all variables included in the vector $\boldsymbol{y}_t$ while the univariate model operates on the information entailed only in the history of the target variable.

The one-step ahead forecast for a VAR(p) model is obtained as follows:[14]

$$\hat{\boldsymbol{y}}_{t+1} = \hat{A}_1\boldsymbol{y}_t + \ldots + \hat{A}_p\boldsymbol{y}_{t-p+1} + \underbrace{\mathrm{E}\left[\boldsymbol{\epsilon}_{t+1}\right]}_{=\,0}. \tag{19}$$

A forecast in $t+h$ is calculated recursively using the forecasts from periods $t+1, \ldots, t+h-1$

$$\hat{\boldsymbol{y}}_{t+h} = \hat{A}_1\hat{\boldsymbol{y}}_{t+h-1} + \ldots + \hat{A}_{h-1}\hat{\boldsymbol{y}}_{t+1} + \ldots + \hat{A}_p\boldsymbol{y}_{t-p+1} + \underbrace{\mathrm{E}\left[\boldsymbol{\epsilon}_{t+h}\right]}_{=\,0} \tag{20}$$

assuming that p is larger than h.

3.2.3 Factor-Augmented Vector Autoregressive Model

Macroeconomic forecasting by means of factor models has been first applied by J. H. Stock and Watson (2002). They introduced the idea to tackle high dimensional data available for forecasting by means of a factor analysis which aims at compressing the high dimensional information contained in large feature spaces to a much smaller dimension. Their approach describes a two-step procedure where in the first step of the forecasting task the information of all possible predictors is pooled and only a handful of final predictors, i.e. the factors, are constructed from this pooled set of information (J. H. Stock & Watson, 2002). This approach allows to reduce the dimension of the feature space while still operating on all the information entailed in it. Unlike VAR models that require judgemental feature selection prior to estimating the model, the information set in factor models spans the complete feature space $\mathcal{I}_t = \{y_1, \ldots, y_t, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_t\}$. This means that forecasting by factors is not a feature selection tool but rather a way to construct a limited number of final factors as linear combinations of all D original features (James et al., 2013). In this sense, factor models are the first class of models introduced in this paper which can inherently cope with the high dimensional feature space of 210 distinct macroeconomic time series. In a second step of building a factor model, the estimated factors enter a standard VAR model which can then be used for forecasting GDP. In their benchmark paper, J. H. Stock and Watson (2002) do not forecast by means of VAR models but rather produce forecasts by regressing a target variable on the lags of the factors and the lags of the target variable itself which they refer to as dynamic factor model. This paper follows the approach in Bernanke, Boivin, and Eliasz (2005) who refine the dynamic factor model to the context of vector autoregressions. In fact, the terminology Factor-Augmented Vector Autoregressive appears in their paper for the first time.

The FAVAR approach is based on a VAR model in both the scalar target variable y_t and the $V \times 1$ vector of unobserved factors, $\boldsymbol{f}_t$, where $V < D$

$$\begin{bmatrix} \boldsymbol{f}_t \\ y_t \end{bmatrix} = A(L) \begin{bmatrix} \boldsymbol{f}_{t-1} \\ y_{t-1} \end{bmatrix} + \boldsymbol{\epsilon_t}. \tag{21}$$

Note that $A(L)$ is a conformable lag polynomial of finite order $p-1$ and $\boldsymbol{\epsilon}_t$ is a serially uncorrelated white noise process with $\mathrm{E}(\boldsymbol{\epsilon}_t) = \mathbf{0}$ and contemporaneous covariance matrix $\mathrm{E}(\boldsymbol{\epsilon}_t\boldsymbol{\epsilon}_t') = \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$.

[14] The subsequent representations are based on the demeaned series $\boldsymbol{y}_t := \boldsymbol{y}_t - \boldsymbol{a_0}$.

Clearly, equation (21) cannot be estimated directly since the latent factors cannot be observed. However, it is assumed that the factors can be constructed from the information contained in the feature space (Bernanke et al., 2005). More precisely, it is assumed that the high dimensional and high informational time series $\boldsymbol{x}_t$ are related to the latent factors $\boldsymbol{f}_t$ in the following form:[15]

$$\boldsymbol{x}_t = \Lambda \boldsymbol{f}_t + \boldsymbol{v}_t. \tag{22}$$

Λ is a $D \times V$ matrix of factor loadings and $\boldsymbol{v}_t$ has zero mean $\mathrm{E}(\boldsymbol{v}_t) = \mathbf{0}$ and is assumed to be either uncorrelated or alternatively exhibits a small amount of contemporaneous correlation.[16]

In this paper, Principal Component Analysis (PCA) is used to extract the V factors from the $D \times T$ feature space $\boldsymbol{X}$. The core idea of PCA is to reduce dimension of the feature space while retaining as much as possible of the variation present in the features. This is achieved by reducing the high dimensional feature space to only those directions with the most variability (James et al., 2013). One thereby transforms the feature space to a new set of uncorrelated predictors, i.e. the principal components, $\boldsymbol{c}_t$. These are sorted such that the first few capture most of the variation present in all of the original variables $x_1, \ldots, x_D$ (Dubey, 2018). At the same time, it is assumed that the directions in which the original variables show most variation are directions associated with the target variable (James et al., 2013).

This paper extracts the principal components from the feature space in the following way. First, the correlation matrix, $Corr(\boldsymbol{x})$, of the feature space is calculated. For the correlation matrix, one calculates the eigenvectors of the V largest eigenvalues. These eigenvectors form the columns of the loading matrix whereas the first column corresponds to the largest eigenvalue, the second column to the second largest eigenvector and so forth. The eigenvectors that belong to the V largest eigenvalues constitute the loading matrix. The principal components are then calculated as linear combinations of the original features with the factor loadings as weights

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{c}\Lambda' \\ \boldsymbol{c} &= \boldsymbol{x}\Lambda \end{aligned} \tag{23}$$

with $\Lambda'\Lambda = I$.

One of the key issues in PCA is to determine the number of principal components. The maximum number of components V is capped at $min(T-1, D)$. Clearly, it is desirable to extract a number of components much smaller than the number of original variables in the feature space in order to achieve a substantial dimension reduction. This implies $V \ll D$. Given the optimal number of principal components, they are used as reasonable proxies for the unobservable factors. Thus, the final factor estimate, $\tilde{\boldsymbol{f}}_t$, equals the principal components

$$\tilde{\boldsymbol{f}}_t = \boldsymbol{c}_t. \tag{24}$$

Ultimately, a VAR in y_t and $\tilde{\boldsymbol{f}}_t$ is estimated which is then used to produce forecasts in y_t

$$\begin{bmatrix} \widehat{\tilde{\boldsymbol{f}}}_{t+1} \\ \hat{y}_{t+1} \end{bmatrix} = f(\tilde{\boldsymbol{f}}_t, \ldots, \tilde{\boldsymbol{f}}_{t-p}, y_t, \ldots, y_{t-p}; \hat{\Theta}) = \hat{A}(L) \begin{bmatrix} \tilde{\boldsymbol{f}}_{t-1} \\ y_{t-1} \end{bmatrix}. \tag{25}$$

In the FAVAR setup described in this section, a technique has been introduced which 'officially' belongs to the field of machine learning. PCA is an unsupervised learning method (see for example Hastie, Tibshirani, and Friedman (2009), James et al. (2013)) which, in the context of this paper, is used to detect similarities among the many features in the high dimensional feature space and, by means of these similarities, aims at reducing the dimension of variables finally entering into the forecasting model. Nonetheless, the final forecasting model is a linear and additive time series model not different to a classical vector autoregression which is why the FAVAR framework is placed into the category of econometric models contrary to the more advanced supervised learning models which will be introduced in the subsequent section.

[15] Note that Bernanke et al. (2005) define factors to be forces which affect many economic variables including the observable target variable(s). The feature space is considered as a collection of 'informational' economic time series which allow to infer something about the factors. This assumption requires them to include the target variable(s) into equation (22) in order to obtain net effects of both the factors and the target variable(s) in the VAR equation (21) which is important for the structural analysis of the model. This paper refrains from such economic considerations as well as from structural interpretations. Instead, it takes a pure statistical approach which aims at effectively deriving a low dimensional set of features from the high dimensional feature space (not including the target variable) by means of PCA.

[16] See J. H. Stock and Watson (2002) for a discussion of the restrictions on cross-correlation in $\boldsymbol{v}_t$ under principal component estimation.

3.3 Machine Learning Algorithms

Most traditional forecasting in economic research has relied on probabilistic time series models of the kind introduced in the previous section. The latest volumes of the *Oxford Handbook of Economic Forecasting* as well as the *Handbook of Economic Forecasting*, both collections of state-of-the art surveys in the sphere of economic forecasting, published in 2011 and 2013, respectively, cover exclusively such models. While the former acknowledges the rise of richer and larger data sets and the availability of improved computational power in the field of economic research (Clements & Hendry, 2011), both handbooks' methodology section does not mention techniques related to machine learning but rather focuses on VAR, DSGE and factor modeling. Only more recently, economists have started to promote the usage of machine learning approaches to tackle economic forecasting issues as suggested by the literature review in section 1.2.

What follows in this section is an extension of the econometric forecasting toolbox by algorithmic approaches which are nowadays typically labeled as machine learning. There are several reasons why macroeconomic forecasting by means of machine learning algorithms offers promising extensions to more traditional time series models from the field of econometrics:

1. Existing macroeconomic forecasting models fail to yield accurate predictions in times of crisis. The failure of econometric models, especially in times of recession, poses the prime motivation to use more sophisticated machine learning methods for predicting economic activity.

2. Times of economic turmoil are characterized by highly nonlinear interactions among key macroeconomic variables. Clearly, if variables are related in a nonlinear fashion, any linear forecasting model tends to perform poorly. Machine learning methods are designed to capture nonlinearities and thus pose a promising class of models that potentially yield more accurate forecasts in economic crises.[17]

3. The doctrine in econometrics focuses on in-sample goodness-of-fit measures such as the all too often reported in-sample R^2 (Breiman, 2001b; Varian, 2014). Machine learning, in contrast, typically assesses model performance based on out-of-sample performance. In forecasting tasks, it is natural to focus on out-of-sample performance. One normally is interested in what a forecasting model predicts to happen in the future given data which has not been used to build the model. Consequently, best practices in machine learning performance assessment fit well in the context of forecasting.

4. In macroeconomic forecasting, one typically faces 'fat' data (Varian, 2014) characterized by a large number of predictors relative to the number of observations. Econometric models run into degrees of freedom problems or issues concerning overfitting when being confronted with this dimension of data. Factor analysis, introduced in section 3.2.3, which makes use of unsupervised learning techniques in order to tackle this problem, is one way to deal with 'fat' data. Also, supervised machine learning can handle large datasets with many variables. This is true because the number of parameters in machine learning models does not grow with the number of variables taken into consideration by the model.[18] Thus, the information set in all subsequent machine learning applications comprise the full feature space $\mathcal{I}_t = \{y_1, \dots, y_t, \boldsymbol{x}_1, \dots, \boldsymbol{x}_t\}$.

In the following, three specific machine learning models, Support Vector Regression, Random Forest and Gradient Boosting are explained in greater detail. The author has chosen to implement these models as they tend to belong to the most widely used group of machine learning methods with most promising results in different fields of research and business. The subsequent mathematical formulations refrain from using time indices as these models do not have the time structure of the data inherent. In order to produce forecasts, all of the models follow a dynamic regression approach with lagged predictors as right-hand variables. The modeling approach takes on the following form:

$$y_t = f\left(y_{(t-h)}, \dots, y_{(t-h)-p}, \boldsymbol{x}_{(t-h)}, \dots, \boldsymbol{x}_{(t-h)-p}; \hat{\Theta}\right). \tag{26}$$

It is important to mention that with this approach only one-step forecasts can be made. However, the step size can be altered by the choice of h. A one-quarter ahead forecast, for example, implies $h = 1$. A one-year ahead forecast, in turn, necessitates that only variables of at least lag 4 enter the right-hand side of the model which implies $h = 4$. Since lags of the variables enter the model as separate features, each additional lag in the model decreases the dataset by one sample while at the same time increasing it by D additional features (one lagging variable for each of the D distinct features).

[17] Clearly, nonlinearity is not exclusive to machine learning methods. There are also nonlinear models in time series econometrics. See Kock and Teräsvirta (2011) for a review.

[18] Pure econometric models, in contrast, usually add at least one parameter for each additional variable incorporated into the model.

3.3.1 Random Forest

Random Forest is an ensemble learning method which has been formally introduced by Breiman (2001a). RF is an interesting candidate for the purpose of GDP forecasting as it allows, by means of a variable importance measure, to analyze the relevance of different features in producing forecasts. In this way, the algorithm is the first approach in this paper that makes it possible to incorporate all 210 macroeconomic time series in the task of forecasting GDP growth, while at the same time giving an idea which of these 210 features have especially strong power in predicting future GDP growth. RF belongs to the class of ensemble methods because it is built from a collection of simple decision trees. Decision trees can be understood as a series of binary decisions assigning each training period to a final tree leaf. Periods falling in the same terminal leaf get the same forecasting values assigned while the algorithms is designed in a way that only periods with similar patterns belong to the same terminal leaves. In a regression context, decision trees are also referred to as regression trees. In the following the concept of regression trees is explained in more detail. Finally, the concept is expanded to Random Forests.

Regression Trees

The idea of regression trees can be thought of as a two-step approach. First, the training observations are split into J non-overlapping regions $R_1, \ldots, R_J$ by means of the predictor variables $x_1, \ldots, x_D$. In a second step, for every region, one makes the same prediction for each variable falling into the respective region. The prediction of observations falling into region R_j simply equals the mean $\hat{y}_{R_j}$ of the target variable of all observations in R_j. In theory, the regions could have any shape but, for the sake of interpretability, the regions are restricted to high dimensional rectangles. Mathematically this translates into an optimization problem which aims at finding the rectangles $R_1, \ldots, R_J$ which minimize the Residual Sum of Squares (RSS) loss function (James et al., 2013):

$$\min_{R_1,\ldots,R_J} = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2. \tag{27}$$

The issue with regard to optimization problem (27) is that it is computationally not feasible to consider all possible partitions of the feature space into J non-overlapping rectangles. Therefore, one follows a top-down recursive binary splitting approach (James et al., 2013). Instead of splitting into J regions at once, this approach sequentially splits the feature space into two new branches. More precisely, when building a regression tree based on binary splits one starts off with all training observations at the top of tree, also called the root node. Starting from the root node one splits the data into two branches based on a specific variable x_d from the feature space and a respective splitting point s. The choice of the splitting variable and the splitting point is based on the improvement in the Residual Sum of Squares resulting from a further split of the data given the respective splitting variable. This means that the algorithm compares for each predictor variable (and a set of splitting points in the domain of the predictor) the RSS before and after the split. The algorithm selects the predictor variable and splitting point which yields the greatest improvement in RSS. The two resulting child nodes can be defined as follows (Hastie et al., 2009):

$$R_1(x_d, s) = \{\boldsymbol{x} | x_d \leq s\} \tag{28}$$

$$R_2(x_d, s) = \{\boldsymbol{x} | x_d > s\}. \tag{29}$$

Given the next optimal predictor variable and its respective optimal splitting point, the two resulting child nodes are split again into further subbranches leading to a second level of childnodes. This binary splitting process is continued until a certain stopping criteria is met. Final nodes at the end of the tree are called leaf nodes. Each leaf node is assigned a prediction value $\hat{y}_{R_j}$ based on the training periods falling into the respective leaf. With the binary splitting approach, the above optimization problem simplifies at each split to[19]

$$\min_{x_d, s} \left(\min_{\hat{y}_{R_1}} \sum_{i \in R_1(x_d,s)} (y_i - \hat{y}_{R_1})^2 + \min_{\hat{y}_{R_2}} \sum_{i \in R_2(x_d,s)} (y_i - \hat{y}_{R_2})^2 \right). \tag{30}$$

It can be shown that for any choice of x_d and s in the outer minimization, the inner minimization is solved by exactly the mean value of the target variable of those observations falling into region R_1 and

[19] Note that in the forecasting context of this paper a combination of features and target $(\boldsymbol{x}_i, y_i)$ corresponds to $(\boldsymbol{x}_{(t-h)}, y_t)$.

R_2, respectively (Hastie et al., 2009).

In this paper the stopping criteria signaling that the tree is not supposed to be grown further is the minimum number of observations in each leaf node, $node_{min}$. This means that the splitting procedure is stopped once a further binary split would lead to a leaf node including less than $node_{min}$ observations. Note that $node_{min}$ is a tunable parameter which steers the size of the tree, i.e. the number of leaf nodes J. A large value of $node_{min}$ leads to rather small trees, while a small value for $node_{min}$ leads to larger, more complex trees. The final regression tree model, partitioning the feature space into J leaf nodes, can be represented as follows

$$T(\boldsymbol{x};\Theta) = \sum_{j=1}^{J} \hat{y}_{R_j} \mathbb{1}(\boldsymbol{x} \in R_j). \tag{31}$$

Ensemble of Regression Trees: Random Forest

Generally, regression trees are known to suffer from high variance. Therefore, applying trees to different samples of the same data can lead to quite different results (James et al., 2013). A natural way to lower the variance and thus to increase prediction accuracy of a regression tree is to build a repeated number of randomized trees and average the resulting predictions. Consequently, the extension of simple regression trees to Random Forests is no more complicated than averaging the results from M randomized trees

$$f(\boldsymbol{x};\Theta) = \frac{1}{M} \sum_{m=1}^{M} T(\boldsymbol{x};\Theta_m) \tag{32}$$

with $\Theta_m = \{R_{jm}, \hat{y}_{R_{jm}}\}_1^{J_m}$.

This means that instead of building only one tree, one estimates several distinct trees and averages the results from all trees for each observation. Generally, the randomization of trees is conducted in two ways: building trees on different samples of the training data on the one hand and, on the other hand, considering only a feature subset as splitting candidates at each node.

The first randomization approach uses bootstrapping as resampling method in order to generate different subsamples of training data. This method is also known as Bagging (James et al., 2013).
A second source of randomization in the process of building trees allows to consider only a random selection of all predictor variables as possible splitting variables at each split node. This method of randomization is known as random subspace method and has first been proposed by Ho (1995). The advantage of random subspacing is that it greatly decorrelates the resulting decision trees. The number of predictor variables which is randomly sampled at each node is defined as $d_{try} \leq D$. Note that d_{try} is another tuning parameter in the RF algorithm.

Breiman (2001a) claims that RF does not overfit with an increasing number of trees. While this gives reason to simply choose M to be sufficiently large to ensure that the training error rate has settled down at a sufficiently low level, this paper still treats M as hyperparameter since larger values of M comes at computational expense.

3.3.2 Gradient Boosting

Gradient Boosting is often seen as *the* state-of-the art machine learning method to tackle data mining issues in different fields of application. In fact, GB counts as the most successful machine learning technique on *Kaggle*, the leading online community of Data Scientists (Chen & Guestrin, 2016). The dominant use of GB among machine learning practitioners makes it a natural candidate for the application of forecasting GDP growth.

The idea of boosting has originally been developed to turn weak learners into strong learners and goes back to the contributions of Friedman (2001). Weak or base learners can be understood as any learning method which is at least slightly better than random guessing (Freund & Schapire, 1997). In the context of this paper, single decision trees are used as base learners. The idea of gradient boosting trees is to develop a number of trees in a sequential fashion. This means that in each step of the algorithm a new tree is built using information from the trees developed in the steps before. More precisely, every new

decision tree fits the training errors resulting from a composite tree model developed in the steps before.

