# Real-Time GDP Nowcasting in a Data-Rich Mixed-Frequency Environment: Harnessing Machine Learning

Charalampos (Haris) Karagiannakis*

May 9, 2025

**Abstract**

This article investigates whether machine-learning (ML) methods, combined with rich mixed-frequency datasets, can be useful in predicting current and future U.S. GDP growth in real-time, and further seeks to answer how ML-based models can be useful, by investigating the composition that constitutes the superior candidates. A set of pseudo real-time mixed-frequency vintages are used in an out-of-sample evaluation exercise to compare the performance of over 70 specifications across 13 ML model classes, alongside a comprehensive set of state-of-the-art econometric models and benchmarks. The results demonstrate that ML approaches can generate more accurate nowcasts and 1-quarter-ahead forecasts compared to benchmarks. In terms of the components (i.e., information sets; linear vs non-linear methods; structured vs component-wise incorporation of predictor dynamics) that make the successful combination, the findings suggest that there is a significant heterogeneity which depends on the machine learning method employed. Overall, models that (1) use linear ML methods or methods based on linear learners, (2) incorporate quarterly factors, and (3) exploit high-frequency predictors, frequently appear among the specifications that consistently rank in the upper quantile of the performance distribution. One of the ML methods that merits more attention is the gradient boosting algorithm with a linear base procedure. When L2 boost is estimated with the information set comprised solely of quarterly factors, it is consistently found among the top performers both overall, and for nowcasting in particularly. Moreover, among the linear ML specifications, bagged linear regressions are also found to consistently rank in the top quartile of model performance, almost regardless of the information set used and across error measures.

*JEL classification*: C53, C55

*Keywords*: Machine learning; Real-time nowcasting; Mixed-frequency data; Macroeconomic forecasting; High-dimensional time series

*haris.karagiannakis@kcl.ac.uk

# 1 Introduction

Many economic indicators describing various aspects of the economy, that are crucial for guiding decision-making, are published with delay and often undergo multiple rounds of revisions after their initial release. One notable example is the gross domestic product (GDP) which is a summary measure of macroeconomic conditions and serves as a key indicator closely monitored by policy institutions and private observers to inform public policy and business decisions. In many countries, GDP is available only on a quarterly basis and is subject to substantial delays. For instance, to illustrate the delay, in the United States, the Bureau of Economic Analysis (BEA) releases three estimates for each quarterly GDP figure, one around the end of each month following the close of the reference quarter. As a result, economic and policy decisions must be made without knowing the current state of the economy (i.e., the running quarter's GDP value), and at the same time the most recent known state is surrounded by uncertainty, as the GDP value for the previous quarter is typically expected to be revised. Tools that monitor economic activity in real time that can capture and accurately depict changes in rapidly evolving economic environments, are essential to enable the effective and prompt reaction of policymakers. Such tools are even more critical during periods of crisis, as they can allow the immediate response of monetary and fiscal authorities, as well as that of private decision-makers, when reaction time is paramount. A recent example that underlines the importance of developing robust monitoring tools is the COVID-19 pandemic, during which monetary and fiscal authorities needed to take immediate action to mitigate the economic fallout from the disruption caused by the lockdowns to the economies around the globe.

Despite the remarkable progress and innovations in the time-series econometrics literature, the COVID-19 pandemic uncovered vulnerabilities in existing methodologies, presenting significant challenges for policy institutions. For instance, in September 2021, the New York Fed suspended updates to its highly regarded real-time GDP nowcasting platform, citing that "the uncertainty around the pandemic and the consequent volatility in the data have posed a number of challenges to the model." Moreover, the exposed weaknesses were not limited to GDP forecasting methodologies but extended to the forecasting of other critical variables, such as inflation. In the fourth quarter of 2021, many central banks, international institutions and private sector forecasters published projections that severely underestimated headline inflation in their respective economies (see European Central Bank, 2022). The implications of these shortcomings were enormous for central banks, which relied on inflation forecasts to determine the timing for rising interest rates and stopping net bond purchases. These systematic weaknesses underscore the need to re-evaluate current forecasting prac-

tises and further suggest the importance of supplementing models with high-frequency data, potentially extending beyond traditional macroeconomic indicators.

## 1.1 Overview of the Machine Learning Literature

In recent years, machine learning (ML) methods have emerged as a promising alternative to conventional econometric approaches, owing to their ability to effectively handle high-dimensional datasets, even when predictors far exceed the number of observations, and to capture complex nonlinear relationships. Advancements in computational power and developments in ML theory have encouraged researchers to revisit the potential of ML methods as macroeconomic forecasting tools, with considerable success. In a seminal work assessing multiple machine learning methodologies alongside traditional econometric benchmarks, Medeiros et al. (2021), showed that ML models, particularly random forests, systematically improve US inflation forecasts in data-rich settings. Focusing on linear techniques, Kotchoni et al. (2019) showed that combining model averaging with various forms of regularization can produce strong forecasts for a range of key macroeconomic indicators. Furthermore, ML methods in conjunction with large datasets have also been used to predict monthly U.S. stock returns. Gu et al. (2020), using over 900 covariates demonstrated that machine learning forecasts, especially those from deep neural networks and tree-based ensemble methods, significantly outperform traditional benchmarks in an out-of-sample evaluation setting that contained several thousands of U.S. listed stocks. Adopting a novel perspective, shifting the focus away from identifying a single best-performing model, Goulet Coulombe et al. (2022) explore the underlying properties that drive the success of machine learning procedures. Through a comprehensive meta-analysis based on an extensive pseudo-out-of-sample forecasting horse race, the authors decompose ML models into their key characteristics (i.e., non-linearities, dimensionality reduction, hyperparameter optimization, loss function specification, and the richness of the information set), and examine how each of these features influence predictive performance. Their findings suggest that employing (1) a nonlinear function approximator, (2) principal components for dimensionality reduction, and (3) K-fold cross-validation (CV) or the standard BIC to tune hyperparameters, all contribute to improving the forecasting accuracy of macroeconomic targets. Employing a similar evaluation framework, Goulet Coulombe et al. (2021a) construct numerous information sets based on different transformed versions of a large set of predictors and evaluate their marginal contributions to forecast accuracy for several monthly macroeconomic targets. Their analysis considers multiple linear and nonlinear ML techniques combined with the following data transformations: levels, stationary-transformed data, principal components, as well as two

newly proposed methods that are used as means for compressing the information within lag polynomials, namely moving average factors (MAF), and moving average rotations of the input data (MARX).[1]. Their findings indicate that MAF, MARX and levels as well as extracting common factors are all associated with lower RMSEs, particularly when paired with the nonlinear nonparametric tree-ensemble algorithms.

Despite the recent surge in popularity, earlier studies had also explored the use of machine learning methods in macroeconomic forecasting. For example, Inoue and Kilian (2008) considered the application of bagging dynamic linear regressions in forecasting U.S. CPI inflation. Using a monthly dataset of 30 real economic activity and financial series, they compared bagging techniques with other ML methodologies, including LASSO, Bayesian regression with a Gaussian prior, and the standard ridge estimator, demonstrating that ML approaches can deliver substantial improvements in prediction accuracy over benchmarks. Bai and Ng (2009) proposed employing the linear least-squares boosting algorithm as a device for selecting the most relevant predictors for specific targets within the framework of factor-augmented autoregressions. They demonstrated that boosting PCA-based factors, derived from a monthly panel of 132 predictors, substantially improves forecasts of various macroeconomic targets compared to standard diffusion index (DI) forecasts à la Stock and Watson (2002). Groen and Kapetanios (2016) studied the theoretical properties of Partial Least Squares (PLS), Bayesian shrinkage regressions, and PCA-based factor regression models under a variety of different unobserved factor structures for the predictor variables, and subsequently compared their out-of-sample performances using a panel of 105 monthly U.S. macroeconomic series to forecast numerous key monthly economic indicators.

The majority of machine learning studies in the macroeconomic forecasting literature, including the aforementioned ones, have focused on predicting monthly real and nominal indicators, such as headline and core inflation, industrial production, (un)employment, and the Federal funds rate, while largely overlooking GDP. A possible explanation for this gap lies in the unique challenges associated with GDP nowcasting (and forecasting), which is inherently a mixed-frequency data problem. Effective evaluation of data-rich methodologies typically requires hundreds of predictors, as using smaller datasets could obscure their true potential, unfairly disadvantaging them compared to approaches that can accommodate