Similar to Random Forests, boosting is another ensemble machine learning method. However, while in Random Forest trees are built independently from each other, each regression tree in boosting highly depends on what has happened in the algorithm before.[20] The core of boosting lies in its sequential nature where an initial weak learner becomes stronger with each iteration since a new tree is trained with respect to the in-sample error of the ensemble that has so far been developed in the algorithm. Thus, adding a new tree improves the composite model in each step of the algorithm. In other words, each step boosts the model. In that sense, the final boosted tree model is an additively connected *sequence* of dependent trees

$$f(\boldsymbol{x};\Theta) = \sum_{m=1}^{M} T(\boldsymbol{x};\Theta_m). \tag{33}$$

Typically, a boosted regression tree model is estimated by minimizing a specific loss function with respect to a given set of parameters. In case of decision trees, this set of parameters comprises the splitting variable and the splitting point at each node defining the splitting regions R_j and the constant $\hat{y}_{R_j}$ assigned to each region, $\Theta_m = \{R_{jm}, \hat{y}_{R_{jm}}\}_1^{J_m}$. J_m is the number of terminal leaf nodes which controls the tree size and is constrained to be equal for each tree. In boosting, J_m is also referred as the interaction depth that specifies the maximum depth of each tree, $depth_{max}$ (i.e., the highest level of variable interactions allowed). Note that the interaction depth is a tunable hyperparameter in the boosting algorithm. In a boosting model, there are M different trees all of whose parameters ideally need to be considered in the model estimation. Similar to RF, M is treated as hyperparameter with the important difference that boosting can overfit if M is too large (James et al., 2013). The resulting optimization problem looks as follows[21]

$$\hat{\Theta} = \underset{\Theta_1,\ldots,\Theta_M}{\arg\min} \sum_{i=1}^{N} L\left(y_i, \sum_{m=1}^{M} T(\boldsymbol{x}_i;\Theta_m)\right). \tag{34}$$

For most loss functions, this optimization problem requires highly complex numerical techniques which are computationally not feasible (Hastie et al., 2009).
Note that the estimation of $\Theta = \{\{R_{j1}, \gamma_{j1}\}_1^{J_1}, \ldots, \{R_{jm}, \gamma_{jm}\}_1^{J_m}\}$ would require $2 \cdot M \cdot J$ parameters to be determined simultaneously.

One solution to this problem is an approximation of the above global solution by means of a forward stagewise additive modeling approach (Hastie et al., 2009). This strategy sequentially adds a new decision tree to the expansion in each step of the algorithm and only optimizes the parameters of the new tree, leaving the parameters of the previously added trees unmodified. This simplifies the optimization procedure as in each step only the parameters of a single tree need to be estimated instead of all parameters of M trees simultaneously. At each stage $m = 1, \ldots, M$ the optimization problem boils down to

$$\hat{\Theta}_m = \underset{\Theta_m}{\arg\min} \sum_{i=1}^{N} L\left(y_i, f_{m-1}(\boldsymbol{x}_i) + \nu T(\boldsymbol{x}_i;\Theta_m)\right) \tag{35}$$

with $f_{m-1}(\boldsymbol{x}) = f_{m-2}(\boldsymbol{x}) + \nu T(\boldsymbol{x};\hat{\Theta}_{m-1}) := \hat{y}_{m-1}$.

In GB, ν is the rate of learning determining how strongly new training errors from the prior step in the stagewise algorithm are corrected by the new tree. Typically, ν is treated as hyperparameter steering the speed of learning (Hastie et al., 2009). Note that Θ_m can be estimated using the top-down recursive partitioning strategy outlined in equation (30).

[20] This distinction implies that parallel computing is not possible in the case of GB. In RF, however, the estimation of trees can be carried out simultaneously on multiple CPUs. Generally, the author uses parallelization techniques to accelerate execution time where possible. Furthermore, trees in boosting are not developed on a bootstrapped subsample of the training data as it is the case in Random Forests. So there is no bootstrapping element in boosting which is another contrast to RF (James et al., 2013).

[21] Note that in the forecasting context of this paper, a pair $(\boldsymbol{x}_i, y_i)$ corresponds to $\left(\boldsymbol{x}_{(t-h)}, y_t\right)$. Similarly, N refers to $\bar{t} - h$ with $\bar{t}$ as the last observation in the training set. For the sake of readability, the following formulations refrain from the use of the exact time indices.

This paper takes the MSE as loss function resulting in the following optimization specification

$$\hat{\Theta}_m = \arg\min_{\Theta_m} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - f_{m-1}(\boldsymbol{x}_i) - \nu T(\boldsymbol{x}_i; \Theta_m)\right)^2 . \tag{36}$$

From optimization problem (36) it becomes obvious that in case of MSE as loss function, the new tree fits the residual vector $res_{m-1} = y - \hat{y}_{m-1}$ which equals the difference between true values and the predictions from the expansion model in the prior step

$$\hat{\Theta}_m = \arg\min_{\Theta_m} \frac{1}{N} \sum_{i=1}^{N} \left(res_{m-1} - \nu T(\boldsymbol{x}_i; \Theta_m)\right)^2 . \tag{37}$$

For loss functions other than the MSE, the optimization problem does not reveal an obvious residual vector in the objective function. This is where the concept of gradient descent and the invention of *Gradient* Boosting in the benchmark paper of Friedman (2001) comes into play. The negative of a loss function's gradient, $-g_{m-1}$, indicates the steepest direction on the loss function given the current location. In other words, it determines the direction of the fastest way on the loss function leading to its minimum. Therefore, one can understand the (negative) gradient as a direction vector in a numerical minimization procedure which forms the basic solution mechanism in Friedman's (2001) paper.

It is possible to calculate the gradient for every differentiable loss function given the current position on the loss function. The current location on the loss function is determined by the predictions of the composite model in the latest step of the boosting algorithm. The gradient can then be calculated as the vector of partial derivatives with respect to the current predictions of all training observations

$$g_{m-1} = \frac{\delta L\big(y, f_{m-1}(\boldsymbol{x})\big)}{\delta f_{m-1}(\boldsymbol{x})}. \tag{38}$$

From (37), one can see that, indeed, for MSE as loss function the gradient boosting algorithm simplifies to fitting a single tree to the residual vector resulting from the prior expansion model. This holds for each of the steps of the boosting algorithm:

$$\begin{aligned} g_{m-1} &= \frac{1}{2} \frac{\delta (y - f_{m-1}(\boldsymbol{x}))^2}{\delta f_{m-1}(\boldsymbol{x})} \\ &= -\big(\underbrace{y - f_{m-1}(\boldsymbol{x})}_{res_{m-1}}\big). \end{aligned} \tag{39}$$

This means that one can interpret residuals as negative gradients. For loss functions other than the square loss, this analogy does not hold. Therefore, the concept of gradients (which exists for every loss function) is generally more useful than the concept of residuals (Li, 2016). In other words, Gradient Boosting can be understood as combination of gradient descent optimization and model boosting.

3.3.3 Support Vector Regression

The groundwork for the development of Support Vector Regression goes back to Vapnik and Chervonenkis and their contributions in the field of statistical learning (Smola & Schölkopf, 2004). Their work, today known as *VC*-theory, resulted in one of the most widely used classification algorithms: the Support Vector Machine. Support Vector Regression can be understood as a generalization of Support Vector Machines for regression tasks. The good performance of Support Vector Regression on predicting time series data, albeit mostly applied on micro financial time series (see for example Müller et al. (1997) and Crone, Hibon, and Nikolopoulos (2011)), makes it an interesting candidate for an application on macroeconomic time series data. Smola and Schölkopf (2004) give an excellent overview of the technical details related to the estimation of SVR algorithms. In line with their paper, the following explanations provide more details on the statistical learning algorithm in SVR.

Unlike SVM which aim at classifying labeled data in a high dimensional space by means of a nonlinear hyperplane, SVR follows algorithmically a similar logic but with the ultimate goal to estimate a nonlinear real-valued function whose realizations allow to predict the target variable given the high dimensional input space, $\boldsymbol{X}$. The explanation of SVR in this paper starts with the introduction of linear SVR and

then expands to nonlinear SVR by incorporating mapping functions and kernels. In the linear case, the above mentioned real-valued function has the following form

$$f(\boldsymbol{x};\Theta) = \boldsymbol{\beta}\boldsymbol{x} + b \tag{40}$$

where $\boldsymbol{\beta}$ is $(1 \times D)$ parameter vector and b a scalar.

Following the ε-SV regression approach in Vapnik (2013), the estimation procedure of $f(\boldsymbol{x};\Theta)$ is designed to fulfill two essential criteria. First, for all training data points $\boldsymbol{x}_i$, $f(\boldsymbol{x};\Theta)$ is not allowed to deviate more than a fixed value ε from y_i, introducing a tube around $f(\boldsymbol{x};\Theta)$.[22] Deviations of y_i from $f(\boldsymbol{x}_i)$ are only allowed within the boundaries of this ε-tube. Second, $f(\boldsymbol{x};\Theta)$ is supposed to be as flat as possible. Flatness of $f(\boldsymbol{x};\Theta)$ means that $\boldsymbol{\beta}$ is chosen to be small. This prevents the model from becoming overly complex (i.e. from overfitting) and therefore useless for generalization purposes. One way to ensure flatness of $f(\boldsymbol{x};\Theta)$ is to minimize the norm value $\boldsymbol{\beta}'\boldsymbol{\beta}$ (Smola & Schölkopf, 2004). This results in the following convex optimization problem:

$$\begin{aligned} \hat{\Theta} \quad &= \quad \underset{\boldsymbol{\beta},\ b}{\arg\min}\ \frac{1}{2}\boldsymbol{\beta}'\boldsymbol{\beta} \\ s.t. \quad & \quad y_i - (\boldsymbol{\beta}\boldsymbol{x}_i + b) \leq \varepsilon \quad \forall\ i \\ & \quad (\boldsymbol{\beta}\boldsymbol{x}_i + b) - y_i \leq \varepsilon \quad \forall\ i. \end{aligned} \tag{41}$$

Note that a function which fulfills the constraint for all points in the training data does not always exist. If this is the case, the above optimization problem is not feasible. The solution to this scenario is the introduction of slack variables, ξ_i and ξ_i^*. Slack variables allow observations to be located outside the ε-tube by exactly the size of ξ_i or ξ_i^*, respectively. Training points 'above' the ε-tube have values $\xi_i > 0$ and $\xi_i^* = 0$, training points 'below' the ε-tube are assigned $\xi_i = 0$ and $\xi_i^* > 0$, while observations within the tube are characterized by $\xi_i = 0$ and $\xi_i^* = 0$.[23] The optimization problem known as primal objective function then becomes:

$$\begin{aligned} \hat{\Theta} \quad &= \quad \underset{\boldsymbol{\beta},\ b}{\arg\min}\ \frac{1}{2}\boldsymbol{\beta}'\boldsymbol{\beta} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ s.t. \quad & \quad y_i - (\boldsymbol{\beta}\boldsymbol{x}_i + b) \leq \varepsilon + \xi_i \qquad \forall\ i \\ & \quad (\boldsymbol{\beta}\boldsymbol{x}_i + b) - y_i \leq \varepsilon + \xi_i^* \qquad \forall\ i \\ & \quad \xi_i, \xi_i^* \geq 0 \qquad \forall\ i. \end{aligned} \tag{42}$$

The hyperparameter C regulates how strong deviations larger than ε are penalized in the minimization problem. It balances the trade-off between flatness of $f(\boldsymbol{x};\Theta)$ and the amount up to which deviations from the ε-environment are permitted. In the fashion of L_2 regularization used in ridge regression, it allows to steer model complexity. High values of C lead to a rather flat function since deviations outside the ε-tube are strongly penalized, while low values of C increase model complexity bearing the risk of overfitting the training data.

The corresponding loss function which only penalizes observations outside the ε-environment is referred to as ε-insensitive loss function. It is defined as follows:

$$L_\varepsilon := \begin{cases} 0 & \text{if } |y - f(\boldsymbol{x};\Theta)| \leq \varepsilon \\ |y - f(\boldsymbol{x};\Theta)| - \varepsilon & \text{else.} \end{cases} \tag{43}$$

Optimization problem (42) can be stated as a Lagrange optimization by introducing nonnegative La-

[22] Note that in the forecasting context of this paper, a pair $(\boldsymbol{x}_i, y_i)$ corresponds to $\left(\boldsymbol{x}_{(t-h)}, y_t\right)$. Similarly, N refers to $\bar{t} - h$ with $\bar{t}$ as the last observation in the training set. For the sake of readability, the following formulations refrain from the use of the exact time indices.

[23] Note that one could also have introduced only one slack variable defined as the absolute value of the difference between $f(x)$ and y. However, the use of two slack variables makes later steps in the derivation of the SVR algorithm less cumbersome.

grangian multipliers α_i and α_i^* as well as η_i and η_i^*:

$$\begin{aligned}\mathcal{L} = \frac{1}{2}\boldsymbol{\beta}'\boldsymbol{\beta} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ - \sum_{i=1}^{N}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ - \sum_{i=1}^{N}\alpha_i(\varepsilon + \xi_i - y_i + \boldsymbol{\beta}\boldsymbol{x}_i + b) \\ - \sum_{i=1}^{N}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \boldsymbol{\beta}\boldsymbol{x}_i - b).\end{aligned} \tag{44}$$

It can be shown that the primal objective function has a saddle point at the optimal set of primal variables $\boldsymbol{\beta}$, b and $\xi_i^{(*)}$ and dual variables $\alpha_i^{(*)}$ and $\eta_i^{(*)}$ (Smola & Schölkopf, 2004).[24] From the saddle point property, it follows that the first partial derivatives of the Lagrange function (with respect to the primal variables) need to be zero at optimality.

$$\frac{\partial}{\partial b}\mathcal{L} = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i) = 0 \tag{45}$$

$$\frac{\partial}{\partial \boldsymbol{\beta}}\mathcal{L} = \boldsymbol{\beta} - \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\boldsymbol{x}_i = 0 \tag{46}$$

$$\frac{\partial}{\partial \xi_i^{(*)}}\mathcal{L} = C - \eta_i^{(*)} - \alpha_i^{(*)} = 0 \tag{47}$$

Substituting (45), (46) and (47) into the Lagrange function results in an alternative representation of the objective function, the so-called dual objective function that can be solved using standard quadratic programming (Wang, Xu, Lu, & Zhang, 2003):

$$\begin{aligned}\hat{\Theta} = & \underset{\alpha,\ \alpha^*}{\arg\max}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\boldsymbol{x}_i'\boldsymbol{x}_j \\ & - \varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N}y_i(\alpha_i - \alpha_i^*) \\ s.t. \quad & \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \\ & 0 \leq \alpha_i^{(*)} \leq C \quad \forall\ i.\end{aligned} \tag{48}$$

Unlike the primal optimization problem, the dual formula is represented as inner product of the training data $\boldsymbol{x}_i'\boldsymbol{x}_j$. In the later nonlinear extension, this dot product representation turns out to be particularly useful. The substitution of the partial derivatives eliminates not just the primal variables but also the dual variables η_i and η_i^*. As a result, the dual optimization problem depends only on α_i and α_i^*.[25]

From equation (46), it results that the parameter vector $\boldsymbol{\beta}$ can be fully described as linear combination of the training observations

$$\boldsymbol{\beta} = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\boldsymbol{x}_i. \tag{49}$$

It follows immediately that the linear SVR function (40) is defined as

$$f(\boldsymbol{x};\Theta) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\boldsymbol{x}_i'\boldsymbol{x} + b. \tag{50}$$

[24] Note that in the remainder of this section, $\xi_i^{(*)}$ refers to both ξ_i and ξ_i^*. The same applies to $\alpha_i^{(*)}$ and $\eta_i^{(*)}$.

[25] Note that the estimation of b is not incorporated into the dual objective function (48). However, for the full specification of $f(\boldsymbol{x};\Theta)$, its computation is required as well. While Smola and Schölkopf (2004) derive an upper and lower bound for b, Keerthi, Shevade, Bhattacharyya, and Murthy (2001) outline further ways of specifying the constant term b. The interested reader is referred to these papers.

The SVR function (50) reveals an important property of SVR. It shows that the prediction of new data points does not require the estimation of the parameter vector $\boldsymbol{\beta} \in \mathbb{R}^D$. This means that $f(\boldsymbol{x}; \Theta)$ is independent of the dimensionality of the feature space $\boldsymbol{X}$ but only depends on the number of support vectors. The number of support vectors required to describe $\boldsymbol{\beta}$ can be analyzed further by means of the Karush–Kuhn–Tucker (KKT) conditions (Karush, 2013; Kuhn & Tucker, 1951). Generally, solutions of optimization problems with *inequality* constraints such as in the case of SVR need to fulfill the KKT conditions to be considered as optimal (Smola & Schölkopf, 2004). One of these conditions is the complementary slackness condition which demands that at an extremum or saddle point, the products between dual variables and constraints need to equal zero:

$$\begin{aligned} \alpha_i(\varepsilon + \xi_i - y_i + \boldsymbol{\beta}\boldsymbol{x}_i + b) &= 0 \\ \alpha_i^*(\varepsilon + \xi_i^* + y_i - \boldsymbol{\beta}\boldsymbol{x}_i - b) &= 0 \\ \underbrace{(C - \alpha_i)}_{\eta_i}\xi_i &= 0 \end{aligned} \tag{51}$$

$$\underbrace{(C - \alpha_i^*)}_{\eta_i^*}\xi_i^* = 0. \tag{52}$$

From KKT conditions (51), it can be seen that both α_i and α_i^* need to be zero for training observations that are strictly located within the ε-tube. For such observations, the second factor in both equations of (51) is strictly nonzero such that α_i and α_i^* have to equal zero in order for the complementary slackness condition to hold. This means that all observations inside the ε-environment disappear in equation (50). For observations which lie outside the tube either α_i or α_i^* is nonzero. These training observations - located outside the ε-tube - are called Support Vectors. This means that the number of parameters in SVR equals the number of Support Vectors and does not depend on the dimension of the feature space $\boldsymbol{X}$. For this reason, SVR is considered as parsimonious machine learning method (Vapnik, Golowich, & Smola, 1996).[26] Introducing a set $\mathbb{S}$ which only includes observations located outside the ε-tube and therefore acting as Support Vectors, the SVR function can be defined as

$$f(\boldsymbol{x}; \Theta) = \sum_{i \in \mathbb{S}} (\alpha_i - \alpha_i^*)\boldsymbol{x}_i'\boldsymbol{x} + b. \tag{53}$$

It is also important to mention that SVR models only depend on *inner products*, $\langle \boldsymbol{x}_i, \boldsymbol{x} \rangle = \boldsymbol{x}_i'\boldsymbol{x} = \sum_{d=1}^{D} x_{id}x_d$, between Support Vectors and the new observation of interest. So specifying equation (53) only requires these inner products and not the observations themselves. Clearly, one needs to know the observations in order to calculate the inner products but the following derivation of the nonlinear extension of SVR shows that it is possible to obtain the inner products directly. This can be achieved by the use of kernel functions (Smola & Schölkopf, 2004).

One way to apply the above linear SVR to nonlinear data is to make use of a nonlinear mapping function $\Phi(\boldsymbol{x})$. The regression function and the optimization problem are then defined by inner products of the corresponding mapping function

$$f(\boldsymbol{x}; \Theta) = \sum_{i \in \mathbb{S}} (\alpha_i - \alpha_i^*)\Phi(\boldsymbol{x}_i)'\Phi(\boldsymbol{x}) + b. \tag{54}$$

Note that a mapping function shifts the optimization problem and the resulting SVR model from the original feature space into a new higher dimensional feature space

$$\Phi : \boldsymbol{X} \rightarrow \boldsymbol{\mathfrak{X}} \tag{55}$$

with $\boldsymbol{\mathfrak{X}} \subset \mathbb{R}^Q$ where $Q > D$.

For example, consider the following mapping function assuming that there are only two distinct variables in the original feature space

$$\Phi(x_{i1}, x_{i2}) = (x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i1}^2) \tag{56}$$

In this example, $\Phi(\boldsymbol{x})$ maps the original two dimensional space into a new three dimensional feature space. The dimensionality of the resulting feature space rises disproportionally with the dimension of the

[26] The parsimony becomes obvious when comparing SVR's fast execution time with the computational expense of other machine learning algorithms with SVR running much faster

original space. Moreover, the dimension of the resulting space also depends on the nature of the mapping function. Therefore, extending SVR solely by the use of nonlinear mapping functions may become computationally highly expensive or even infeasible.