[1]MAF applies principal components to summarize the lags of each predictor, while MARX involves forming sets of simple moving averages for each predictor

fewer predictors.[2,3] At the same time, real-time vintages of (rich) mixed-frequency datasets containing large numbers of relevant predictors are often unavailable and difficult to compile. These challenges, combined with the complexity of integrating predictors sampled at varying frequencies into machine learning models, have likely contributed to the relative lack of attention given to GDP in the machine learning forecasting literature. Nevertheless, the popularity and success of machine learning (ML) algorithms in macroeconomic forecasting have spurred researchers to extend standard ML methodologies to the mixed-frequency domain, aiming to exploit the predictive content of the readily available timely (high-frequency) indicators. For instance, Babii et al. (2022) introduce penalized MIDAS regressions leveraging the sparse-group LASSO (sg-LASSO) framework, originally proposed by Simon et al. (2013). Their sg-LASSO-MIDAS adaptation effectively handles high-dimensional predictors with varying frequencies, making it particularly suitable for nowcasting applications within data-rich mixed-frequency settings that include large frequency mismatches. Similarly, Hepenstrick and Marcellino (2019) build on the Three-Pass Regression Filter (3PRF) of Kelly and Pruitt (2015) which is an extension of the PLS algorithm, rendering it applicable to large and irregular information sets with mixed-frequencies and ragged-edges. The appeal of three-pass regression filter lies in its ability to distinguish between the subset of latent factors that influence the target, referred to as target-relevant factors, and those that are irrelevant to the target, but may still drive a large set of predictors. Building on the success of Medeiros et al. (2021) in forecasting inflation with random forest methods, Clark et al. (2022) focus on modifying the classic random forest methodology to accommodate high-frequency predictors and other patterns of missing data. Their proposed methodology, generalizes the standard random forest by allowing a linear relationship between the target and the splitting variable at each node, retaining the standard RF model as a special case when the slope is constrained

[2]In fact, the recent proliferation (and success) of ML methods in economic applications is attributed to the availability of large datasets, as discussed in Goulet Coulombe et al. (2022). Notably, many empirical macroeconomic studies investigating big data methods, including several cited herein (e.g., Carriero et al., 2019a; Giannone et al., 2021; Goulet Coulombe et al., 2022; Kotchoni et al., 2019; Medeiros et al., 2021), have benefited from the availability of a large monthly macroeconomic dataset known as FRED-MD. Developed by McCracken and Ng (2016), the dataset was compiled and is regularly updated with the goal of providing easy access to a standardized comprehensive dataset in order to encourage research on data-rich methodologies. The success and widespread adoption of FRED-MD inspired McCracken and Ng (2020) to introduce a quarterly counterpart (FRED-QD), and has led researchers to develop similar monthly datasets for other countries, such as Canada (Fortin-Gagnon et al., 2022) and the UK (Goulet Coulombe et al., 2021b).

[3]It is noteworthy that Carriero et al. (2019b), in a comprehensive meta-analysis of GDP growth and inflation point and density forecasts across multiple advanced economies, find that models incorporating a large number of predictors do not outperform those estimated on medium-sized datasets with a dozen carefully selected predictors. However, their analysis restricts the use of high-dimensional models to standard and MIDAS factor models, forecast combinations of single-predictor MIDAS regressions, and large BVARs, thereby excluding machine learning methodologies beyond factor models and simple ensembles. Furthermore, their set of predictors is limited to variables sampled exclusively at a monthly frequency.

to zero. More recently, Ballarin et al. (2024) proposed a methodology based on a relatively novel family of machine learning models called reservoir computing (RC), specifically adapting the Echo State Network (ESN) architecture, to handle mixed-frequency time series data. ESNs are recurrent neural networks (RNNs) whose core advantage is that the underlying state equation features fixed, randomly sampled parameter matrices, that do not require estimation, unlike conventional RNNs which makes their training difficult. The authors evaluate the performance of alternative network architectures of their proposed Multi-Frequency Echo State Network (MFESN) model against standard benchmarks through a multistep out-of-sample exercise for U.S. GDP, utilizing two information sets containing a total of 9 and 33 predictors, respectively. Despite most of these studies focusing on GDP as their target variable, they often limit their analysis to a single ML method and confine their comparative evaluation to a small set of well-established workhorse models and standard benchmarks, such as (dynamic) factor models and various univariate approaches. Comprehensive studies that provide a systematic evaluation of data-rich methodologies for predicting GDP remain scarce.

## 1.2 Contributions and Key Takeaways

This study contributes to filling this gap by employing an extensive array of linear and nonlinear machine learning models to nowcast and forecast U.S. real GDP growth, alongside state-of-the-art econometric workhorse models, and simple benchmarks. The linear ML algorithms encompass techniques like Ridge regression, LASSO, Elastic Net, and their adaptive variants, as well as the Sparse-group-LASSO-MIDAS regularized regression, specifically designed for mixed-frequency data. On the nonlinear ML side, the study considers Support Vector Regression (SVR) and Long Short-Term Memory (LSTM) Recurrent Neural Networks. The analysis also includes several linear and nonlinear ensemble methods such as Complete Subset Regressions (CSR), Bagged Linear Regressions, Boosted Linear Regressions, Random Forests, Boosted Regression Trees, as well as a methodology proposed herein: linear mixed-frequency gradient boosting algorithm with a structured (block-wise) inclusion of predictor dynamics. Different univariate time-series models are used to establish a baseline for comparison. Additionally, given the study's focus on high-dimensional data settings, factor-augmented autoregressions (FAR) and large Bayesian vector autoregressions (BVARs) are also included to provide a comparison with established econometric methods developed particularly for leveraging extensive datasets.