This is where the kernel functions come into play. The core of the extension of SVR to learn nonlinear functions lies in the usage of kernel functions. The fundamental property of kernel functions is that they shift the original feature space *implicitly* to an higher order feature space via the mapping function $\Phi(\boldsymbol{x})$. Kernel functions do this implicitly as they never compute the actual coordinates of the original features in this higher order space but rather define them as the inner products in this transformed space

$$K(\boldsymbol{x}_i, \boldsymbol{x}) = \Phi(\boldsymbol{x}_i)'\Phi(\boldsymbol{x}). \tag{57}$$

In the context of kernel functions, the higher order feature space is therefore also referred to as inner product space. For the above mapping function, the corresponding kernel looks as follows

$$\begin{aligned} K(\boldsymbol{x}_i, \boldsymbol{x}) &= \Phi(\boldsymbol{x}_i)'\Phi(\boldsymbol{x}) \\ &= (x_{i1}^2 \ \sqrt{2}x_{i1}x_{i2} \ x_{i1}^2)(x_1^2 \ \sqrt{2}x_1x_2 \ x_1^2)' \\ &= x_{i1}^2x_1^2 + 2x_{i1}x_1x_{i2}x_2 + x_{i2}^2x_2^2 \\ &= (x_{i1}x_1 + x_{i2}x_2)^2 \\ &= (\boldsymbol{x}_i'\boldsymbol{x})^2 \end{aligned} \tag{58}$$

which is a second order polynomial kernel.

As kernels take input vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}$ as function arguments and directly return the value of the inner product of their images $\Phi(\boldsymbol{x}_i)$ and $\Phi(\boldsymbol{x})$, the functional form of $\Phi(\boldsymbol{x})$ does not have to be known (Bhattacharyya, 2018). Therefore, extending SVR to nonlinear problems by means of kernel functions is computationally inexpensive. The necessary and sufficient conditions which guarantee that $K(\boldsymbol{x}_i, \boldsymbol{x})$ is an inner product in the higher order feature space are known as Mercer's Theorem.[27]

Given that $K(\boldsymbol{x}_i, \boldsymbol{x})$ satisfies Mercer's Theorem, the nonlinear SVR function can be defined as follows:

$$f(\boldsymbol{x}; \Theta) = \sum_{i \in \mathbb{S}} (\alpha_i - \alpha_i^*) K(\boldsymbol{x}_i, \boldsymbol{x}) + b. \tag{59}$$

[27] For more details on Mercer's Theorem, refer to Smola and Schölkopf (2004).

4 Results

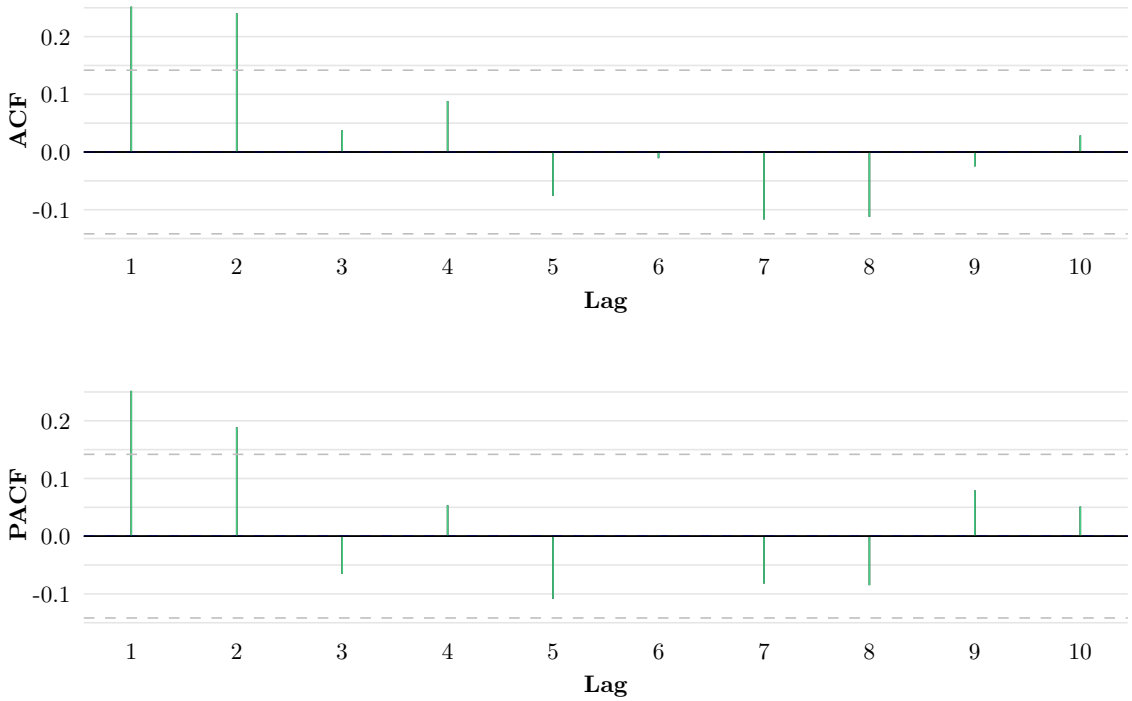
4.1 Model Building

This section presents the tuning results that determine the hyperparameter calibration of the respective forecasting models. The model calibration process takes exclusively place on the training data in the inner loop of cross-validation. The strict demarcation of model building on the training set and model validation, as presented in section 4.2, on the test set allows to assess the models' generalization performance most accurately. In the following, the tuning results are presented for each model separately.

Univariate Autoregressive Model

The crucial part of developing an ARMA model is to determine the number of lags of the target variable and of the error term. Prior to specifying the model order, it is helpful to analyze the patterns in the ACF and PACF of the seasonally adjusted quarterly GDP growth variable as presented in figure 5. From a pure visual inspection of the correlograms, it is difficult to draw any conclusions regarding the model order. The significant spike at lag 2 in the PACF gives weak hint to an AR(2) process. However, in order to find out the best model order the use of information criteria is necessary.

Figure 5: Empirical Autocorrelation Functions



Note: Top figure shows ACF of real quarterly U.S. GDP growth up to lag 10. Bottom figure shows PACF for the same series. Dashed gray lines indicate 95% confidence bands.

Following the seven-step procedure of Hyndman and Athanasopoulos (2018), the final order selection is based on the corrected Akaike Information Criterion (AICc), an information criterion based on the well known Akaike Information Criterion (AIC) with a small sample size correction term (Hurvich & Tsai, 1989). For reasons of sparsity, the maximum of lags for both target variable and error term is restricted to 10 in the calculation of AICc.

Applying AICc to the training set indicates that ARMA$(2, 1)$ describes U.S. GDP growth best. Diagnostic analyses of the training residuals resulting from an ARMA$(2, 1)$ strongly support the model selection. Portmanteau tests with the underlying H_0 that the autocorrelation of the residuals is not significantly different from zero cannot be rejected at any conventional level. This clearly supports the model choice. Visualizations of the residuals as well as detailed results from the Portmanteau tests can be found in

appendix C.

Vector Autoregressive Model

Despite VAR's advantage of acting on a richer information set as compared to its univariate counterpart, VAR models strongly suffer the curse of dimensionality as the number of parameters increases disproportionally with the number of incorporated variables. Note that for a given number of K variables in the VAR system, K^2 coefficients for each of the p distinct coefficient matrices need to be estimated. In addition, the $K(K-1)/2$ elements in the covariance matrix of the error terms need to be determined. At some point, a VAR model runs out of degrees of freedom, making the inclusion of additional variables impossible. In fact, VAR models are rarely used to model more than six to eight variables (Bernanke et al., 2005). In this paper, the degrees of freedom problem is of particular concern since the training data comprises more variables than observations. Generally, the researcher needs to decide which variables to include as a first step in specifying a VAR model. In a scenario where the researcher faces a high dimensional space of possible predictors the advantages of VAR models vanish due to their limitation in coping with larger sets of variables. The choice of variables typically refers to economic theory or a priori ideas of the researcher. In this fashion, this paper considers two distinct VAR models.

The first one (later referred to as NK VAR) is 'inspired' by a New Keynesian approach of modeling macroeconomic aggregates. In its simplest form, a New Keynesian economy is described by three variables: output gap (i.e. the gap between actual GDP and potential GDP), inflation (CPIAUCSL) and the nominal interest rate (FEDFUNDS).[28] Economically motivated VAR models of this kind are closely related to the heavily criticized doctrine of DSGE modeling in macroeconomics. Giacomini (2013) gives a detailed explanation of how, under certain identification strategies, a DSGE model can be represented as a reduced-form VAR model. In simple terms, DSGE models focus on structural equations which are based on economic rationale and allow for contemporaneous relations among its variables but at the same time demand uncorrelated error terms across equations. Orthogonality of the error terms then allows to shock one equation and analyze the effects of this isolated shock on all other variables via impulse response functions (Tenhofen, Guntram, & Heppke-Falk, 2010). Typically, a New Keynesian DSGE model is based on the following structural relations. It describes aggregate demand via the IS curve by modeling current output gap as the difference between expected output gap and the disparity between real interest rate and the natural interest rate. Furthermore, it incorporates aggregate supply via the New Keynesian Phillips curve which relates inflation today to both expected inflation one period ahead and current output gap. Finally, it considers the Taylor Rule that implies that nominal interest rates respond to current inflation and output (see for example Galí (2018) for more information on the current state of New Keynesian DSGE models). It is the predominance of such New Keynesian approaches in modeling and forecasting economic aggregates which has been target of much of the criticism in the aftermath of the global financial crisis. A pure VAR model, in contrast, is a statistical approach to yield forecasts from the interrelation of several time series. It explains the realizations of a target variable today with its own lags and lagged values of other variables incorporated into the system of equations. In this sense, VAR models exploit the autocorrelation and the intertemporal cross-correlation of the variables included in the system in order to produce forecasts. This paper does not intend to model overly stylized relations between macroeconomic aggregates, nor is it interested in detecting 'causalities' by means of exogenous shocks as it is typically done in DSGE frameworks. Rather, it attempts to exploit information entailed in the intertemporal correlation among multiple time series in order to produce GDP growth forecasts. In fact, a VAR model incorporating the variables of a typical three-variable New Keynesian DSGE model is the closest this paper gets in setting up a model based on economic theory.

The second VAR specification (later referred to as LI VAR) focuses more on the potential forecasting quality of the features included in the vector autoregression. It aims at exploiting the signaling effect of variables which tend to have a leading relationship to movements in the business cycle. Therefore, the second VAR model includes the six U.S. leading indicators introduced in section 2.2. According to the OECD (2019), these comprise housing starts (HOUST), manufacturers' new orders of durable goods (AMDMN_OX), S&P 500 stock price index (S_P_500), consumer sentiments (UMCSEN_TX), weekly hours worked in manufacturing (AWHMAN) and interest rate spread between 3-Month treasury constant maturity and federal funds rate (TB3SMFFM) for the U.S. economy (see J. H. Stock and Watson

[28] See Chauvet and Potter (2013) and J. H. Stock and Watson (2002) for the implementation of a New Keynesian forecasting model and J. H. Stock and Watson (2001) who instead of GDP incorporate unemployment via Okun's Law into their three-variable VAR model. This paper takes real GDP growth, y_t, instead of output gap in the respective VAR models.

(2002) for a similar approach).

The lag length of the VAR(p) models is determined by the Bayesian Information Criterion. The maximum lag order considered in building the model is capped at 10 for reasons of parsimony. The coefficients in the system of equations in (16) can be estimated by OLS, equation by equation. Since the white noise components are assumed to be independent of the lagged values of $\boldsymbol{y}_t$, the OLS estimates are consistent (Verbeek, 2004). Elements in the covariance matrix are estimated from the sample covariance matrix of the residuals. It is necessary to test for the assumption of serially uncorrelated residuals to justify the model's correct specification. In this paper, a multivariate extension of the Box-Pierce test is used to test for serially correlated residuals (Box & Pierce, 1970). If this form of Portmanteau test hints to serial correlation in the residuals, the lag order is extended sequentially, moving closer to the model order suggested by AIC. This process is continued until the Portmanteau test gives sufficient confidence for serially uncorrelated residuals at a reasonable number of lags. A traditional F-Test based Granger causality analysis is conducted both for the overall model and specifically for the equation in the VAR system with real GDP growth, y_t, on the left-hand side. Model building results including information criteria, final order selection and model specification tests can be found in table 2 for both VAR models. Both Granger causality tests as well as the Portmanteau tests suggest that the models are specified correctly.

Table 2: VAR Results

(A) New Keynesian VAR(2) model

Joint Granger causality		**Single Granger causality**		**Multivariate Box-Pierce test**	
left-hand variable	p-value	included variable	p-value	lags	p-value
y_t	< 0.01			30	< 0.01
CPIAUCSL	< 0.01	CPIAUCSL	< 0.01	40	0.04
FEDFUNDS	< 0.01	FEDFUNDS	< 0.01	50	0.22
BIC	2				
AIC	10				

(B) Leading indicator VAR(1) model

Joint Granger causality		**Single Granger causality**		**Multivariate Box-Pierce test**	
left-hand variable	p-value	included variable	p-value	lags	p-value
y_t	< 0.01			20	< 0.01
HOUST	< 0.01	HOUST	0.02	25	< 0.01
AMDMN_OX	< 0.01	AMDMN_OX	0.04	30	< 0.01
S_P_500	0.01	S_P_500	0.28	35	0.11
UMCSEN_TX	< 0.01	UMCSEN_TX	0.09	40	0.24
AWHMAN	< 0.01	AWHMAN	< 0.01	45	0.26
T5YFFM	< 0.01	T5YFFM	< 0.01	50	0.65
BIC	1				
AIC	2				

Note: For each equation in the VAR model joint, Granger causality tests determine whether the simultaneous inclusion of all right-hand variables other than the lags of the left-hand variable leads to a significant reduction in RSS. This means that the test compares the RSS of the AR model of the left-hand variable with the RSS of the VAR model including all variables. The column 'left-hand variable' reflects the respective equation in the VAR model for which the joint Granger causality is conducted. If the p-value is below the 5% level, the lagged values of the right-hand variables are said to 'Granger cause' the left-hand variable.
The single Granger causality is designed only for the equation where y_t is on the left-hand side and tests whether the inclusion of the lags of the listed right-hand variable leads to a significant decrease in RSS. The column 'included variable' reflects the right-hand variable which is included in the minimal model where GDP is only regressed on its own lags. If the p-value is below the 5% level, the lagged values of the included variable is said to 'Granger cause' real GDP growth, i.e. y_t. Note that the in-sample reduction in RSS due to the inclusion of the S&P 500 price index in the leading indicator model is insignificant at the 10% level. For this reason, S&P 500 price index will not be considered in the final leading indicator forecasting model.
The multivariate Box-Pierce test reflects joint statistical significance of the H_0 of no autocorrelation up to the number of specified lags. P-values above the 5% level suggest that the residuals in VAR model are not autocorrelated. The choice of the number of lags incorporated into Portmanteau tests is controversially discussed in literature (see, for example, Hyndman (2014) for a reflection on the issue). The asymptotic χ^2-distribution under the null hypothesis only holds if the number of lags is sufficiently large; yet if the number is too large, the test loses its power. Both VAR models indicate non-existence of autocorrelation in the residuals at a reasonable number of lags.

Factor-Augmented Vector Autoregressive Model

Specifying FAVAR models requires two essential steps: finding the optimal number of components and specifying the right number of lags. This paper uses cross-validation with a rolling-origin-recalibration strategy, as it has been described in section 3.1.2, to determine the optimal number of principal components. The determination of the optimal number of components is exclusively conducted on the training set. For reasons of parsimony, the maximum number of components considered during cross-validation is limited to 10. Components are extracted from the correlation matrix of the feature space.
The ultimate number of principal components entering the forecasting model can be found in table 3. Appendix D shows further details regarding the explained variance and the loadings of the respective principal components.

This paper implements two different FAVAR models. The first one (later referred to as Full FAVAR) considers all variables of the feature space in the Principal Component Analysis. A second FAVAR model (later referred to as S&W FAVAR) only considers features which have also been used by J. Stock and Watson (2012) in their paper 'Disentangling the Channels of the 2007 - 2009 Recession'. Their paper is particularly interesting in the context of this study for two reasons. First, they use a very similar dataset as in this paper with variables from the Federal Reserve Bank. Both datasets comprise a large number of macroeconomic time series for the U.S. economy with the earliest observations dating back to the first quarter of 1959. The maintainers of FRED-QD provide a label that indicates which of the variables have been used in the paper of J. Stock and Watson (2012).[29] Their dataset comprises mainly disaggregated variables and excludes aggregate variables such as total consumption or the like. Second, using a dynamic factor model J. Stock and Watson (2012) analyze whether in the course of the global financial crisis a structural break in the factor loadings has been registered. They find little evidence for a rise of 'new' factors in the financial crisis of 2007, leading them to the conclusion that the crisis has been the result of shocks that were not substantially different to previous shocks but just larger (J. Stock & Watson, 2012). Therefore, they argue that the 'economy responded in an historically predictable way' (J. Stock & Watson, 2012, p. 129). Their finding is fundamental to the widespread criticism of macroeconomic forecasting outlined in the introduction of this paper. If the last crisis could have been predicted, then the methodological criticism is very much legitimate since existing forecasting models failed to foresee what potentially would have been predictable. This reinforces the necessity to find new ways that are capable of forecasting macroeconomic aggregates more reliably.

Random Forest

A big advantage of Random Forest is that the algorithm has a built-in feature selection procedure which allows to assess the forecasting importance of the different forecasting features. This gives RF a greater degree of interpretability as compared to many other machine learning methods. Splitting the training data in the decision tree by means of a specific predictor variable, typically improves the RSS. This makes it possible to assign to each feature acting as splitting variable in at least one of the trees a variable importance measure. Since a feature may act as splitting variable several times in multiple trees, the overall importance measure equals the improvement in RSS averaged over all trees in which it operates as splitting criterion (Hastie et al., 2009). Note that the random subspacing method makes it more likely that all features act as splitting variable at least once in a forest. The variable importance measure, displayed in figure 6, provides a good insight into the feature space and allows to assess which features are particularly important in predicting the target variable.

On the one-quarter horizon, variables of the National Income and Product Accounts such as real personal consumption expenditures in the service sector (PCES_VX_{t-1}) and residential real private fixed investments (PRF_IX_{t-1}) have strong power in short-term GDP growth forecasting. These variables are direct components of GDP and it seems natural that changes in these variables have strong impact on short-term changes in GDP. Also, the ratio of published job vacancies to the total number of unemployed (HWIURATI_OX_{t-1}) has strong forecasting power with regard to future short-term movements in GDP. This seems plausible since a high number of job vacancies relative to the number of unemployed people, for instance, suggests that companies have full order books and expect good business prospects. This, in turn, translates into future growth in GDP.

On the one-year horizon, information from the yield curve seems to be particularly important in forecasting U.S. GDP growth. Both 5-year treasury constant maturity minus federal funds rate (T5YFFM_{t-4}) and 10-year treasury constant maturity minus 3-month treasury bill rate (GS10TB3MX_{t-4}) are important GDP predictors. Changes in interest rate spreads reflect investors' expectations concerning the future. An

[29] See appendix A, fourth column, for a label indicating which variables were used in J. Stock and Watson (2012).