The second contribution of this study lies in evaluating various methods for incorporating the information from a large set of predictors sampled at different frequencies into the

ML models considered. To achieve this, I construct and compare three information sets, based on the same set of 257 predictors, that differ only in the way they handle the uneven frequencies between the quarterly target variable and the predictors.[4] Specifically, I consider a simple temporal aggregation scheme where high-frequency predictors are converted to quarterly by averaging with uniform weighting, as well as two alternatives inspired by the MIDAS literature. The two methods correspond to alternative approaches proposed in the literature for parameterizing the weights of lag polynomials. The first method disaggregates high-frequency series into lagged terms and incorporates both high- and same-frequency covariates (and their lags) in an *unrestricted* manner, similar to that used in U-MIDAS and standard ADL regressions. The second method aggregates the lagged values of each covariate (ex-ante) using different sets of weights derived from Legendre polynomials of varying orders, producing several temporally aggregated versions of each predictor. While for tree-based nonparametric ML methods there is no estimation of weights involved, I use the underlying sets of covariates produced by each method as a feature engineering step to harmonize frequencies and generate a potentially relevant set of regressors for training these models. Consequently, the aim is, for nonparametric methods to select the individual high- and/or same-frequency lags or the sets of linear combinations of lags (in the latter case) that best predict future values of $y_t$. The idea of imposing reasonable restrictive assumptions on the pattern of weights in Autoregressive Distributed Lag (ADL) models, dates back to the early distributed lag literature (Almon, 1965). However, recently researchers have started exploring the advantages of using orthogonal polynomials, with the seminal work of Babii et al. (2022), who proposed using Legendre dictionaries to approximate the MIDAS weight function in the context of high-dimensional regularized MIDAS regressions. To the best of my knowledge, the present study represents the first attempt to combine Legendre aggregation with any machine learning method beyond the sparse-group LASSO regression considered in the aforementioned seminal work. This contribution extends recent efforts in the macroeconomic forecasting literature to identify the key determinants of forecasting performances of ML algorithms and other statistical models (e.g., Carriero et al., 2019b; Goulet Coulombe et al., 2022) and to assess the impact of alternative data transformations on predictive accuracy (e.g., Goulet Coulombe et al., 2021a).[5]

To evaluate the competing forecasting methodologies and information sets, I compile a

[4]Given that the time-series forecasting problem is inherently a dynamic problem, the three methods for handling mixed-frequency data also correspond to alternative methods for handling the underlying dynamics of the predictors sampled at the same or higher frequencies relative to the target.

[5]One could also consider constructing additional information sets using the MARX and MAF transformations proposed by Goulet Coulombe et al. (2021a) to aggregate the MIDAS lags of high-frequency predictors and evaluate their performance in predicting GDP growth alongside those proposed here. However, this is reserved for future work.