Table 3: FAVAR Results

(A) Full FAVAR(3) model including one factor

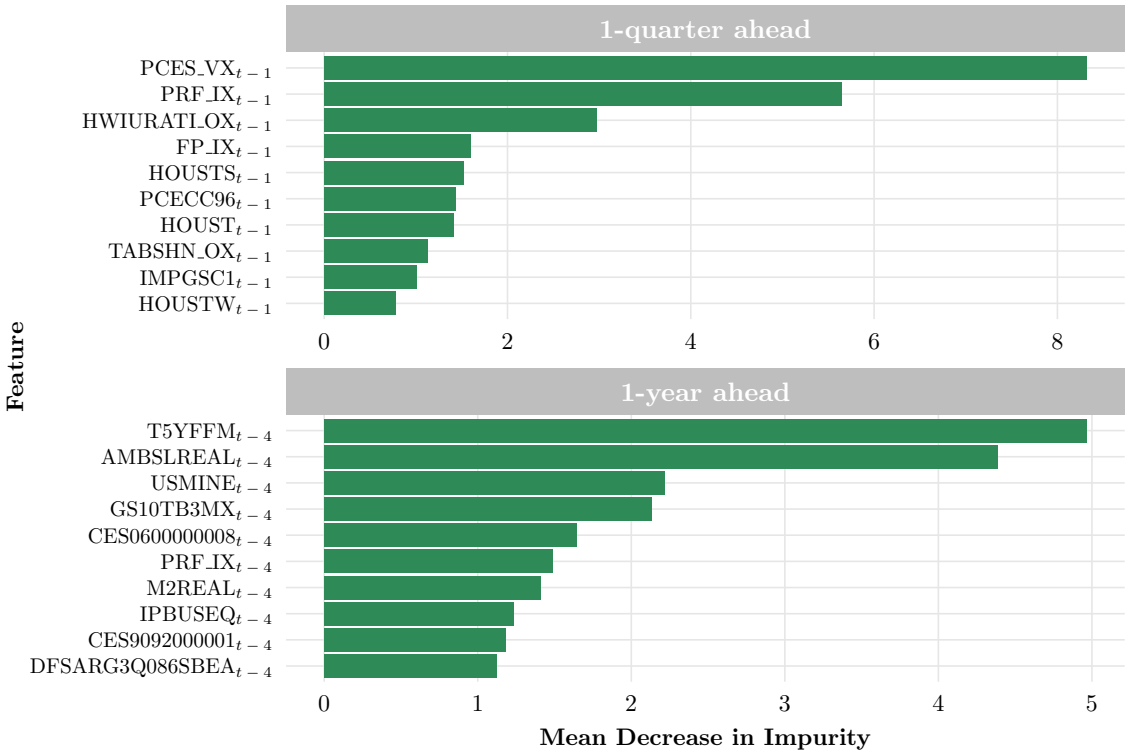
Joint Granger causality		**Single Granger causality**		**Multivariate Box-Pierce test**	
left-hand variable	p-value	included variable	p-value	lags	p-value
y_t	< 0.01			10	0.06
FACTOR1	0.03	FACTOR1	< 0.01	20	0.38
				30	0.26
				BIC	1
				AIC	3

(B) S&W FAVAR(3) model including two factors

Joint Granger causality		**Single Granger causality**		**Multivariate Box-Pierce test**	
left-hand variable	p-value	included variable	p-value	lags	p-value
y_t	< 0.01			10	0.10
FACTOR1	< 0.01	FACTOR1	< 0.01	20	0.47
FACTOR2	< 0.01	FACTOR2	< 0.01	30	0.47
				BIC	2
				AIC	3

Note: The full FAVAR(3) model takes the complete feature space of 210 variables into consideration when estimating the factors. Cross-validation on the training set yields the lowest RMSE if only the first principal component is included in the forecasting model. The Principal Component Analysis in the S&W FAVAR(3) model is based on the variables considered in J. Stock and Watson (2012). These comprise 100 variables from the original feature space. For the S&W FAVAR(3) model, cross-validation yields the lowest RMSE if the first two principal components are incorporated. The small number of factors which enter the VAR equation is consistent with previous empirical work (see, for example, J. H. Stock and Watson (2005)). See also notes on table 2 for more details regarding the tests presented in this table.

Figure 6: Variable Importance of Random Forest



Note: This figure shows the total decrease in node impurities from splitting the training data by means of the respective feature averaged over all trees. Node impurity is measured by RSS. Variable importance is measured on the out-of-sample test set and is only displayed for the ten most important splitting variables.

inverted yield curve, for instance, which implies that longer-term returns are lower than short-term rates,

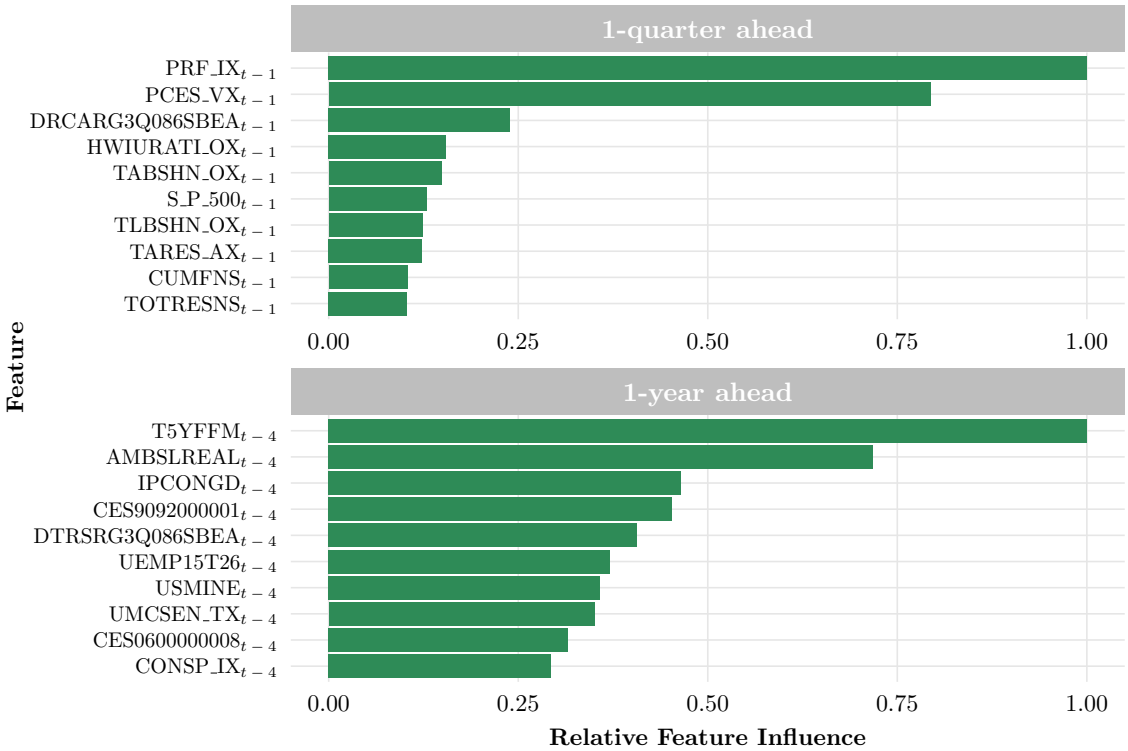
signals poor expectations concerning the future state of the economy. The yield curve as expectation-based measure is typically referred to as leading indicator of changes in the business cycle (OECD, 2019). Therefore, the high importance of yield curve-related variables seems highly plausible. Another variable which has long-term forecasting power is the adjusted monetary base (AMBSLREAL_{t-4}). Adjusted monetary base is the sum of currency in circulation outside the Federal Reserve Banks and the U.S. Treasury. It is the variable that captures monetary policy interventions of the Federal Reserve System (FED). Increasing the monetary base through purchases of bonds improves the investment landscape which ultimately favors GDP growth. With this reasoning, the strong forecasting power of the adjusted monetary base seems plausible as well.

The final RF specification which results from calibrating the model on the training set can be found in table 4. As it turns out, the best (short-term) GDP growth forecasting model consists of only 90 decision trees. With a minimum terminal leaf size of 95 observations, these trees are rather small. Moreover, the trees randomly consider a rather large number of 183 out of the 210 possible splitting variables at each node. A visual inspection of how first-stage tuning narrows down the initially wide search spaces which ultimately lead to the here mentioned optimal values can be found in appendix E.

Gradient Boosting

Similar to RF, Gradient Boosting also returns a variable importance measure (Friedman, 2001). Figure 7 shows that the most important features for GB are very similar to those selected by RF on both horizons. This suggests that the results obtained from both algorithms are robust. Besides information from the National Income and Product Accounts, GB additionally attaches high forecasting importance to changes in the stock market (S_P_500_{t-1}) on the short-term horizon. Changes in the stock market are typically seen as leading indicator because fluctuations in stock prices reflect investors' expectations concerning the future state of the economy. Moreover, it is interesting to see that in the GB framework the consumer sentiment index of the University of Michigan (UMCSEN_TX_{t-4}) is also among the top ten variables with highest feature importance on the one-year horizon. Sentiment indices are highly expectation-based and their changes often precede later movements in the business cycle. Therefore, it makes sense that GB attaches strong forecasting power to UMCSEN_TX.

Figure 7: Variable Importance of Gradient Boosting



Note: Scaled variable importance is measured on the out-of-sample test set and is only displayed for the ten most important splitting variables. See Friedman (2001) for the exact calculation of relative importance measure in boosted regression trees.

The final GB specification which results from calibrating the model on the training set can be found in table 4. As it turns out, the best (short-term) GDP growth forecasting model consists of a sequence of 500 decision trees which are no deeper than 9 variable interactions. The tree sequence learns at a rate of 0.025 which is consistent with literature (James et al., 2013). A visual inspection of how first-stage tuning narrows down the initially wide search spaces which ultimately lead to the here mentioned optimal values can be found in appendix E.

Support Vector Regression

This paper tries different kernels to shift the original feature space to an implicit higher order feature space. The following kernels are used for this task:

$$\text{Polynomial kernel:} \quad K(\boldsymbol{x}_i, \boldsymbol{x}) = (\gamma \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle)^w \tag{60}$$

$$\text{Radial kernel:} \quad K(\boldsymbol{x}_i, \boldsymbol{x}) = \exp(-\gamma (\boldsymbol{x}_i - \boldsymbol{x})'(\boldsymbol{x}_i - \boldsymbol{x})) \tag{61}$$

$$\text{Sigmoid kernel:} \quad K(\boldsymbol{x}_i, \boldsymbol{x}) = \tanh(\gamma \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle) \tag{62}$$

The kernels are treated as hyperparameters and the best performing kernel on the training data is the final kernel used to produce forecasts on the test set. It is important to mention that w and γ are tunable kernel parameters. However, following the approach in Cherkassky and Ma (2004), tuning only takes C and ε as well as the *type* of kernel into account. While the above kernels are all considered in the tuning process, their parameters are held fix at reasonable values (w equals 3 and $\gamma = (\text{number of features})^{-1}$). The final SVR specification which results from calibrating the model on the training set including the best performing kernel can be found in table 4. It turns out that a sigmoid kernel performs best in the (short-term) GDP forecasting task. The optimal regularization parameter and the optimal radius of the ε-tube assume reasonable values. A visual inspection of how first-stage tuning narrows down the initially wide search spaces which ultimately lead to the here mentioned optimal values can be found in appendix E.

Table 4: Optimal Hyperparameter Setting

Model	Hyperparameter	Optimal value
RF	M	90
	d_{try}	183
	$node_{min}$	95
GB	M	500
	ν	0.025
	$depth_{max}$	9
SVR	Kernel	Sigmoid
	C	0.0677
	ε	0.0587

Note: Optimal values according to second-stage finetuning based on one-quarter ahead forecasts.

4.2 Generalization Performance

In this section, the tuned models' generalization performance is examined. The hyperparameters which result from the calibration strategies applied on the training data form the basis of the final models. The final evaluation of these models takes place on the hold-out test set that comprises the period from 2007-Q2 to 2019-Q2. This out-of-sample assessment allows to compare the different forecasting models based on how well they perform on unseen data. Moreover, the inclusion of the global financial crisis in the test set permits to assess the models' capability to forecast recessions in advance. In a first analysis, it is tested how the models perform given different information sets. In a second assessment, the focus lies on the models' performance in the last crisis. All assessments comprise the performance of both one-quarter ahead and one-year ahead forecasts.

4.2.1 Performance based on Different Information Sets

In the following analysis, the performance of the forecasting models is evaluated in the context of the amount of information the models are trained with. The analysis moves from a sparse information set including only the target variable and lags thereof to a richer information set which includes all significant leading indicators introduced in section 2.2, and ends with the richest information set possible comprising the complete feature space $\boldsymbol{X}$. In this way, it is possible to get an understanding of how well the models perform against each other if they are facing the same amount of information. Moreover, it highlights that econometric models such as RW, AR and VAR are limited in the amount of information they can cope with. Indeed, most econometric models are only designed to cope with sparse information while machine learning models display their full potential when being confronted with rich information sets.

Information Set I

In the first step, the models are running on the most basic information set comprising only the target variable and its lags $\mathcal{I}_t = \{y_1, \dots, y_t\}$. This means that the machine learning methods only consider lags of the target variable as predictors, although they are capable of extracting information from a much larger number of predictors. Table 5 shows the out-of-sample performance as measured by both MdRAE and RelRMSE.[30] Based on absolute errors, MdRAE is little susceptible to large forecast errors. RelRMSE, in contrast, is based on squared errors and the mean as aggregation measure and is thus much more sensitive to large forecasting errors. Assessing the models' generalization performance on both measures sheds more light on the models' strengths and weaknesses. Both measures are relative metrics which can be easily interpreted: If the measure is smaller than one, the respective model performs better than the benchmark model; if the measure is greater than one, the opposite holds. In the analysis based on information set I, the RW model serves as benchmark. This means that the errors in table 5 are relative to the errors produced by the RW model. The best results for both measures on the respective forecasting horizons are highlighted in color.

Two important insights result from table 5. First, ARIMA outperforms the RW model on both horizons and both measures. Measured by RelRMSE, ARIMA is even the best performing model which means that it outperforms the more complex machine learning models. Generally, machine learning models perform relatively poor on information set I. Only RF outperforms RW consistently (while being outperformed by ARIMA in most instances). GB produces in all instances even worse forecasts than the naïve RW model. SVR is rather volatile. Based on MdRAE, SVR is the best model on information set I and slightly outperforms ARIMA, while it shows poor results for the RelRMSE. This suggests that the model is likely to produce large errors in periods which are difficult to forecast. Overall, the results on information set I are little surprising as the machine learning models remain far below their potential when being confronted with the history of the target variable only.

Information Set II

The second step extends the information set to the leading indicators and their lags $\mathcal{I}_t = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_t\}$ with $\boldsymbol{y}_t$ as vector including the target variable as well as all leading indicators with significant Granger causality. In the analysis based on information set II, the leading indicator VAR model is used as benchmark accordingly.

In table 6, it is interesting to see how the machine learning models start to perform better as they run on a richer information set. Especially GB clearly outperforms the benchmarking VAR model on the one-

[30] See section 3.1.4 for a definition of the error measures and a description of their properties.

Table 5: Model performance: Information Set I

Model	Information set	Benchmark model	1-quarter ahead MdRAE	1-quarter ahead RelRMSE	1-year ahead MdRAE	1-year ahead RelRMSE
RW	I	×				
ARIMA	I		0.808	0.928	0.755	0.800
RF	I		0.971	0.945	0.690	0.806
GB	I		1.050	1.006	1.015	1.056
SVR	I		0.776	1.006	0.755	0.804

Note: Input space in machine learning models comprise real GDP growth up to lag 2 in order to be in line with the AR(2) framework in the autoregression.

quarter ahead forecast. On both forecasting horizons, the best models come from the machine learning family: GB on the one-quarter horizon and SVR on the one-year horizon. Nonetheless, in some instances, the machine learning methods perform worse than the benchmark. It is also noteworthy that ARIMA outperforms the benchmark model in all instances which shows that univariate autoregression is also a good model for the task of forecasting U.S. GDP growth.

Table 6: Model performance: Information Set II

Model	Information set	Benchmark model	1-quarter ahead MdRAE	1-quarter ahead RelRMSE	1-year ahead MdRAE	1-year ahead RelRMSE
RW	I		0.957	1.050	1.216	1.215
ARIMA	I		0.803	0.974	0.993	0.972
LI VAR	II	×				
NK VAR	II		0.717	1.034	0.999	0.993
RF	II		0.938	0.978	1.022	0.993
GB	II		0.666	0.909	1.273	0.993
SVR	II		0.863	1.102	0.965	0.958

Note: Input space in machine learning models comprise real GDP growth and all Granger causing leading indicators up to lag 1 in order to be in line with the leading indicator VAR(1) framework.

Information Set III

In the final step, the models are confronted with all variables in the feature space $\mathcal{I}_t = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$ with $\boldsymbol{x}_t$ as vector including the target variable as well as all other macroeconomic series observed in t. The Full FAVAR model that optimally extracts one principal component from the full feature space is used as benchmark model.

There are three important conclusions one can draw from the results in table 7. First, extending the information set to the full feature space leads to a clear improvement in the overall out-of-sample forecasting performance. All models which run on the inferior information sets I and II show on both horizons and both measures larger forecasting errors. This is obvious because all measures in the upper two parts of table 7 take on values greater than one which means that they perform worse compared to the benchmark FAVAR model that is capable of operating on information set III. Second, the best models in all instances, again, belong to the machine learning family. RF performs best as measured by the MdRAE and GB outperforms all other models based on RelRMSE. GB's superiority on the RelRMSE points towards its capability to forecast difficult periods where other models throw large errors. Third, despite the fact that the best models come from the machine learning family, their performance is not consistently better than the results of the benchmark FAVAR model. In fact, SVR is even outperformed by the benchmark FAVAR in all instances. Also, the differences in forecasting performance of the models within information set III is mostly marginal. In order to evaluate whether the superior performance of machine learning methods is significant, the Diebold-Marino (DM) Test serves as useful tool. Based on the forecasting error differential between two models, the original DM-Test formulates a test statistic which, under the null hypothesis that both models perform equally well, is normally distributed. Figure 8 displays the test results of the DM-Test.
Figure 8 provides further insights regarding the question of how useful machine learning is for the task of macroeconomic forecasting. It can be seen that the machine learning methods' accuracy increase com-

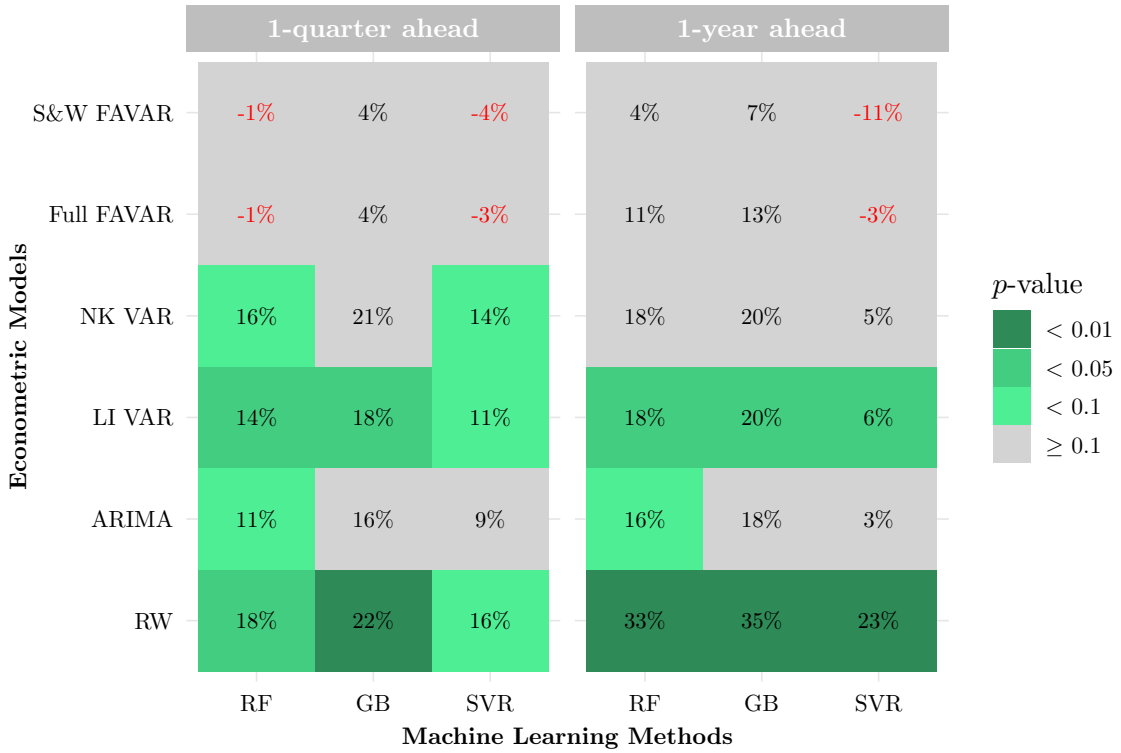
Table 7: Model performance: Information Set III

Model	Information set	Benchmark model	1-quarter ahead MdRAE	1-quarter ahead RelRMSE	1-year ahead MdRAE	1-year ahead RelRMSE
RW	I		1.567	1.226	1.522	1.329
ARIMA	I		1.178	1.138	1.136	1.064
LI VAR	II		1.707	1.168	1.306	1.094
NK VAR	II		1.007	1.207	1.179	1.086
Full FAVAR	III	×				
S&W FAVAR	III		1.052	1.000	1.049	0.930
RF	III		0.984	1.009	0.811	0.895
GB	III		1.197	0.959	0.981	0.869
SVR	III		1.056	1.035	1.086	1.029

Note: Input space in machine learning models comprise real GDP growth and all features up to lag 1 which is the most parsimonious forecasting feature space.

pared to the econometric models operating on information set I and II is not only substantial in size but also in many instances statistically significant as indicated by the DM-Test. There are two important exceptions to this. On the one hand, ARIMA is only outperformed by RF in a significant fashion. This reaffirms that ARIMA is not completely uncompetitive in forecasting U.S. real GDP growth. On the other hand, although the New Keynesian VAR model produces worse forecasts on the one-year horizon, the error differentials are not significant according to the DM-Test.

Figure 8: Diebold-Marino (DM) Test Results of Machine Learning Methods against Econometric Models



Note: Values depict the accuracy improvement (accuracy worsening as highlighted in red) of the forecasts produced by the machine learning methods relative to the forecasts from the respective econometric model in percent. Percentage accuracy improvement (deterioration) is measured by RelRMSE. Significance of the accuracy differential is highlighted by the colored grids. Significance is calculated by the original DM-Test test with alternative hypothesis that the machine learning methods' forecasts are more accurate than the econometric forecasts. All machine learning models operate on information set III.

More importantly, figure 8 shows that FAVAR models are a serious competitor to machine learning methods. In fact, the one-quarter ahead forecasts of the FAVAR models even slightly outperform RF and SVR. On the one-year horizon, they outperform SVR. If, by contrast, the FAVAR models are outperformed

by one of the machine learning methods, none of the accuracy improvements is statistically significant. Nonetheless, GB outperforms both FAVAR models in all instances suggesting that GB is the most successful forecasting model in this study. Although this paper counts FAVAR models to the econometric family, it is important to highlight that factor models incorporate with PCA an unsupervised learning component. Based on this consideration and the clear pattern in the significance analysis, one can conclude that machine learning methods successfully contribute to the task of macroeconomic forecasting.

This section has demonstrated that machine learning techniques successfully extract patterns entailed in past realizations of macroeconomic variables for the prediction of future GDP growth. The results in this paper support the hypothesis that machine learning can make important contributions to the field of macroeconomic forecasting. When they are exposed to rich information sets they tend to outperform (often significantly) more traditional time series models. One crucial question remains open. How do machine learning models perform in times of crisis? The following subsection focuses on this issue.

4.2.2 Performance in Crisis

In normal times of positive economic growth, traditional forecasting models tend to produce reliable projections and some structural models even allow to uncover causal channels (Chauvet & Potter, 2013). The great weak spot of existing models has been their incapability to forecast times of recession as it has been highlighted in the introduction of this paper. Therefore, it is paramount to conduct an out-of-sample assessment with special focus on a period of economic crisis. In this way, it is hoped to gain some further insights into the different models' capability to forecast times of economic crisis.

Figure 9 and figure 10 visualize the one-quarter ahead and one-year ahead forecasts in the period of the past financial crisis which, according to NBER, lasted from the first quarter of 2008 until the second quarter of 2009 (gray-shaded area). Several insights can be gained from the visualizations. First of all, the figures show prediction intervals based on 95% confidence level for the probabilistic time series models. Since machine learning methods are non-probabilistic, one cannot directly quantify the uncertainty associated with machine learning forecasts. This problem can be mitigated by using bootstrapped confidence intervals which, in case of RF, even result as by-product of the algorithm. However, for demonstrational purposes, the intervals for RF are kept aside in order to highlight the fundamental difference between the probabilistic and the algorithmic class of models with the latter class not being able to directly display the uncertainty associated with a point forecast.

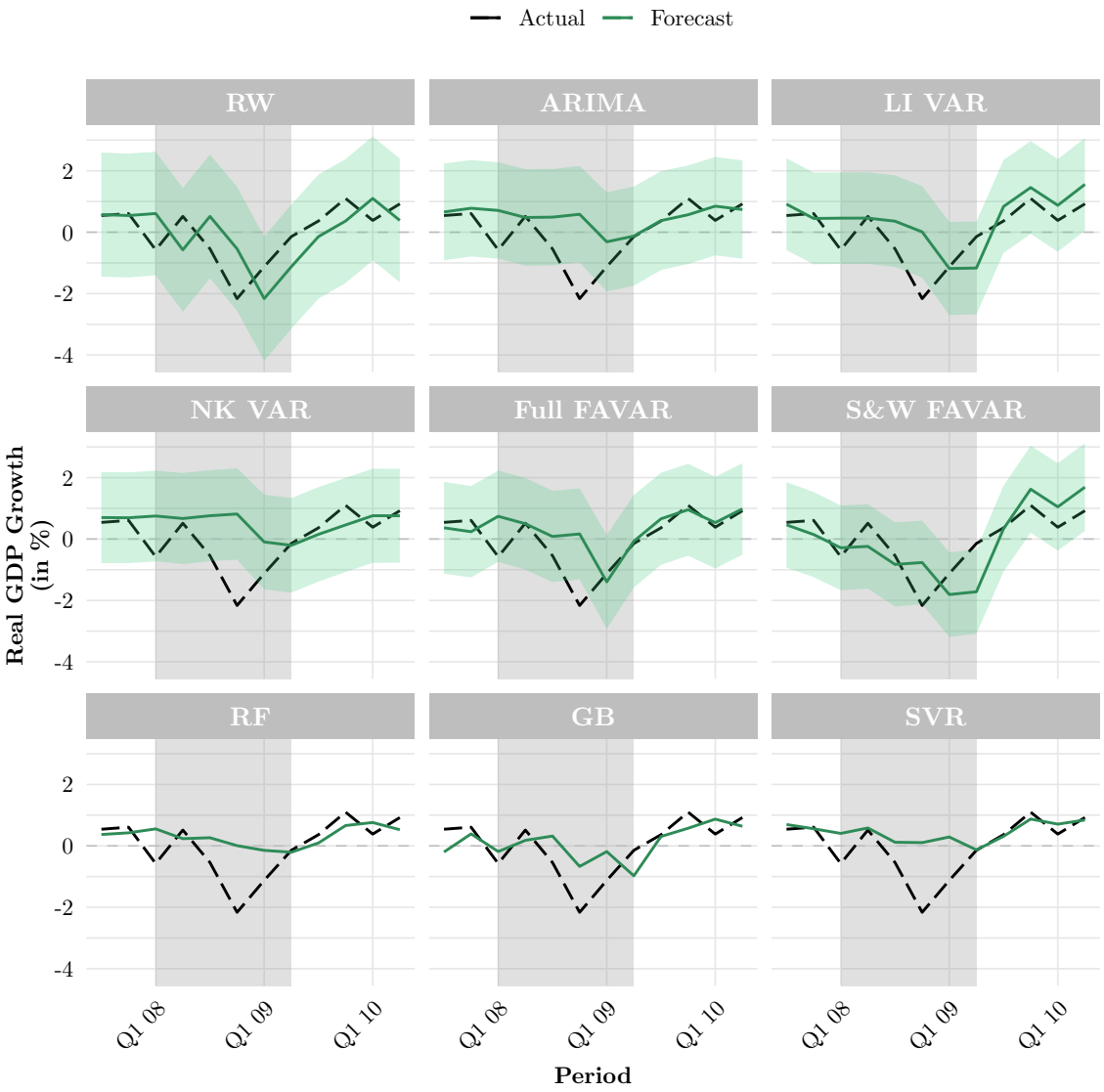
It becomes also obvious that in times of negative GDP growth, econometric one-quarter ahead forecasts tend to clearly lag the actual observations. The strong U.S. downturn in the fourth quarter of 2008 with a contraction of more than 2% is missed by almost all econometric models. Only one to two periods later - after the crisis has been revealed to the models - they forecast a recession albeit smaller in magnitude. This behavior is owed to the autocorrelative structure of the models. The only exception to this behavior is the S&W FAVAR model which forecasts negative GDP growth at an early stage on the one-quarter horizon.[31] In the one-year ahead forecasting scenario, the econometric time series formulations tend to level off due to the convergence of the autoregressive part to the series' long-term mean. This shows that time series models tend to be rather unsuitable for long-term forecasting.

Machine learning methods, in contrast, indicate more promising results. Although they are not capable of forecasting the magnitude of the crisis, they tend to grasp the emergence of a recession at least one-quarter ahead where all of the three machine learning models forecast a small decline in GDP growth at an early stage. GB even forecasts a recession with a projection of -0.7% for the fourth quarter of 2008. This shows that GB is competitive with the well-performing S&W FAVAR model. On the one-year horizon, RF and SVR behave similar to the econometric models and fail to foresee the crisis. However, this is not true for the GB model. GB is the only model which is capable of forecasting a recession on the one-year horizon. In fact, it forecasts a decline in GDP of -0.5% in the fourth quarter of 2008 *one year in advance*. The good forecasting results of the S&W FAVAR model, in contrast, vanish on the one-year horizon.

If one refrains from forecasting the exact magnitude of GDP growth but focuses only on the sign of the forecast and analyzes whether the forecast sign matches the sign of the actual GDP figure, the picture becomes even clearer. This approach is interesting since it resembles the task of recession detection more

[31] Although the Full FAVAR model forecasts a decline in real GDP growth at an early stage, it clearly projects a recession only after the crisis has been revealed to it.

Figure 9: One-quarter ahead Forecasts in Global Financial Crisis

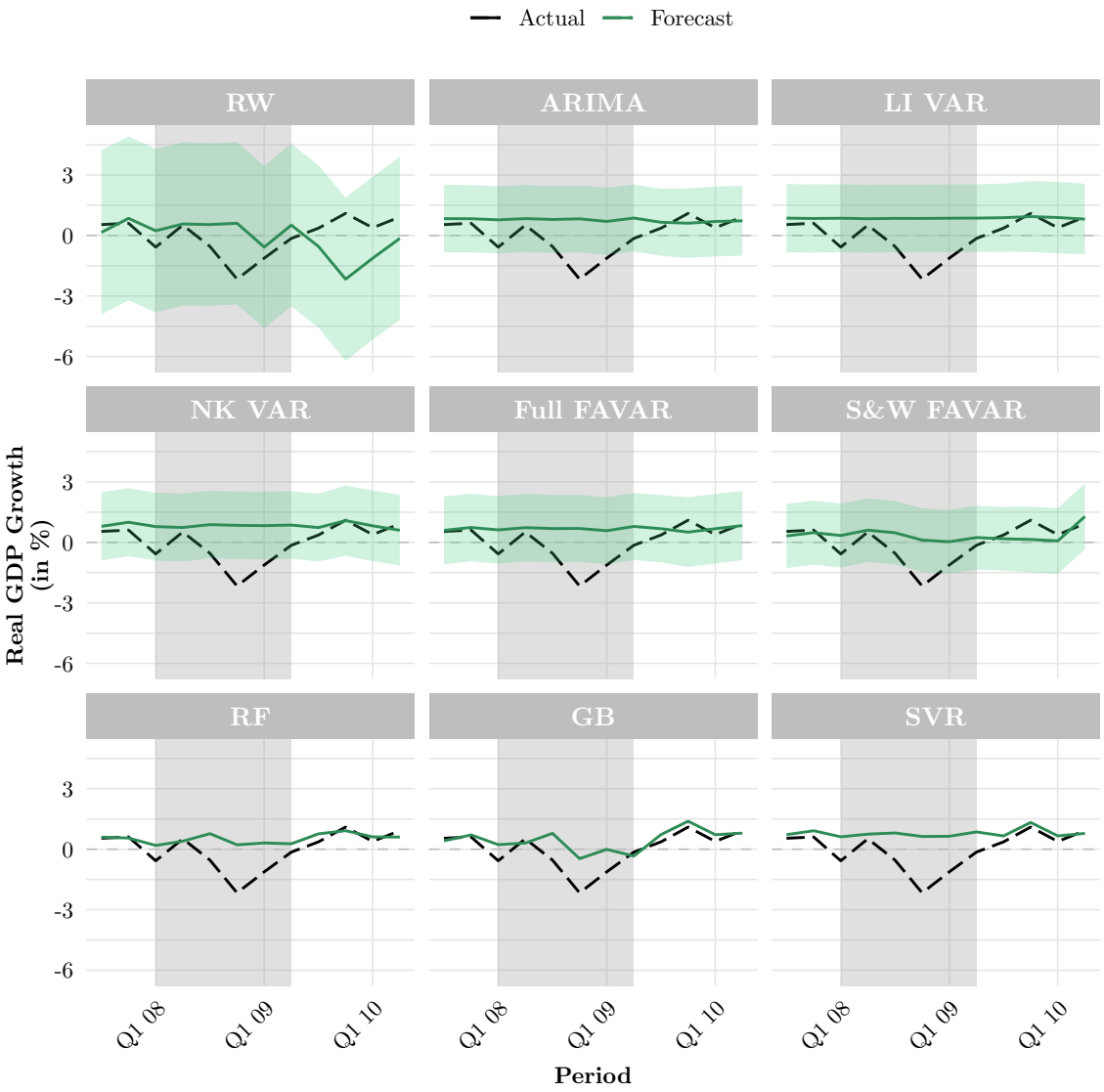


Note: Figure shows one-quarter ahead out-of-sample forecasts around the global financial crisis produced by different models. Solid green line corresponds to the forecasted values, green-shaded area shows the prediction intervals at the 95% confidence level (if applicable) and the dashed black line shows the actual value. The gray-shaded area indicates the official U.S. recession period starting in the first quarter of 2008 and lasting until the second quarter of 2009 according to NBER. All machine learning models operate on information set III.

closely. Table 8 shows the fraction of correctly forecasted GDP growth signs during the U.S. recession of 2008 - 2009. On the one-quarter horizon, only the S&W FAVAR as well as the GB model are capable of outperforming the naïve RW forecasts. In fact, both models miss only one period and forecast the wrong sign. However, on the one-year horizon, only GB continues to outperform RW. In four out of six periods, it forecasts the right sign. *All* other models, including S&W FAVAR, produce very poor sign forecasts. In fact, they only forecast the correct sign in one of the six recession quarters. Thus, they are even outperformed by RW which produces two correct sign forecasts. The sign forecast analysis generally shows that macroeconomic forecasting models indeed underperform in times of crisis. This is especially true for longer forecasting horizons. However, the results in this paper suggest that forecasting models which can cope with high dimensional data can contribute decisively to improving macroeconomic forecasting. One promising candidate is Gradient Boosting which yields the best forecasting results within the scope of this paper.

Despite the good results of GB, it is not possible to draw any conclusion concerning the question whether the model would have predicted the global financial crisis of 2007 - 2009. This is the case since there is no real-time data available for all of the 210 features. Without real-time data, the model does not run on the same information that was available to researchers prior to the crisis. In fact, the analysis

Figure 10: One-year ahead Forecasts in Global Financial Crisis



Note: Figure shows one-year ahead out-of-sample forecasts around the global financial crisis produced by different models. Solid green line corresponds to the forecasted values, green-shaded area shows the prediction intervals at the 95% confidence level (if applicable) and the dashed black line shows the actual value. The gray-shaded area indicates the official U.S. recession period starting in the first of quarter 2008 and lasting until the second quarter of 2009 according to NBER. All machine learning models operate on information set III.

Table 8: Sign Forecast in Global Financial Crisis

Model	1-quarter ahead	1-year ahead
RW	3\6	2\6
ARIMA	3\6	1\6
LI VAR	3\6	1\6
NK VAR	3\6	1\6
Full FAVAR	3\6	1\6
S&W FAVAR	5\6	1\6
RF	3\6	1\6
GB	5\6	4\6
SVR	2\6	1\6

Note: Table shows the number of periods in which the sign of U.S. GDP growth during the six quarter long crisis is forecasted correctly by the respective model. According to NBER, the global financial crisis comprises the periods 2018-Q1 to 2019-Q2. All machine learning models operate on information set III.

in this paper is based on the latest vintage of 2019-Q2. Figure 2 suggests that observations have gone through major revisions which makes comparisons between forecasts produced in 2007 and forecasts produced in this paper difficult. Moreover, it makes it impossible to judge whether GB would have predicated the upcoming global financial crisis prior to 2008. Nonetheless, the results of this paper clearly suggest that forecasting models based on machine learning algorithms pose great potential for the field of macroeconomic forecasting.

5 Conclusion

Macroeconomists have often struggled to accomplish the admittedly challenging task of forecasting the future state of the economy. In times of economic crises such as the 2007 - 2009 global financial crisis, they have even spectacularly failed to do so. Many critics see flawed economic models and economists' resistance to interdisciplinary cooperation at the root of this failure (Bank of England, 2016; Reis, 2018). Given the poor track record of macroeconomic forecasting and the rise and praise of machine learning as problem solving tool across different disciplines in recent years, this paper explores whether statistical learning algorithms can contribute to GDP forecasting. Based on a high-dimensional dataset of the Federal Reserve Bank of St. Louis with data of the U.S. economy over the last 60 years and more than 200 distinct time series, this paper finds promising forecasting results produced by different machine learning methods.

Methodologically, the study forecasts U.S. real GDP growth in a dynamic regression setup by means of Random Forest, Gradient Boosting and Support Vector Regression. All of the three statistical learning methods are designed to deal with high-dimensional data which allow to refrain from judgmental variable selection in the building process of the forecasting model. This makes it possible to run these models on rich information sets which more traditional theory-based models and pure time series models are usually not capable to cope with.

In a comparison of the out-of-sample performance between machine learning methods and time series models, this paper finds lower forecasting errors the richer the information set is machine learning methods are confronted with. If all variables in the dataset are included in the analysis, the machine learning algorithms clearly outperform ARIMA and VAR models that can only operate on a much smaller set of possible predictors. Moreover, the best performing supervised learning approach, Gradient Boosting, consistently (though not significantly) outperforms FAVAR models which are able to operate on the same high-dimensional information set. Although FAVAR models produce forecasts in a vector autoregression setup, the extraction of factors is based on unsupervised learning techniques, only enabling them to operate on high-dimensional datasets.

Including the financial crisis of 2007 - 2009 in the out-of-sample test set, the paper takes also a closer look at the models' performance in times of crises. It turns out that none of the models in this study is capable of forecasting the *magnitude* of the global financial crisis, neither on a one-year forecasting horizon nor on a short-term one-quarter horizon. However, Gradient Boosting is capable of predicting an economic downturn with negative GDP growth both one-quarter ahead and even one-year ahead for the fourth quarter of 2018. The sign forecasts of GB for the global financial crisis are promising and suggest that GB can be useful for macroeconomic forecasts, especially as data-driven feature selection tool. In fact, according to the feature importance measure of GB, changes in National Income and Product Accounts as well as stock market movements seem to have strong forecasting importance in the short term. One-year-ahead forecasts, in turn, can be produced most accurately from information implicitly captured by the yield curve and information encrypted in sentiment surveys.

While this paper uses the supervised learning methods directly to forecast the target variable, future research should explore how to align both worlds - theory-based econometric models and algorithmic machine learning methods. In settings where economists face high-dimensional data with a large number of potential predictors, machine learning methods can indeed assist in identifying which variables help to explain most of the variation in the target variable. Based on this objective and data-driven feature selection, one could elaborate more nuanced forecasting models which are built on economic theory and incorporate the series' autocorrelation. There are more complex machine learning methods which by construction are capable of incorporating the lag structure of time series data. Recurrent Neural Network (RNN) with their most recent variant Long Short-term Memory (LSTM), for example, are one possible framework that could produce even more accurate macroeconomic forecasts. However, their application often comes at the cost of interpretability. If the complexity of a model is so high that it indeed is no more than a 'black box' to the user, the costs of potential accuracy gains become too heavy. For this reason, the author sees the major challenge of future research to align economic theory with machine learning based frameworks which are designed within reasonable bounds of complexity.

Moreover, future research should expand the analysis of machine learning based GDP growth forecasting to a wider set of countries. As this study has shown, the application is promising for U.S. data. It remains to be evaluated how the same analysis performs on macroeconomic data of other countries. Jung

et al. (2018), for example, suggest that the application on the United Kingdom, Germany, Spain, Mexico, Philippines and Vietnam also leads to forecasting improvements. Yet, an even wider set of countries should be considered to validate the results.

Furthermore, this analysis refrains from the question of how to train forecasting algorithms most effectively in light of structural breaks. Preferably, a machine learning method should only be trained with data realized after the last major structural break. One could think of learning methods which are constructed in a way that allows the algorithm to detect structural breaks in the time series data. First attempts towards this direction have been made in an OECD working paper which introduces adaptive trees that, in case of a structural break, attaches greater weight to more recent training realizations (Woloszko, 2018). This is another field which should be emphasized more strongly by future research.