novel comprehensive mixed-frequency dataset for the U.S., by integrating the unique information from the FRED-QD and FRED-MD datasets. Earlier in this section, I emphasized the importance of evaluating model performance in a setting that reflects forecasters' access to extensive datasets, to uncover the potential of ML algorithms and enable a robust comparison against alternatives. However, large datasets are only one defining characteristic of today's forecasting environment. As noted above, policy institutions have increasingly shifted to monitoring economic activity in nearly real-time. Consequently, a proper evaluation of nowcasting and forecasting methodologies requires a well-designed experimental procedure which takes into account that economic decisions are made in real-time, and monitoring occurs at frequent intervals, or even, as soon as new data are released. To that end, to create an experimental setup that aligns with the evolving data landscape and how economic activity is monitored today, the following steps were implemented. First, I constructed a set of monthly pseudo real-time vintages that take into account the release schedule of economic data, replicating the ragged-edge the forecaster would encounter at the last day of each month. Each vintage consists of a pair of unbalanced panels containing 87 quarterly and 171 monthly macroeconomic and financial indicators. Second, I designed two pseudo-out-of-sample evaluation settings, spanning 18 years of out-of-sample observations, that differ with respect to the underlying frequency that GDP is assumed to be monitored. The first experiment assumes GDP is tracked once per quarter and uses quarter-end vintages (i.e., corresponding to the third month of each quarter) to conduct the forecast evaluation. The second experiment replicates monthly monitoring, where GDP nowcasts and forecasts are generated three times per quarter, at the end of each month. While the standard quarterly OOS enables the use of statistical tests to assess the robustness of the results, the monthly evaluation verifies whether the algorithms and information sets identified in the quarterly OOS are equally suitable for tracking GDP at higher frequencies, in line with current practices. Cimadomo et al. (2021) draw a parallel between the challenges of monitoring economic activity and the defining characteristics of Big Data, encapsulated by the *3 V's* (of Big Data): Volume, Velocity, and Variety.[6] In the context of nowcasting (and forecasting) macroeconomic data, these three dimensions can be roughly interpreted as follows: *Volume* reflects data richness; *Velocity* relates to the high frequency flow of data, which enables the real-time updates of predictions; and *Variety* encompasses diverse types of data from different sources, published asynchronously and at varying frequencies, leading to mixed-frequency datasets with ragged edges. By utilizing a comprehensive mixed-frequency dataset and a set of month-end vintages designed to reflect real-time data availability, this study evaluates the comparative performance of models within an experimental framework that incorporates all

[6]The concept of '3 V's of Big Data' was popularized by Berman (2013).

three dimensions.

Despite the ability of machine learning (ML) methods to handle high-dimensional data, recent studies have found that training ML models on a handful of factors that summarize a large set of predictors, or combining these common factors with the individual predictors, substantially improves the forecast accuracy of macroeconomic targets compared to applying ML directly to all individual series (e.g., Goulet Coulombe et al., 2021a, 2022).[7] This has motivated researchers to explore the effectiveness of combining various ML models together with factor-only information sets, derived from different factor extraction methods, often achieving considerable success. For example, Chinn et al. (2023) demonstrate the effectiveness of this approach in an empirical application for nowcasting world trade. On the other hand, the findings of Medeiros et al. (2021) from their ML horse race for forecasting inflation demonstrate that while the best performing models do not impose sparsity, assuming a factor representation to model future inflation is also inadequate, whether using standard linear factor-augmented autoregressive (FAR) models or boosting factors. This underscores the ongoing debate in the economic forecasting literature on the most effective way to represent the predictive relationships of macroeconomic targets, which often centers on choosing between *sparse* and *dense* modelling techniques.[8] In an effort to reconcile this debate, Giannone et al. (2021), instead of choosing between dense or sparse modelling techniques, proposed a Bayesian framework that allows for both, leaving the data to decide. Applying their methodology to the problem of forecasting U.S. industrial production growth and using the FRED-MD data, they found that the best-fitting predictive models included on average only 25% of the total number of predictors (approximately 32 variables). However, the subset of the selected predictors was different each time, rejecting a conclusion for a sparse representation. Their findings suggests that Bayesian and other types of model averaging techniques pooling models each incorporating several predictors, can potentially outperform purely sparse models. As an alternative frequentist solution, a new class of models known as *sparse plus dense*, has recently emerged, nesting both types of signals, thereby departing from the standard assumption that coefficients must be either sparse or dense (Chernozhukov et al., 2017; Fan et al., 2023). Furthermore, Beyhum and Striaukas (2023) extend sparse plus dense regression methods to handle mixed-frequency data by proposing two MIDAS adaptations. Specifically, they propose a factor-augmented

[7]Similarly, Bai and Ng (2009) find that applying least-squares boosting on principal components overall performs better compared to boosting the underlying set of observables.

[8]Ng (2013) classified predictive models designed to handle high-dimensional datasets into two broad categories based on their dimensionality reduction mechanisms: sparse and dense modelling techniques. *Sparse* techniques assume that only a small subset of the available predictors is relevant (e.g., LASSO), whereas *dense* models rely on the assumption that all variables are meaningful for the prediction (e.g., ridge regressions and factor models).

variant of the sg-LASSO-MIDAS regression, in which the coefficients of the common factors (i.e., dense component) do not enter the sg-LASSO penalty function. Their second adaptation extends the LAVA estimator of Chernozhukov et al. (2017) to the mixed-frequency domain by incorporating structured sparsity in the spirit of Babii et al. (2022), where the parameters of (MIDAS) lags associated with each (high-frequency) predictor are grouped together. Applying these methods to nowcast U.S. GDP growth, and assuming the dense component consists of monthly macroeconomic (PCA-based) factors, while weekly financial and monthly macroeconomic indicators make up the sparse component, they demonstrate that sparse plus dense MIDAS regression methods can effectively capture shocks during the COVID pandemic, improving GDP growth nowcasts compared to sparse-only and dense-only approaches.[9]

The final contribution of this study extends this line of research. Given the established importance of including common factors as regressors in predicting GDP growth, and building on the recent shift in the ML macroeconomic forecasting literature toward training ML algorithms with factor-only information sets, I investigate whether GDP growth is best predicted using only factors on the RHS or composite information sets that combine both factors and the individual series. To clarify, the primary contribution is not the introduction or experimentation with sparse plus dense techniques, but rather to provide guidance on the best ways to combine ML methods with information sets that incorporate common factors. As such, for sparsity-inducing algorithms, I do not impose a sparse plus dense structure, as the respective algorithm is allowed to perform variable selection on the included sets of factors. Using machine learning techniques that induce sparsity, together with information sets composed solely of factors, can be viewed as leveraging the variable-selection capabilities of these algorithms to identify the most relevant factors and their lags for predicting a specific target variable (for a similar treatment, see Bai and Ng, 2009).

The results of this study complement previous studies that have highlighted the benefits of utilizing ML techniques together with high-dimensional datasets to form macroeconomic predictions. Through a rare attempt to compare the performance of so many models, including both machine learning and standard state-of-the-art econometric models, this thesis demonstrates that machine learning methods consistently generate more accurate nowcasts and short-term forecasts of U.S. GDP growth rates compared to standard workhorse models. While the analysis evaluated model performance for nowcasts and for up to one year ahead forecasts, the results suggest that the ability of ML methodologies to predict GDP growth, primarily concerns shorter horizons, of up to 2 quarters ahead. This is largely in line with the

[9]Their recommended configuration permits sparse plus dense patterns on macroeconomic predictors, while imposing a purely sparse structure on financial indicators.

broader GDP forecasting literature suggesting that the gains of statistical and institutional GDP forecast over the constant growth model are substantial typically only for the current and previous quarters (e.g., Bańbura et al., 2013). The evidence from the comparison of the various information sets considered, suggests that the most effective approach for handling the heterogeneous frequencies is to temporally aggregate the high-frequency predictors to match the target variable's frequency. Furthermore, while material gains in nowcast accuracy can overall be realized when adding the individual predictors alongside the principal component factors that summarize these series, this outcome largely depends on the specific ML method being used, as well as the selected treatment for the inclusion of predictors sampled at different frequencies. Notably, reductions in nowcast errors are observed specifically when models are trained on the information sets that use mixed-frequency data, while adding the quarterly-aggregated predictors on the models trained solely on quarterly factors is found to deteriorate out-of-sample performance. The findings from the comparison of the different information sets suggests that in order to answer the question on how to best treat mixed-frequency data and whether one should consider incorporating individual predictors alongside estimated factors, we need to examine the top-performing models and their constituent components. Among the evaluated algorithms, the L2 boosting method with a linear base learner trained on the quarterly-factor information set consistently emerges as top performer, demonstrating robustness across horizons and error metrics. Overall, linear ML methodologies, such as factor-augmented ARs, Ridge Regressions, Bagging, CSR, LASSO, EN, and their adaptive variants, are found to dominate the upper quartile of the performance distribution as measured by the 5-horizon average RMSE, while nonlinear models, particularly random forests, gain prominence when MAE rankings are used. Among the linear specifications, bagged linear regressions are found to consistently rank in the top quartile of model performance, almost regardless of the information set used and across both error measures. Additionally, the diffusion-index approach of Stock and Watson (2002), estimated with target-factors following Bai and Ng (2008), is identified as a strong candidate due to its simplicity and reliable performance, though it does not rank as the absolute best in any scenario.[10] Finally, one notable distinction between monitoring GDP on a monthly basis instead of once at the end of every quarter, is that in the out-of-sample experiment that assumes the former setting, there is an increased representation of information sets con-

[10]It is noteworthy that, in this study, the standard linear diffusion index model and its targeted counterpart are based on factors extracted from several quarterly (and quarterly-aggregated monthly) series, that are available more timely than GDP, and therefore the estimated factors contain several leading observations. This configuration enables within-quarter updates of predictions generated from the FAR specification, making these benchmarks more challenging to outperform, compared to how they are usually depicted in studies comparing alternative approaches for GDP nowcasting with mixed-frequency data.

taining high-frequency panels in the upper quartile of both rankings, implying that using high-frequency predictors helps capture useful within-quarter signals early in the quarter.