Finally, one issue that could not be solved by this paper is the hurdle of data revisions common in research that is based on macroeconomic data. Comparing the performance of a forecasting model which has been used at different points in time is only possible if real-time data is available for all variables included in the model. Machine learning methods tend to operate on large datasets with many variables. Such methods require clear data management strategies which ensure that for all variables each reporting vintage is stored and available to researchers. Only in this way can macroeconomics produce more nuanced research results in the future.

In conclusion, this paper contributes to macroeconomic research by demonstrating that machine learning provides useful techniques that can help with predicting future economic development and potentially other yet unresolved research problems. In light of these results, the author advocates more interdisciplinarity in the field of economics to make the best use of these rather new techniques. Researchers should not waste this opportunity and let the call for new approaches in economics go unheard.

References

Armstrong, J. S. (2001). Evaluating forecasting methods. In *Principles of forecasting* (pp. 443–472). Boston, MA: Springer.

Armstrong, J. S., & Collopy, F. (1992). Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting*, *8*(1), 69–80.

Athey, S., & Imbens, G. W. (2017). The state of applied econometrics: Causality and policy evaluation. *Journal of Economic Perspectives*, *31*(2), 3–32.

Bank of England. (2016). *The dappled world*. Retrieved from https://bankofengland.co.uk/-/media/boe/files/speech/2016/the-dappled-world.pdf

Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, *191*, 192–213.

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, *120*, 70–83.

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

Bernanke, B. S., & Boivin, J. (2003). Monetary policy in a data-rich environment. *Journal of Monetary Economics*, *50*(3), 525–546.

Bernanke, B. S., Boivin, J., & Eliasz, P. (2005). Measuring the effects of monetary policy: A factor-augmented vector autoregressive (FAVAR) approach. *The Quarterly Journal of Economics*, *120*(1), 387–422.

Besley, T., & Hennessy, P. (2009). The global financial crisis - why didn't anybody notice? *British Academy Review*, *14*, 8–10.

Bhattacharyya, S. (2018). *Support Vector Machine: Kernel Trick; Mercer's Theorem*. Retrieved from https://towardsdatascience.com/understanding-support-vector-machine-part-2-kernel-trick-mercers-theorem-e1e6848c6c4d

Biau, O., & D'Elia, A. (2010). *Euro area GDP forecasting using large survey datasets: A random forest approach*. Unpublished Paper, European Commission.

Blanchard, O. (2014). Where danger lurks. *Finance & Development*, *51*(3), 28–31.

Box, G. E. P., & Jenkins, G. M. (1970). *Time series analysis: Forecasting and control* (1st ed.). San Francisco, CA: Holden-Day.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2016). *Time series analysis: Forecasting and control* (5th ed.). Hoboken, NJ: John Wiley & Sons.

Box, G. E. P., & Pierce, D. A. (1970). Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association*, *65*(332), 1509–1526.

Breiman, L. (2001a). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L. (2001b). Statistical modeling: The two cultures. *Statistical Science*, *16*(3), 199–231.

Buchen, T., & Wohlrabe, K. (2011). Forecasting with many predictors: Is boosting a viable alternative? *Economics Letters*, *113*(1), 16–18.

Bureau of Economic Analysis. (2016). *Concepts and methods of the US National Income and Product Accounts*. Retrieved from https://bea.gov/system/files/2019-12/All-Chapters.pdf

Chauvet, M., & Potter, S. (2013). Forecasting output. In *Handbook of economic forecasting* (Vol. 2, pp. 141–194). Amsterdam, Netherlands: Elsevier.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). New York, NY: ACM.

Cherkassky, V., & Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, *17*(1), 113–126.

Clements, M. P., & Hendry, D. F. (2011). Introduction. In *The oxford handbook of economic forecasting* (pp. 1–6). New York, NY: Oxford University Press.

Cowan, B. D., Smith, S., & Thompson, S. (2018). Seasonal adjustment in the National Income and Product Accounts: Results from the 2018 comprehensive update. *Survey of Current Business*, *98*(8), 1–15.

Crone, S. F., Hibon, M., & Nikolopoulos, K. (2011). Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting*, *27*(3), 635–660.

Croushore, D. D., & Stark, T. (2000). A funny thing happened on the way to the data bank: A real-time data set for macroeconomists. *Federal Reserve Bank of Philadelphia - Business Review*, *5*, 15–27.

Diebold, F. X., & Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *20*(1), 134–144.

Döpke, J., Fritsche, U., & Pierdzioch, C. (2017). Predicting recessions with boosted regression trees. *International Journal of Forecasting*, *33*(4), 745–759.

Dubey, A. (2018). *The mathematics behind Principal Component Analysis*. Retrieved from https://towardsdatascience.com/the-mathematics-behind-principal-component-analysis-fff2d7f4b643

Fildes, R., & Stekler, H. (2002). The state of macroeconomic forecasting. *Journal of Macroeconomics*, *24*(4), 435–468.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232.

Galí, J. (2018). The state of New Keynesian Economics: A partial assessment. *Journal of Economic Perspectives*, *32*(3), 87–112.

Giacomini, R. (2013). The relationship between DSGE and VAR models. In *VAR models in macroeconomics - new developments and applications: Essays in honor of Christopher A. Sims (Advances in Econometrics, 32)* (pp. 1–25). Bingley, England: Emerald Group Publishing.

Giacomini, R. (2015). Economic theory and forecasting: Lessons from the literature. *The Econometrics Journal*, *18*(2), 22–41.

Gogas, P., Papadimitriou, T., Matthaiou, M., & Chrysanthidou, E. (2015). Yield curve and recession forecasting in a machine learning framework. *Computational Economics*, *45*(4), 635–645.

Gogas, P., Papadimitriou, T., & Takli, E. (2013). Comparison of simple sum and Divisia monetary aggregates in GDP forecasting: A support vector machines approach. *Economics Bulletin*, *33*(2), 1101–1115.

Harrell, F. (2019). *Road map for choosing between statistical modeling and machine learning*. Retrieved from https://fharrell.com/post/stat-ml

Harvey, D., Leybourne, S., & Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, *13*(2), 281–291.

Hassania, H., & Silva, E. S. (2015). Forecasting with big data: A review. *Annals of Data Science*, *2*(1), 5–19.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction* (2nd ed.). New York, NY: Springer Science & Business Media.

Ho, T. K. (1995). Random decision forests. In *Proceedings of the third international conference on document analysis and recognition* (pp. 278–282). New York, NY: IEEE.

Hurvich, C. M., & Tsai, C.-L. (1989). Regression and time series model selection in small samples. *Biometrika*, *76*(2), 297–307.

Hyndman, R. J. (2010). *Benchmarks for forecasting*. Retrieved from https://robjhyndman.com/hyndsight/benchmarks

Hyndman, R. J. (2014). *Thoughts on the Ljung-Box test*. Retrieved from https://robjhyndman.com/hyndsight/ljung-box-test

Hyndman, R. J. (2016). *Cross-validation for time series*. Retrieved from https://robjhyndman.com/hyndsight/tscv

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.). Melbourne, Australia: OTexts.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, *27*(3), 1–22.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in R*. New York, NY: Springer Science & Business Media.

Jung, J.-K., Patnam, M., & Ter-Martirosyan, A. (2018). An algorithmic crystal ball: Forecasts-based on machine learning. *IMF Working Paper*, (18/230).

Karush, W. (2013). Minima of functions of several variables with inequalities as side conditions. In *Traces and emergence of nonlinear programming* (pp. 217–245). Basel, Switzerland: Springer Science & Business Media.

Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., & Murthy, K. R. K. (2001). Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, *13*(3), 637–649.

Kock, A. B., & Teräsvirta, T. (2011). Forecasting with nonlinear time series models. In *The oxford handbook of economic forecasting* (pp. 61–87). New York, NY: Oxford University Press.

Kuhn, H. W., & Tucker, A. W. (1951). Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on mathematical statistics and probability* (pp. 481–492). Berkeley, CA: University of California Press.

Landefeld, J. S., Moulton, B. R., & Vojtech, C. M. (2003). Chained-dollar indexes: Issues, tips on their use, and upcoming changes. *Survey of Current Business*, *83*(11), 8–16.

Lehmann, R., & Wohlrabe, K. (2016). Looking into the black box of boosting: The case of Germany. *Applied Economics Letters*, *23*(17), 1229–1233.

Lehmann, R., & Wohlrabe, K. (2017). Boosting and regional economic forecasting: The case of Germany. *Letters in Spatial and Resource Sciences*, *10*(2), 161–175.

Li, C. (2016). *A gentle introduction to gradient boosting*. Retrieved from http://ccs.neu.edu/home/vip/teach/MLcourse/4_boosting/slides/gradient_boosting.pdf

McCracken, M. W., & Ng, S. (2016). FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, *34*(4), 574–589.

Mullainathan, S., & Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, *31*(2), 87–106.

Müller, K.-R., Smola, A. J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., & Vapnik, V. (1997). Predicting time series with support vector machines. In *International conference on artificial neural networks* (pp. 999–1004). Berlin, Heidelberg, Germany: Springer.

National Bureau of Economic Research. (2012). *US business cycle expansions and contractions*. Retrieved from https://nber.org/cycles/US_Business_Cycle_Expansions_and_Contractions_20120423.pdf

Ng, S. (2014). Boosting recessions. *Canadian Journal of Economics*, *47*(1), 1–34.

OECD. (2012). *OECD system of composite leading indicators*. Retrieved from http://oecd.org/sdd/leading-indicators/41629509.pdf

OECD. (2019). *OECD composite leading indicators: Turning points of reference series and component series*. Retrieved from http://oecd.org/sdd/leading-indicators/CLI-components-and-turning-points.pdf

Pagan, A. (2003). Report on modelling and forecasting at the Bank of England. *Bank of England - Quarterly Bulletin*, *43*(1), 60–91.

Palachy, S. (2019). *Stationarity in time series analysis*. Retrieved from https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2019). Cross-validation: Evaluating estimator performance. In *Scikit-learn user guide* (Vols. 0.22, pp. 460–474).

Reis, R. (2018). Is something really wrong with macroeconomics? *Oxford Review of Economic Policy*, *34*(1–2), 132–155.

Romer, P. (2016). The trouble with macroeconomics. *The American Economist*. Advance online publication. Retrieved from https://ccl.yale.edu/sites/default/files/files/The%20Trouble%20with%20Macroeconomics.pdf

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*, 199–222.

Stock, J. H., & Watson, M. W. (2001). Vector autoregressions. *Journal of Economic Perspectives*, *15*(4), 101–115.

Stock, J. H., & Watson, M. W. (2002). Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, *20*(2), 147–162.

Stock, J. H., & Watson, M. W. (2005). *Implications of dynamic factor models for VAR analysis*. Unpublished Paper, Princeton University.

Stock, J., & Watson, M. (2012). Disentangling the channels of the 2007-2009 recession. *Brookings Papers on Economic Activity*, *2012*(1), 81–135.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*, *16*(4), 437–450.

Tenhofen, J., Guntram, W. B., & Heppke-Falk, K. H. (2010). The macroeconomic effects of exogenous fiscal policy shocks in Germany: A disaggregated SVAR analysis. *Journal of Economics and Statistics (Jahrbücher für Nationalökonomie und Statistik)*, *230*(3), 328–355.

Tiffin, A. (2016). Seeing in the dark: A machine-learning approach to nowcasting in Lebanon. *IMF Working Paper*, (16/56).

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

Vapnik, V. (2013). *The nature of statistical learning theory*. New York, NY: Springer Science & Business Media.

Vapnik, V., Golowich, S. E., & Smola, A. J. (1996). Support vector method for function approximation, regression estimation and signal processing. In *Advances in neural information processing systems 9* (pp. 281–287). Cambridge, MA: MIT Press.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives*, *28*(2), 3–28.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, *7*(91).

Verbeek, M. (2004). *A guide to modern econometrics* (2nd ed.). West Sussex, England: John Wiley & Sons.

Wang, W., Xu, Z., Lu, W., & Zhang, X. (2003). Determination of the spread parameter in the Gaussian kernel for classification and regression. *Neurocomputing*, *55*(3–4), 643–663.

Woloszko, N. (2018). Adaptive Trees: A novel approach to macroeconomic forecasting. Retrieved from https://techpolicyinstitute.org/wp-content/uploads/2017/12/Woloszko_Forecasting-GDP-growth-with-adaptive-trees-002.pdf

Appendices

A FRED-QD Variables

Table 9: Feature Overview

Variable	Description	Trans-Code	S&W Factors
Group A: NIPA			
PCECC96	Real Personal Consumption Expenditures (Billions of Chained 2012 Dollars)	5	
PCD_GX	Real personal consumption expenditures: Durable goods (Billions of Chained 2012 Dollars), deflated using PCE	5	×
PCES_VX	Real Personal Consumption Expenditures: Services (Billions of 2012 Dollars), deflated using PCE	5	×
PCN_DX	Real Personal Consumption Expenditures: Nondurable Goods (Billions of 2012 Dollars), deflated using PCE	5	×
GPDIC1	Real Gross Private Domestic Investment, 3 decimal (Billions of Chained 2012 Dollars)	5	
FP_IX	Real private fixed investment (Billions of Chained 2012 Dollars), deflated using PCE	5	
Y033RC1Q027SBE_AX	Real Gross Private Domestic Investment: Fixed Investment: Nonresidential: Equipment (Billions of Chained 2012 Dollars), deflated using PCE	5	×
PNF_IX	Real private fixed investment: Nonresidential (Billions of Chained 2012 Dollars), deflated using PCE	5	×
PRF_IX	Real private fixed investment: Residential (Billions of Chained 2012 Dollars), deflated using PCE	5	×
A014RE1Q156NBEA	Shares of gross domestic product: Gross private domestic investment: Change in private inventories (Percent)	2	×
GCEC1	Real Government Consumption Expenditures & Gross Investment (Billions of Chained 2012 Dollars)	5	
A823RL1Q225SBEA	Real Government Consumption Expenditures and Gross Investment: Federal (Percent Change from Preceding Period)	1	×
FGRECP_TX	Real Federal Government Current Receipts (Billions of Chained 2012 Dollars), deflated using PCE	5	×
SLC_EX	Real government state and local consumption expenditures (Billions of Chained 2012 Dollars), deflated using PCE	5	×
EXPGSC1	Real Exports of Goods & Services, 3 Decimal (Billions of Chained 2012 Dollars)	5	×
IMPGSC1	Real Imports of Goods & Services, 3 Decimal (Billions of Chained 2012 Dollars)	5	×
DPIC96	Real Disposable Personal Income (Billions of Chained 2012 Dollars)	5	
OUTNFB	Nonfarm Business Sector: Real Output (Index 2012=100)	5	
OUTBS	Business Sector: Real Output (Index 2012=100)	5	
B020RE1Q156NBEA	Shares of gross domestic product: Exports of goods and services (Percent)	2	
B021RE1Q156NBEA	Shares of gross domestic product: Imports of goods and services (Percent)	2	
Group B: Industrial Production			
INDPRO	Industrial Production Index (Index 2012=100)	5	
IPFINAL	Industrial Production: Final Products (Market Group) (Index 2012=100)	5	
IPCONGD	Industrial Production: Consumer Goods (Index 2012=100)	5	
IPMAT	Industrial Production: Materials (Index 2012=100)	5	
IPDMAT	Industrial Production: Durable Materials (Index 2012=100)	5	×
IPNMAT	Industrial Production: Nondurable Materials (Index 2012=100)	5	×
IPDCONGD	Industrial Production: Durable Consumer Goods (Index 2012=100)	5	×
IPB51110SQ	Industrial Production: Durable Goods: Automotive products (Index 2012=100)	5	×
IPNCONGD	Industrial Production: Nondurable Consumer Goods (Index 2012=100)	5	×
IPBUSEQ	Industrial Production: Business Equipment (Index 2012=100)	5	×
IPB51220SQ	Industrial Production: Consumer energy products (Index 2012=100)	5	×
CUMFNS	Capacity Utilization: Manufacturing (SIC) (Percent of Capacity)	2	×
IPMANSICS	Industrial Production: Manufacturing (SIC) (Index 2012=100)	5	
IPB51222S	Industrial Production: Residential Utilities (Index 2012=100)	5	
IPFUELS	Industrial Production: Fuels (Index 2012=100)	5	
Group C: Employment and Unemployment			
PAYEMS	All Employees: Total nonfarm (Thousands of Persons)	5	
USPRIV	All Employees: Total Private Industries (Thousands of Persons)	5	
MANEMP	All Employees: Manufacturing (Thousands of Persons)	5	
SRVPRD	All Employees: Service-Providing Industries (Thousands of Persons)	5	
USGOOD	All Employees: Goods-Producing Industries (Thousands of Persons)	5	
DMANEMP	All Employees: Durable goods (Thousands of Persons)	5	×
NDMANEMP	All Employees: Nondurable goods (Thousands of Persons)	5	
USCONS	All Employees: Construction (Thousands of Persons)	5	×
USEHS	All Employees: Education & Health Services (Thousands of Persons)	5	×
USFIRE	All Employees: Financial Activities (Thousands of Persons)	5	×
USINFO	All Employees: Information Services (Thousands of Persons)	5	×
USPBS	All Employees: Professional & Business Services (Thousands of Persons)	5	×
USLAH	All Employees: Leisure & Hospitality (Thousands of Persons)	5	×
USSERV	All Employees: Other Services (Thousands of Persons)	5	×
USMINE	All Employees: Mining and logging (Thousands of Persons)	5	×
USTPU	All Employees: Trade, Transportation & Utilities (Thousands of Persons)	5	×
USGOVT	All Employees: Government (Thousands of Persons)	5	
USTRADE	All Employees: Retail Trade (Thousands of Persons)	5	×
USWTRADE	All Employees: Wholesale Trade (Thousands of Persons)	5	×
CES9091000001	All Employees: Government: Federal (Thousands of Persons)	5	×
CES9092000001	All Employees: Government: State Government (Thousands of Persons)	5	×
CES9093000001	All Employees: Government: Local Government (Thousands of Persons)	5	×