### 1.3 Organization of the Article

Section 2 establishes the methodological framework, and presents the models. Section 3 details the methods for handling input series sampled at different frequencies, setting the stage for the alternative information sets used to train the machine learning algorithms. Section 4 gives an overview of the large mixed-frequency dataset, and explains the process for constructing the set of pseudo real-time vintages. Section 5 presents the forecast evaluation framework used to assess and compare the candidate models and information sets. Section 6 presents the findings, and Section 7 concludes.

## 2 Methodology

The goal is to predict $y_{t+h}$ over horizons $h = 1, \ldots, H$ using a large set of potential predictors represented by the $N$-dimensional vector $\mathbf{X}_t = (x_{1t}, \ldots, x_{Nt})'$. The general direct-forecasting framework is given by:

$$y_{t+h} = f_h(\mathbf{X}_t) + u_{t+h}, \quad h = 1, \ldots, H, \quad t = 1, \ldots, T, \tag{1}$$

where $f_h(\cdot)$ is an unknown function that maps the information spanned by the covariates to the future values of the target time series, and $u_{t+h}$ captures the error which is assumed to be zero-mean. $\mathbf{X_t}$, encompasses the typical set of covariates which includes a large set of economic indicators sampled at various frequencies, and possibly lagged values of the dependent variable (autoregressive terms), as well as common estimated factors.[11] The direct-forecasting framework implies that a distinct mapping is estimated for each forecasting horizon, meaning that $f(\cdot)$ varies with $h$. The purpose of the forecasting problem is to identify the method that provides the best estimate $\widehat{f_h}$ for the target function $f_h(\cdot)$, with the aim of minimizing a given measure of prediction accuracy.

Let $Y_t$ denote the economic aggregate of interest to this study, the real US GDP. The macroeconomic forecasting literature commonly assumes that real GDP is best described by

[11] Given the focus of this study on generating real-time predictions taking into consideration the delay in the release of the target variable and the availability of more timely indicators, it should be noted that the information available to the forecaster for predicting the target variable $h$ periods ahead will typically extend beyond $t$ and even beyond $t + h$. However, for simplicity, I maintain the standard notation, and use subscript $t$ to denote the conditioning information throughout this section. The notation is clarified in the sections that introduce the different information sets.

an $I(1)$ process. Consequently, to approximate stationarity in the target variable, the models presented in this section are set to forecast the continuously compounded (c.c.) quarter-on-quarter (QoQ) growth rate in quarter $t+h$, defined as

$$y_{t+h} = 100 \ln(Y_{t+h}/Y_{t+h-1}) \tag{2}$$

where $Y_t$ is observed for quarters $t = 1, \ldots, T$. Figure 1 displays the log-transformed QoQ growth rate of real US GDP over the period 1959Q2 to 2021Q1 (248 quarters). I next introduce the predictive models evaluated in this article. A comprehensive list of all models is provided in Table 1.

## 2.1 Standard Econometric Benchmarks

### 2.1.1 Autoregressive Model

This article employs several autoregressive (AR) benchmarks using the iterated forecasting formulation, with alternative methods for determining the lag order. The forecasts for the $P$-th order autoregressive model, AR($P$), are obtained by first estimating the parameters in the following one-period-ahead model using OLS:

$$y_{t+1} = \phi_0 + \sum_{p=1}^{P} \phi_p y_{t+1-p} + \varepsilon_t. \tag{3}$$

The $h$-step ahead forecast is then recursively computed using: $\hat{y}_{t+h|t} = \hat{\phi}_0 + \sum_{p=1}^{P} \hat{\phi}_p \hat{y}_{t+h-p|t}$.

### 2.1.2 Random Walk

The second benchmark in the univariate time-series family, is the constant-growth model, derived by constraining the autoregressive parameters in Equation 3 to be zero, i.e., setting $P = 0$. Specifically, restricting $\phi_p = 0$, yields the white noise model for GDP growth ($y_t$), which implies an underlying random walk (RW) with drift process for the (log) level of GDP. We refer to the constant-growth model as the RW model, and include it among the benchmarks.