CE16OV	Civilian Employment (Thousands of Persons)	5	
CIVPART	Civilian Labor Force Participation Rate (Percent)	2	
UNRATE	Civilian Unemployment Rate (Percent)	2	
UNRATES_TX	Unemployment Rate less than 27 weeks (Percent)	2	
UNRATEL_TX	Unemployment Rate for more than 27 weeks (Percent)	2	
LNS14000012	Unemployment Rate - 16 to 19 years (Percent)	2	×
LNS14000025	Unemployment Rate - 20 years and over, Men (Percent)	2	×
LNS14000026	Unemployment Rate - 20 years and over, Women (Percent)	2	×
UEMPLT5	Number of Civilians Unemployed - Less Than 5 Weeks (Thousands of Persons)	5	×
UEMP5TO14	Number of Civilians Unemployed for 5 to 14 Weeks (Thousands of Persons)	5	×
UEMP15T26	Number of Civilians Unemployed for 15 to 26 Weeks (Thousands of Persons)	5	×
UEMP27OV	Number of Civilians Unemployed for 27 Weeks and Over (Thousands of Persons)	5	×
LNS12032194	Employment Level - Part-Time for Economic Reasons, All Industries (Thousands of Persons)	5	×
HOABS	Business Sector: Hours of All Persons (Index 2012=100)	5	
HOANBS	Nonfarm Business Sector: Hours of All Persons (Index 2012=100)	5	
AWHMAN	Average Weekly Hours of Production and Nonsupervisory Employees: Manufacturing (Hours)	5	×
AWOTMAN	Average Weekly Overtime Hours of Production and Nonsupervisory Employees: Manufacturing (Hours)	2	×
UEMPMEAN	Average (Mean) Duration of Unemployment (Weeks)	2	
CES0600000007	Average Weekly Hours of Production and Nonsupervisory Employees: Goods-Producing	2	
HWIURATI_OX	Ratio of Help Wanted/No. Unemployed	2	
CLAIM_SX	Initial Claims	5	
Group D: Housing			
HOUST	Housing Starts: Total: New Privately Owned Housing Units Started (Thousands of Units)	5	
HOUST5F	Privately Owned Housing Starts: 5-Unit Structures or More (Thousands of Units)	5	
HOUSTMW	Housing Starts in Midwest Census Region (Thousands of Units)	5	×
HOUSTNE	Housing Starts in Northeast Census Region (Thousands of Units)	5	×
HOUSTS	Housing Starts in South Census Region (Thousands of Units)	5	×
HOUSTW	Housing Starts in West Census Region (Thousands of Units)	5	×
Group E: Inventories, Orders, and Sales			
CMRMTSP_LX	Real Manufacturing and Trade Industries Sales (Millions of Chained 2012 Dollars)	5	
RSAF_SX	Real Retail and Food Services Sales (Millions of Chained 2012 Dollars), deflated by Core PCE	5	×
AMDMN_OX	Real Manufacturers' New Orders: Durable Goods (Millions of 2012 Dollars), deflated by Core PCE	5	×
AMDMU_OX	Real Value of Manufacturers' Unfilled Orders for Durable Goods Industries (Millions of 2012 Dollars), deflated by Core PCE	5	×
BUSIN_VX	Total Business Inventories (Millions of Dollars)	5	
ISRATI_OX	Total Business: Inventories to Sales Ratio	2	
Group F: Prices			
PCECTPI	Personal Consumption Expenditures: Chain-type Price Index (Index 2012=100)	6	
PCEPILFE	Personal Consumption Expenditures Excluding Food and Energy (Chain-Type Price Index) (Index 2012=100)	6	
GDPCTPI	Gross Domestic Product: Chain-type Price Index (Index 2012=100)	6	
GPDICTPI	Gross Private Domestic Investment: Chain-type Price Index (Index 2012=100)	6	×
IPDBS	Business Sector: Implicit Price Deflator (Index 2012=100)	6	×
DGDSRG3Q086SBEA	Personal consumption expenditures: Goods (chain-type price index)	6	
DDURRG3Q086SBEA	Personal consumption expenditures: Durable goods (chain-type price index)	6	
DSERRG3Q086SBEA	Personal consumption expenditures: Services (chain-type price index)	6	
DNDGRG3Q086SBEA	Personal consumption expenditures: Nondurable goods (chain-type price index)	6	
DHCERG3Q086SBEA	Personal consumption expenditures: Services: Household consumption expenditures (chain-type price index)	6	
DMOTRG3Q086SBEA	Personal consumption expenditures: Durable goods: Motor vehicles and parts (chain-type price index)	6	×
DFDHRG3Q086SBEA	Personal consumption expenditures: Durable goods: Furnishings and durable household equipment (chain-type price index)	6	×
DREQRG3Q086SBEA	Personal consumption expenditures: Durable goods: Recreational goods and vehicles (chain-type price index)	6	×
DODGRG3Q086SBEA	Personal consumption expenditures: Durable goods: Other durable goods (chain-type price index)	6	×
DFXARG3Q086SBEA	Personal consumption expenditures: Nondurable goods: Food and beverages purchased for off-premises consumption (chain-type price index)	6	×
DCLORG3Q086SBEA	Personal consumption expenditures: Nondurable goods: Clothing and footwear (chain-type price index)	6	×
DGOERG3Q086SBEA	Personal consumption expenditures: Nondurable goods: Gasoline and other energy goods (chain-type price index)	6	×
DONGRG3Q086SBEA	Personal consumption expenditures: Nondurable goods: Other nondurable goods (chain-type price index)	6	×
DHUTRG3Q086SBEA	Personal consumption expenditures: Services: Housing and utilities (chain-type price index)	6	×
DHLCRG3Q086SBEA	Personal consumption expenditures: Services: Health care (chain-type price index)	6	×
DTRSRG3Q086SBEA	Personal consumption expenditures: Transportation services (chain-type price index)	6	×
DRCARG3Q086SBEA	Personal consumption expenditures: Recreation services (chain-type price index)	6	×
DFSARG3Q086SBEA	Personal consumption expenditures: Services: Food services and accommodations (chain-type price index)	6	×

DIFSRG3Q086SBEA	Personal consumption expenditures: Financial services and insurance (chain-type price index)	6	×
DOTSRG3Q086SBEA	Personal consumption expenditures: Other services (chain-type price index)	6	×
CPIAUCSL	Consumer Price Index for All Urban Consumers: All Items (Index 1982-84=100)	6	
CPILFESL	Consumer Price Index for All Urban Consumers: All Items Less Food & Energy (Index 1982-84=100)	6	
WPSFD49207	Producer Price Index by Commodity for Finished Goods (Index 1982=100)	6	
PPIACO	Producer Price Index for All Commodities (Index 1982=100)	6	
WPSFD49502	Producer Price Index by Commodity for Finished Consumer Goods (Index 1982=100)	6	×
WPSFD4111	Producer Price Index by Commodity for Finished Consumer Foods (Index 1982=100)	6	×
PPIIDC	Producer Price Index by Commodity Industrial Commodities (Index 1982=100)	6	×
WPSID61	Producer Price Index by Commodity Intermediate Materials: Supplies & Components (Index 1982=100)	6	×
WPU0561	Producer Price Index by Commodity for Fuels and Related Products and Power: Crude Petroleum (Domestic Production) (Index 1982=100)	5	×
OILPRIC_EX	Price Real Crude Oil Prices: West Texas Intermediate (WTI) - Cushing, Oklahoma (2012 Dollars per Barrel), deflated by Core PCE	5	
WPSID62	Price Index: Crude Materials for Further Processing (Index 1982=100)	6	
PPICMM	Price Index: Commodities: Metals and metal products: Primary nonferrous metals (Index 1982=100)	6	
CPIAPPSL	Price Index for All Urban Consumers: Apparel (Index 1982-84=100)	6	
CPITRNSL	Price Index for All Urban Consumers: Transportation (Index 1982-84=100)	6	
CPIMEDSL	Price Index for All Urban Consumers: Medical Care (Index 1982-84=100)	6	
CUSR0000SAC	Price Index for All Urban Consumers: Commodities (Index 1982-84=100)	6	
CUSR0000SAD	Consumer Price Index for All Urban Consumers: Durables (Index 1982-84=100)	6	
CUSR0000SAS	Consumer Price Index for All Urban Consumers: Services (Index 1982-84=100)	6	
CPIULFSL	Consumer Price Index for All Urban Consumers: All Items Less Food (Index 1982-84=100)	6	
CUSR0000SA0L2	Consumer Price Index for All Urban Consumers: All items less shelter (Index 1982-84=100)	6	
CUSR0000SA0L5	Consumer Price Index for All Urban Consumers: All items less medical care (Index 1982-84=100)	6	

Group G: Earnings and Productivity

CES2000000008X	Real Average Hourly Earnings of Production and Nonsupervisory Employees: Construction (2012 Dollars per Hour), deflated by Core PCE	5	
CES3000000008X	Real Average Hourly Earnings of Production and Nonsupervisory Employees: Manufacturing (2012 Dollars per Hour), deflated by Core PCE	5	
COMPRNFB	Nonfarm Business Sector: Real Compensation Per Hour (Index 2012=100)	5	×
RCPHBS	Business Sector: Real Compensation Per Hour (Index 2012=100)	5	×
OPHNFB	Nonfarm Business Sector: Real Output Per Hour of All Persons (Index 2012=100)	5	×
OPHPBS	Business Sector: Real Output Per Hour of All Persons (Index 2012=100)	5	
ULCBS	Business Sector: Unit Labor Cost (Index 2012=100)	5	
ULCNFB	Nonfarm Business Sector: Unit Labor Cost (Index 2012=100)	5	×
UNLPNBS	Nonfarm Business Sector: Unit Nonlabor Payments (Index 2012=100)	5	×
CES0600000008	Average Hourly Earnings of Production and Nonsupervisory Employees: Goods-Producing (Dollars per Hour)	6	

Group H: Interest Rates

FEDFUNDS	Effective Federal Funds Rate (Percent) Federal Funds Rate (Percent)	2	×
TB3MS	3-Month Treasury Bill: Secondary Market Rate (Percent)	2	×
TB6MS	6-Month Treasury Bill: Secondary Market Rate (Percent)	2	
GS1	1-Year Treasury Constant Maturity Rate (Percent)	2	
GS10	10-Year Treasury Constant Maturity Rate (Percent)	2	
AAA	Bond Moody's Seasoned Aaa Corporate Bond Yield© (Percent)	2	
BAA	Bond Moody's Seasoned Baa Corporate Bond Yield© (Percent)	2	
BAA10YM	Moody's Seasoned Baa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity (Percent)	5	×
TB6M3MX	6-Month Treasury Bill Minus 3-Month Treasury Bill, secondary market (Percent)	1	×
GS1TB3MX	1-Year Treasury Constant Maturity Minus 3-Month Treasury Bill, secondary market (Percent)	2	×
GS10TB3MX	10-Year Treasury Constant Maturity Minus 3-Month Treasury Bill, secondary market (Percent)	2	×
CPF3MTB3MX	3-Month Commercial Paper Minus 3-Month Treasury Bill, secondary market (Percent)	2	×
GS5	5-Year Treasury Constant Maturity Rate	2	
TB3SMFFM	3-Month Treasury Constant Maturity Minus Federal Funds Rate	2	
T5YFFM	5-Year Treasury Constant Maturity Minus Federal Funds Rate	1	
AAAFFM	Moody's x Aaa Corporate Bond Minus Federal Funds Rate	2	
CP3M	3-Month AA Financial Commercial Paper Rate	2	
COMPAPFF	3-Month Commercial Paper Minus Federal Funds Rate	2	

Group I: Money and Credit

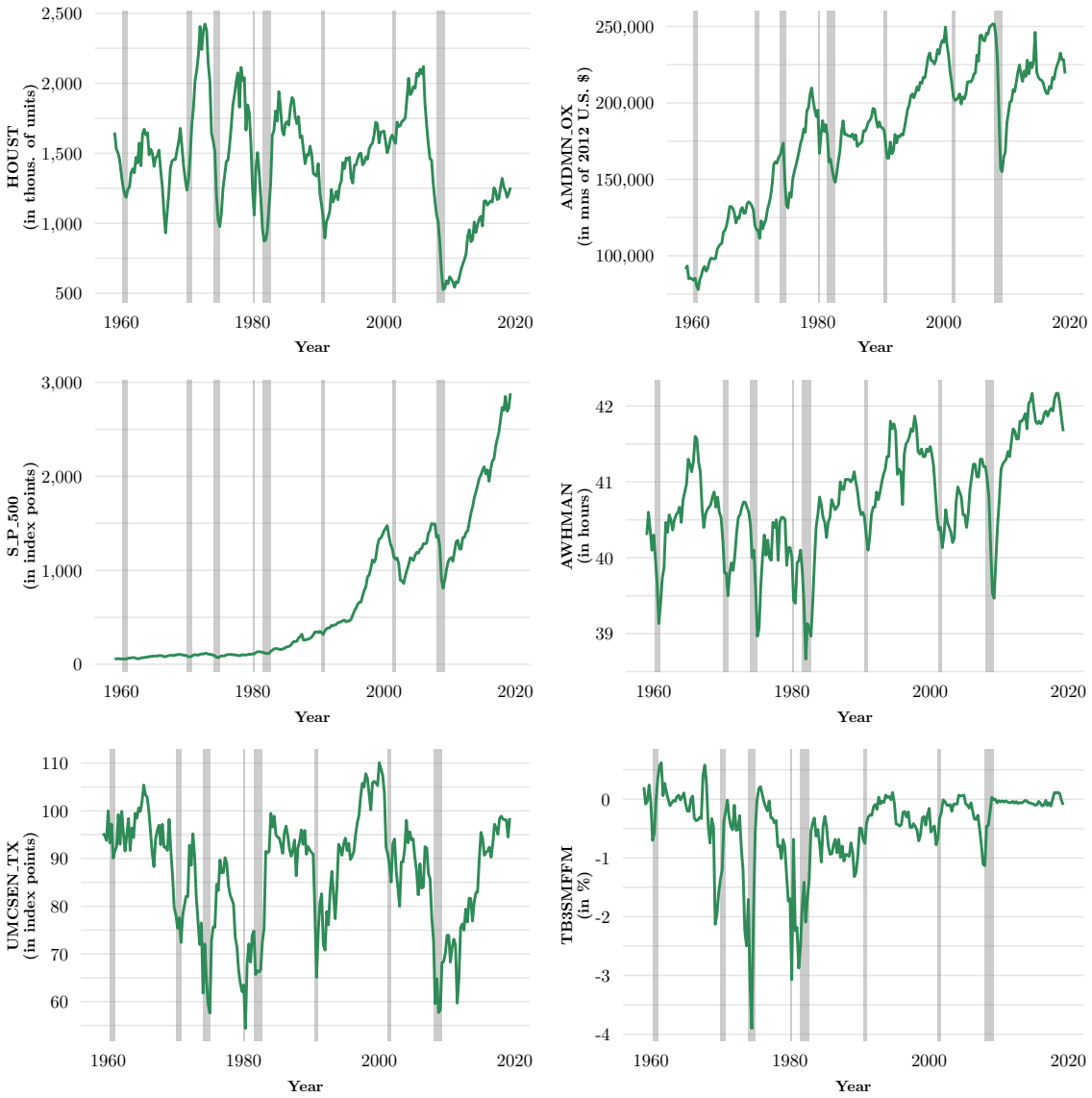
AMBSLREAL	St. Louis Adjusted Monetary Base (Billions of 1982-84 Dollars), deflated by CPI	5	
M1REAL	Real M1 Money Stock (Billions of 1982-84 Dollars), deflated by CPI	5	
M2REAL	Real M2 Money Stock (Billions of 1982-84 Dollars), deflated by CPI	5	
MZMREAL	Real MZM Money Stock (Billions of 1982-84 Dollars), deflated by CPI	5	
BUSLOAN_SX	Real Commercial and Industrial Loans, All Commercial Banks (Billions of 2012 U.S. Dollars), deflated by Core PCE	5	×
CONSUME_RX	Real Consumer Loans at All Commercial Banks (Billions of 2012 U.S. Dollars), deflated by Core PCE	5	×
NONREVS_LX	Total Real Nonrevolving Credit Owned and Securitized, Outstanding (Billions of 2012 Dollars), deflated by Core PCE	5	×

REALL_NX	Real Real Estate Loans, All Commercial Banks (Billions of 2012 U.S. Dollars), deflated by Core PCE	5	×
TOTALS_LX	Total Consumer Credit Outstanding (Billions of 2012 Dollars), deflated by Core PCE	5	
TOTRESNS	Total Reserves of Depository Institutions (Billions of Dollars)	6	
NONBORRES	Reserves Of Depository Institutions, Nonborrowed (Millions of Dollars)	7	
DTCOLNVHFNM	Consumer Motor Vehicle Loans Outstanding Owned by Finance Companies (Millions of Dollars)	6	
DTCTHFNM	Total Consumer Loans and Leases Outstanding Owned and Securitized by Finance Companies (Millions of Dollars)	6	
INVEST	Securities in Bank Credit at All Commercial Banks (Billions of Dollars)	6	
Group J: Household Balance Sheets			
TABSHN_OX	Real Total Assets of Households and Nonprofit Organizations (Billions of 2012 Dollars), deflated by Core PCE	5	
TLBSHN_OX	Real Total Liabilities of Households and Nonprofit Organizations (Billions of 2012 Dollars), deflated by Core PCE	5	×
LIABP_IX	Liabilities of Households and Nonprofit Organizations Relative to Personal Disposable Income (Percent)	5	
TNWBSHN_OX	Real Net Worth of Households and Nonprofit Organizations (Billions of 2012 Dollars), deflated by Core PCE	5	×
NWP_IX	Net Worth of Households and Nonprofit Organizations Relative to Disposable Personal Income (Percent)	5	
TARES_AX	Real Assets of Households and Nonprofit Organizations excluding Real Estate Assets (Billions of 2012 Dollars), deflated by Core PCE	5	×
HNOREMQ027SX	Real Real Estate Assets of Households and Nonprofit Organizations (Billions of 2012 Dollars), deflated by Core PCE	5	×
TFAABSHN_OX	Real Total Financial Assets of Households and Nonprofit Organizations (Billions of 2012 Dollars), deflated by Core PCE	5	×
CONSP_IX	Nonrevolving consumer credit to Personal Income	2	
Group K: Exchange Rates			
EXSZU_SX	Switzerland / U.S. Foreign Exchange Rate	5	×
EXJPU_SX	Japan / U.S. Foreign Exchange Rate	5	×
EXUSU_KX	U.S. / U.K. Foreign Exchange Rate	5	×
EXCAU_SX	Canada / U.S. Foreign Exchange Rate	5	×
Group L: Other			
UMCSEN_TX	University of Michigan: Consumer Sentiment (Index 1st Quarter 1966=100)	5	×
Group M: Stock Markets			
NIKKEI225	Nikkei Stock Average	5	
S_P_500	S&P's Common Stock Price Index: Composite	5	
S_P_INDUST	S&P's Common Stock Price Index: Industrials	5	
S_P_DIV_YIELD	S&P's Composite Common Stock: Dividend Yield	2	
S_P_PE_RATIO	S&P's Composite Common Stock: Price-Earnings Ratio	5	
Group N: Non-Household Balance Sheets			
TLBSNNC_BX	Real Nonfinancial Corporate Business Sector Liabilities (Billions of 2012 Dollars), Deflated by Implicit Price Deflator for Business Sector IPDBS	5	
TLBSNNCBBD_IX	Nonfinancial Corporate Business Sector Liabilities to Disposable Business Income (Percent)	2	
TTAABSNNC_BX	Real Nonfinancial Corporate Business Sector Assets (Billions of 2012 Dollars), Deflated by Implicit Price Deflator for Business Sector IPDBS	5	
TNWMVBSNNC_BX	Real Nonfinancial Corporate Business Sector Net Worth (Billions of 2012 Dollars), Deflated by Implicit Price Deflator for Business Sector IPDBS	5	
TNWMVBSNNCBBD_IX	Nonfinancial Corporate Business Sector Net Worth to Disposable Business Income (Percent)	2	
TLBSNN_BX	Real Nonfinancial Noncorporate Business Sector Liabilities (Billions of 2012 Dollars), Deflated by Implicit Price Deflator for Business Sector IPDBS	5	
TLBSNNBBD_IX	Nonfinancial Noncorporate Business Sector Liabilities to Disposable Business Income (Percent)	2	
TABSNN_BX	Real Nonfinancial Noncorporate Business Sector Assets (Billions of 2012 Dollars), Deflated by Implicit Price Deflator for Business Sector IPDBS	5	
TNWBSNN_BX	Real Nonfinancial Noncorporate Business Sector Net Worth (Billions of 2012 Dollars), Deflated by Implicit Price Deflator for Business Sector IPDBS	5	
TNWBSNNBBD_IX	Nonfinancial Noncorporate Business Sector Net Worth to Disposable Business Income (Percent)	2	
CNC_FX	Real Disposable Business Income, Billions of 2012 Dollars (Corporate cash flow with IVA minus taxes on corporate income, deflated by Implicit Price Deflator for Business Sector IPDBS)	5	

Note: Column Trans-Code refers to the final data transformation conducted to obtain stationary time series: (1) no transformation, (2) first-order differencing Δx_t, (3) second-order differencing $\Delta^2 x_t$, (4) natural logarithm $log(x_t)$, (5) first-order differencing of natural logarithm (continuous growth) $\Delta log(x_t)$, (6) second-order differencing of natural logarithm $\Delta^2 log(x_t)$, (7) discrete growth $(x_t - x_{t-1}) / x_{t-1}$. Transformations closely follow the suggestions in McCracken and Ng (2016) but are adjusted to ensure stationarity where necessary.
Column S&W Factor indicates which of the variables are considered in the S&W FAVAR model. This feature selection follows the factor analysis in J. Stock and Watson (2012).

B Leading Indicators

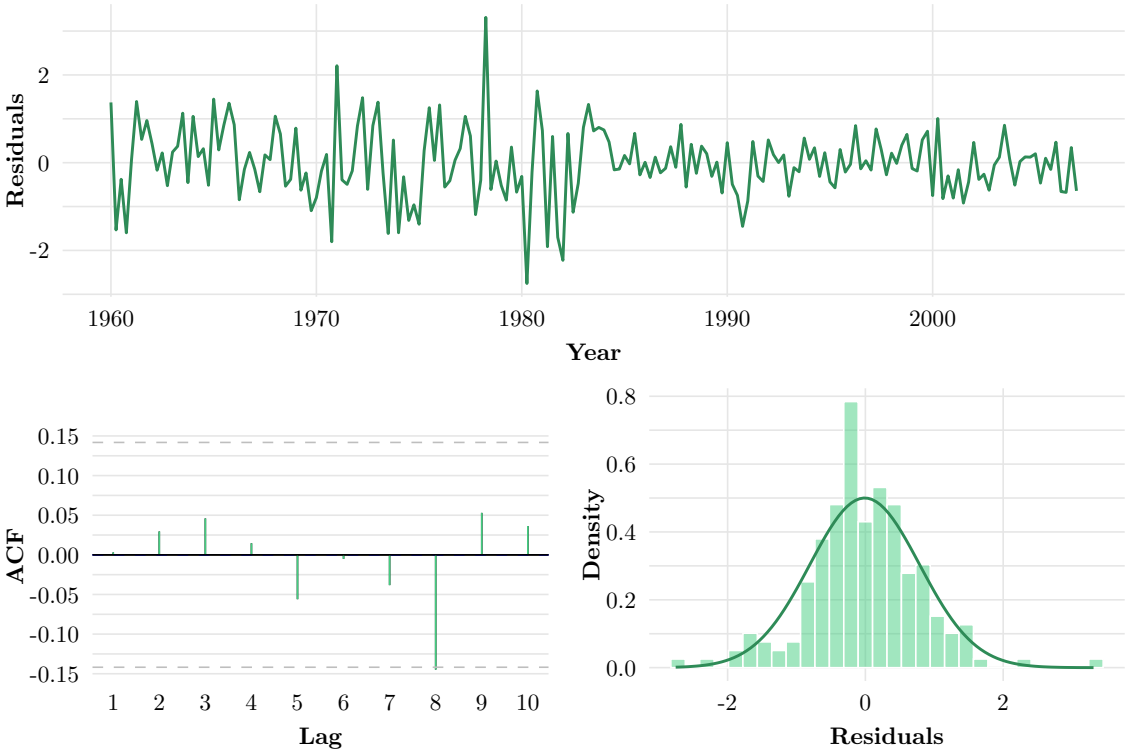
Figure 11: Leading Indicators and Recession Periods



Note: Figure shows leading indicator time series from 1959-Q1 to 2019-Q2 as well as periods of recessions as highlighted in gray according to the recession definition of NBER. Series tend to show a leading cyclicality with respect to business cycle downturns. Units of new housing constructions (HOUST), for instance, tend to reach its peak some quarters before a recession hits the economy.

C Residual Analysis ARIMA

Figure 12: Visual Inspection of Residuals



Note: Top figure shows time series of residuals resulting from an ARMA$(2,1)$ model. Visual inspection suggest white noise. Bottom left figure shows the corresponding empirical Autocorrelation Function up to lag 10 with 95% confidence bands. At lag 8, ACF shows a spike which is significant at the 5% level but insignificant at the 10% level. All other autocorrelations are highly insignificant both in size and statistically. This suggests that all autocorrelation is captured by the model and none is left to the residuals. Bottom right figure shows distribution of the residuals which suggests normality. The results strongly support the model specification.

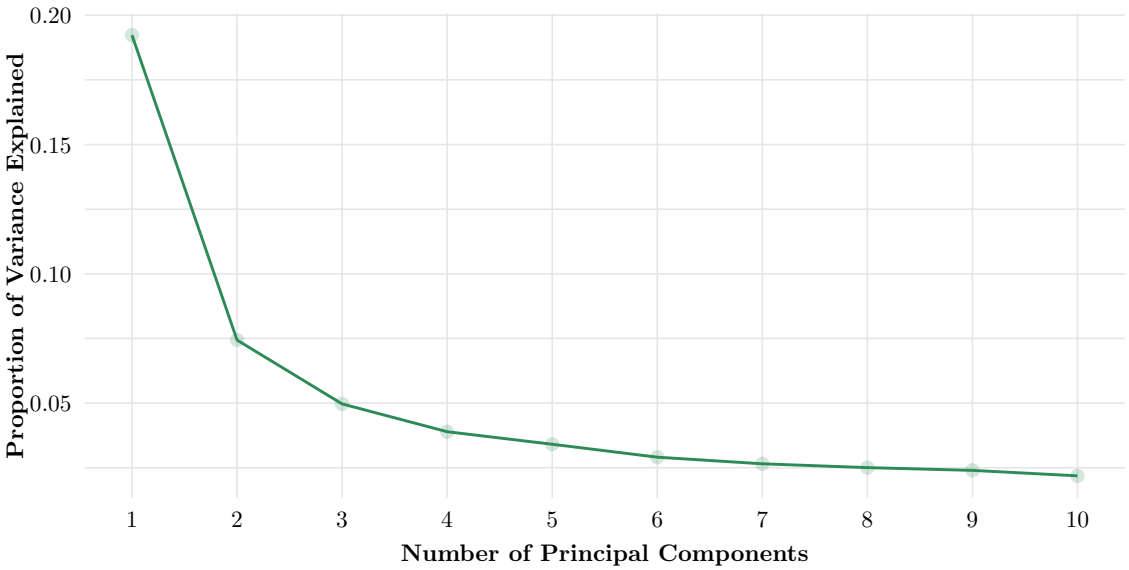
Table 10: Portmanteau Test Results

Test	lags	p-value
Ljung-Box	10	0.77
Box-Pierce	10	0.80

Note: Both Ljung-Box and Box-Pierce test do not allow to reject the H_0 that autocorrelation in residual series is not statistically different from zero up to lag 10. This result strongly supports the model specification.

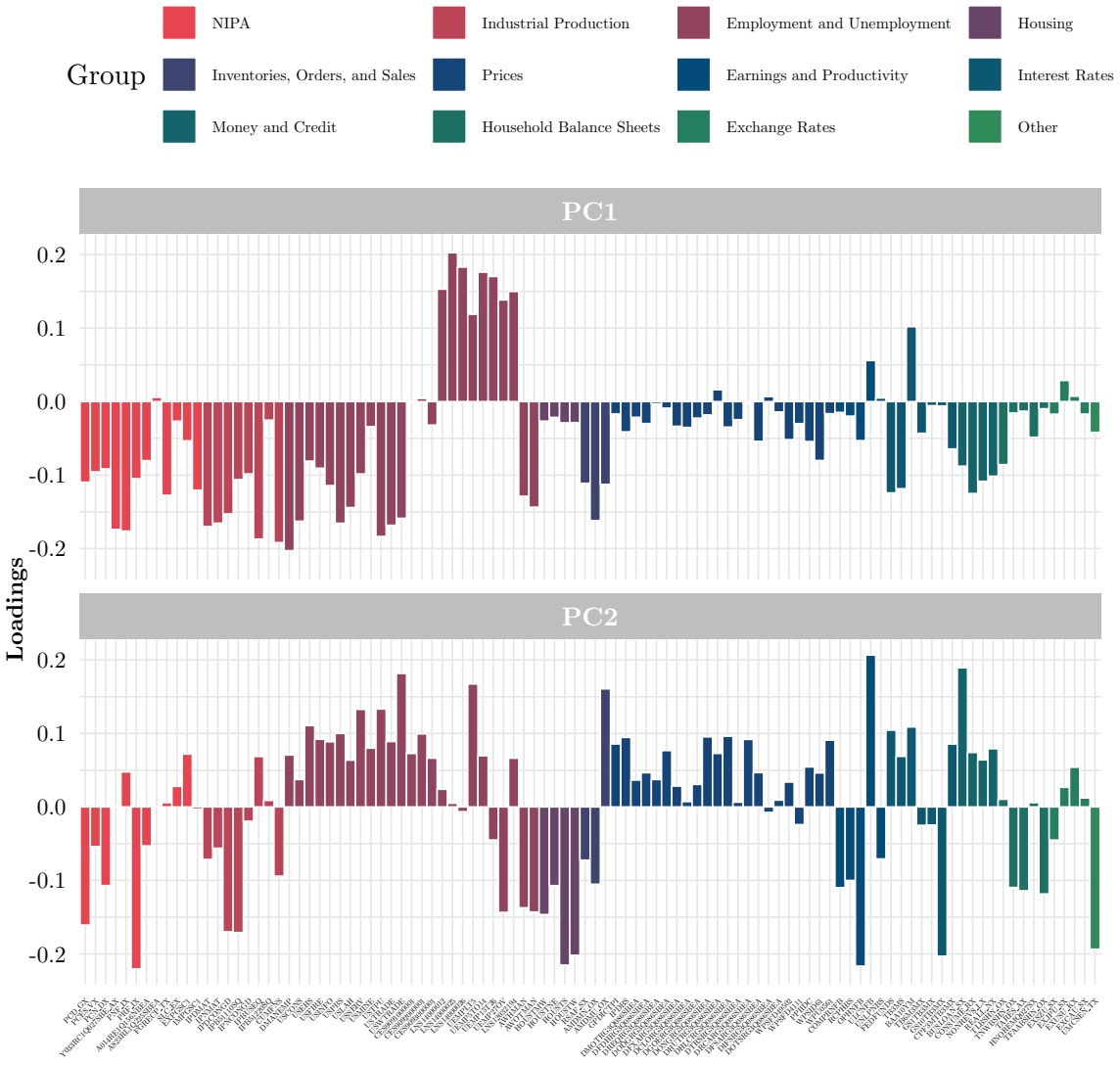
D Principal Component Analysis

Figure 13: Scree Plot



Note: Figure shows explained variance of the first 10 components of the S&W FAVAR model. According to cross-validation on training data, the optimal number of components that enter the final FAVAR model amounts to two. The first two components explain 26.7% of the overall variance in the reduced feature space based on the S&W FAVAR variables.

Figure 14: Loading Analysis

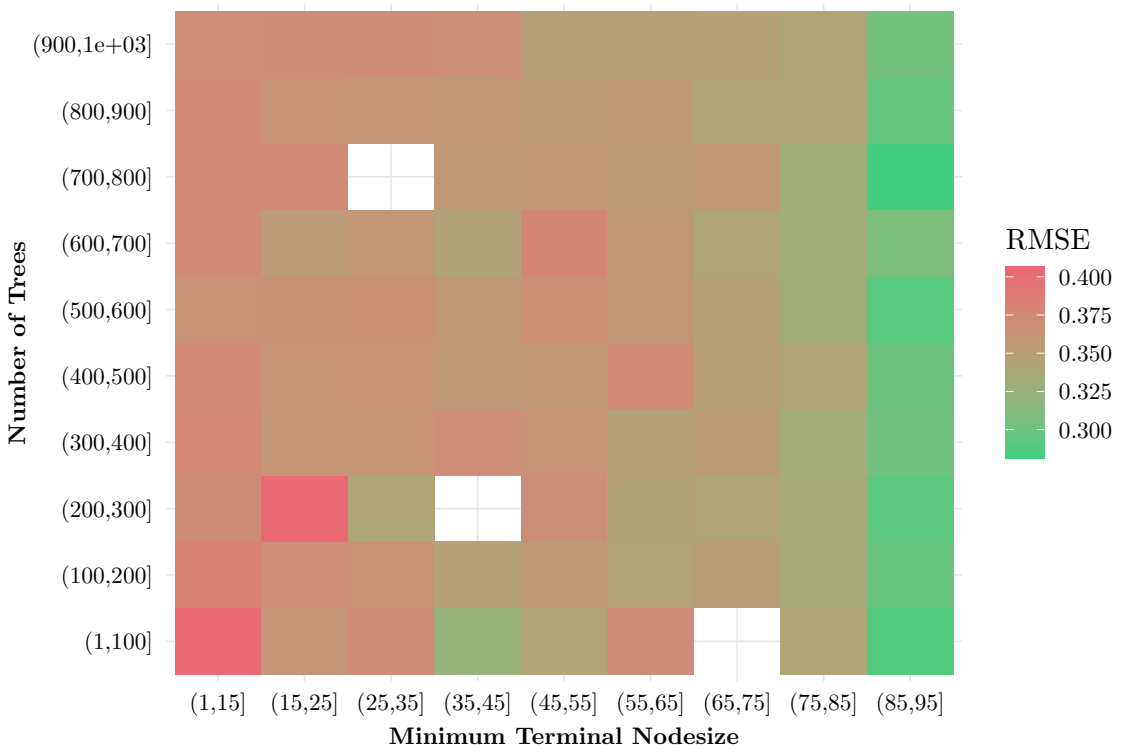


Note: Figure shows loadings of the first two components on the features used in the S&W FAVAR model.
It becomes obvious that the first principal component extracts mostly information from variables belonging to National Income and Product Accounts (NIPA), from industrial production variables, from features related to the labor market, from the group of inventories, orders and sales as well as from household balance sheet data. The first component strongly loads on variables from these groups in absolute value.
Loadings of the second principal component are less clear cut among the groups. However, it can be said that, in contrast to the first component, the second component captures information from housing prices, interest rates and consumer sentiment (variable UMCSEN_TX in group Other).

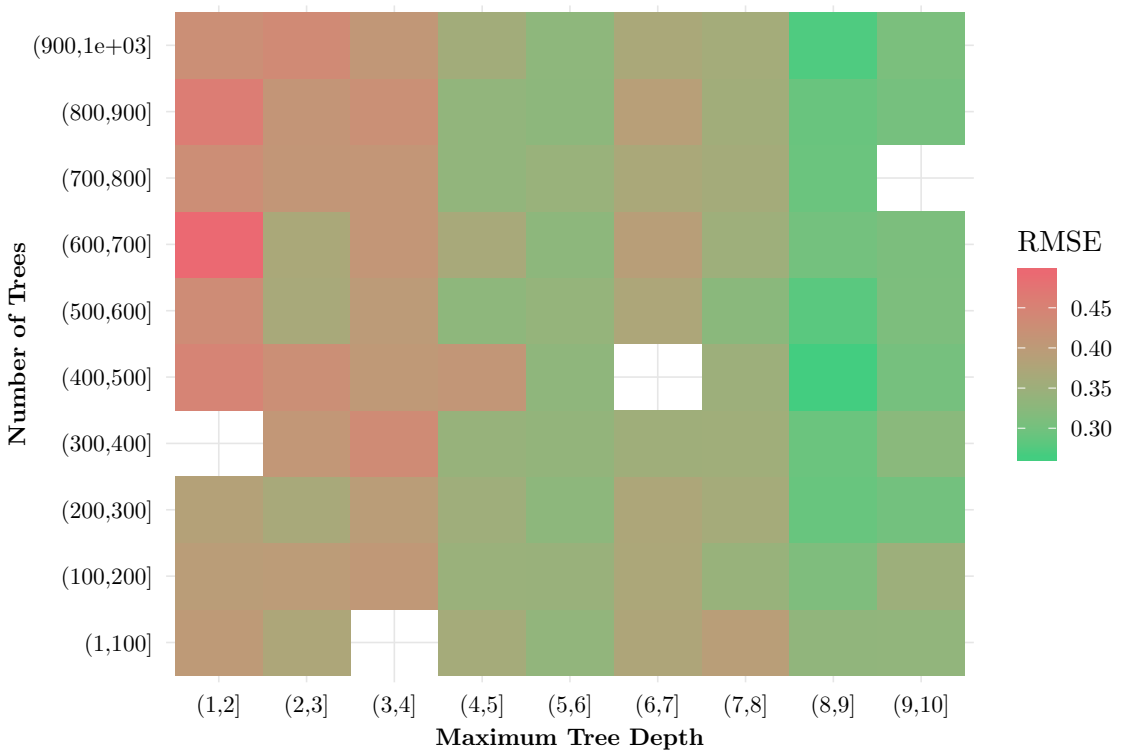
E Tuning Results Machine Learning Methods

Figure 15: Search Spaces and Optimal Parameter Constellations

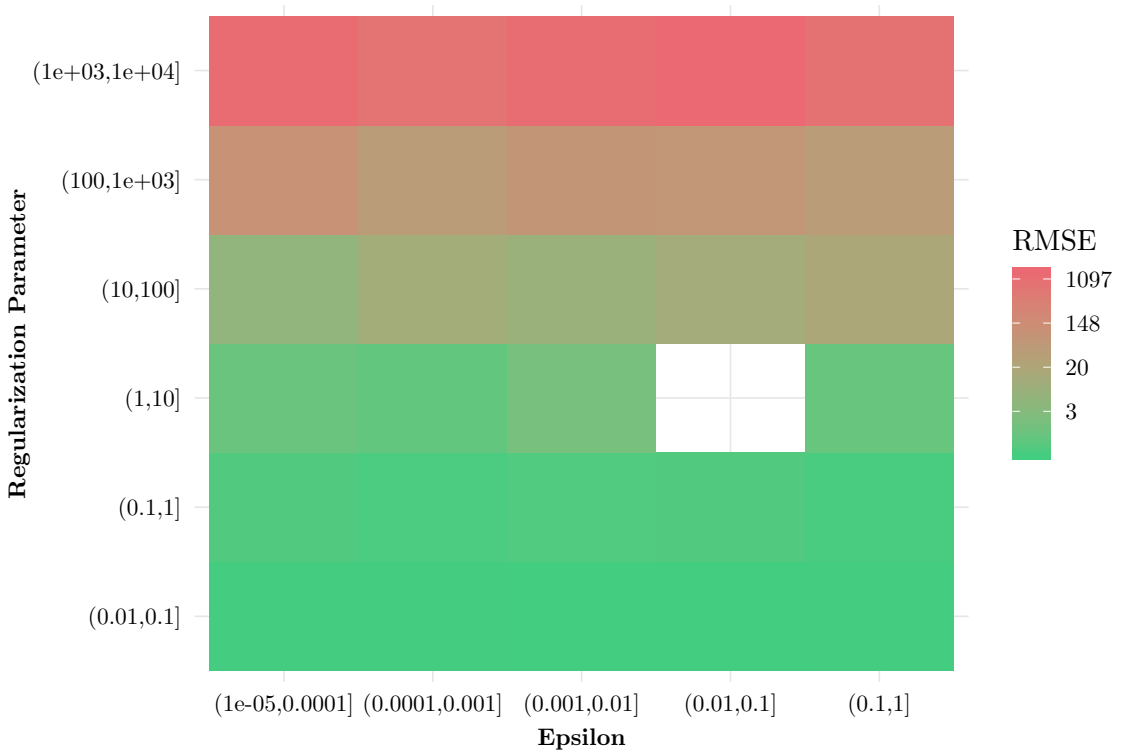
(A) Random Forest



(B) Gradient Boosting



(c) Support Vector Regression



Note: Figure shows search results of first-stage tuning for different parameter constellations as measured by RMSE on the training data. For this purpose, search spaces are separated into equally distanced (in case of SVR in log scale) raster and the training RMSE is displayed for each raster in form of a heatmap. An uncolored raster indicates that random search did not select the respective hyperparameter combination.
RMSE for RF clearly favors small trees with a large minimum terminal nodesize ($node_{min}$) of more than 85 observations. Given that the trees are small enough, the algorithm seems to be rather insensitive to the overall number of trees (M).
RMSE for GB rather favors deep trees with more than 8 interactions (hyperparameter $depth_{max}$). Given that the trees are deep enough, the algorithm seems to be rather insensitive to the overall number of trees (M).
Given a sigmoid kernel, RMSE for SVR clearly favors a regularization parameter (C) within the range of $(0.01,\ 0.1]$. Given that C is small enough, the algorithm seems to be rather insensitive to the radius of the epsilon tube (ε).

Declaration of Authorship

I hereby declare that the Masterthesis "The Nonlinearity of Crises: Machine Learning Approaches to Economic Forecasting" submitted to the Otto-Friedrich-University of Bamberg in partial fulfillment of a Master of Science degree in European Economic Studies (EES) is my own work. All sources used, either directly or indirectly, are acknowledged as references.

I am aware that the digital copy of my Masterthesis will be examined for the use of unauthorized aid and partial or complete plagiarism. I agree that my Masterthesis will be entered in a database in order to compare it with the existing literature. Further rights of reproduction and publication are not granted here.

The submitted paper was neither previously presented in another examination procedure nor has it been published so far.

Place, date

Signature

Eidesstattliche Erklärung

Ich erkläre hiermit gem. § 5 Abs. 3 PuStO, dass ich die vorstehende Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ort, Datum

Unterschrift