### 2.1.3 Large Vector Autoregressions

Vector autoregressions (VARs) are among the workhorse models for both forecasting and policy analysis. Let $\mathbf{y}_t = (y_{1t}, \ldots, y_{Nt})'$ represent the vector of $N$ observables. The general VAR($p$) model can then be expressed as:

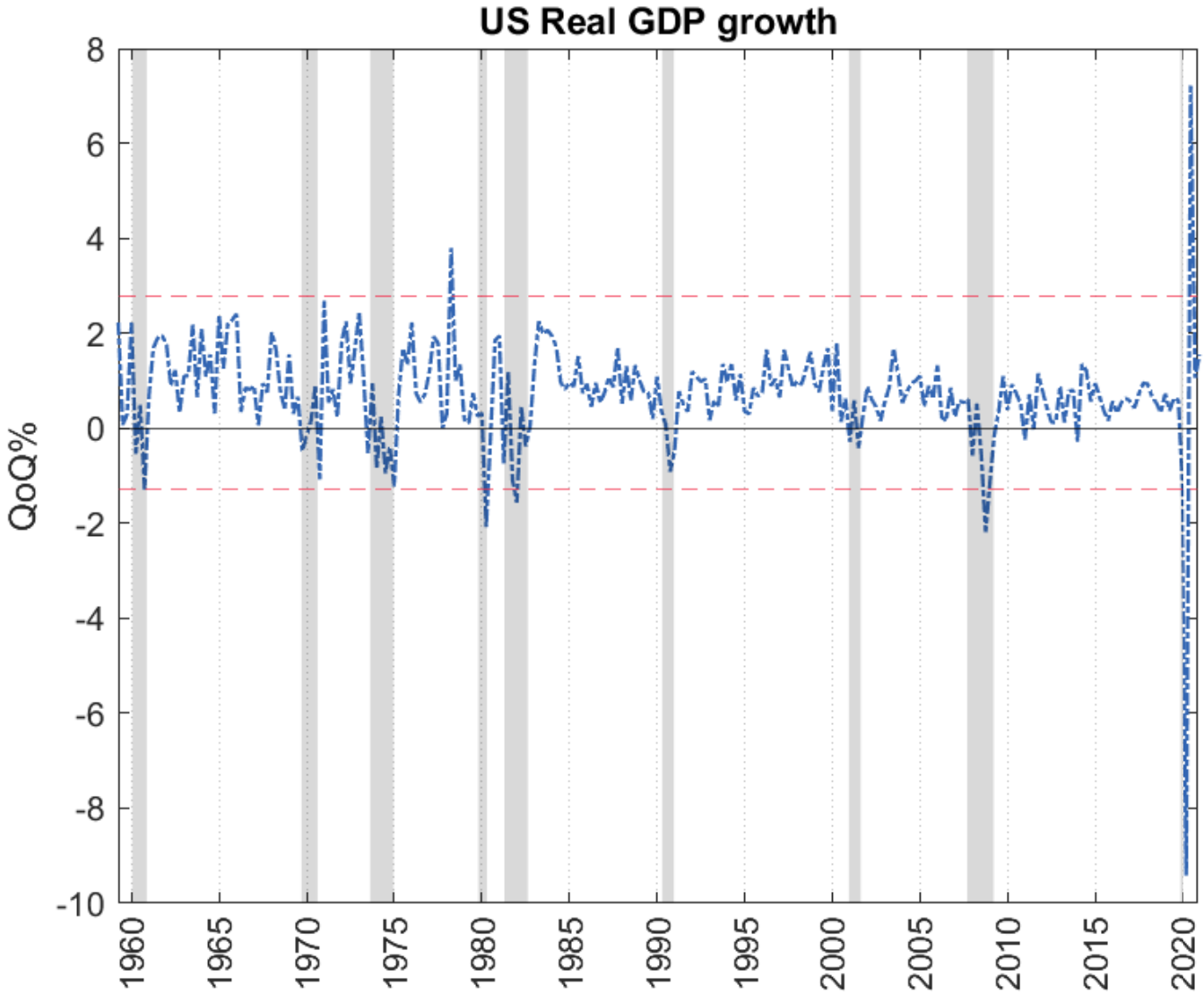


Figure 1: Real GDP growth rate, 1959Q2 to 2021Q1

The two horizontal dashed lines indicate thresholds corresponding to the median of the series plus and minus 2.5 times its interquartile range, highlighting unusually large deviations. Shaded areas denote NBER recession periods.

Table 1: List of time series forecasting models

| Acronym | Model Description | Reference |
|---|---|---|
| AR(P) | Autoregressive iterated-specification | |
| RW | Random walk | |
| ARDI(K) | Autoregressive diffusion indices with K factors. Optimal lag-order via BIC | Stock and Watson (2002) |
| T.ARDI(K) | ARDI with target-factors. Hard-threshold set to $\|t\text{-stat}\|>1.96$ | Bai and Ng (2008) |
| BVAR-Minn | Homoscedastic large Bayesian VAR | Bańbura et al. (2010) |
| BVAR-CSV | Large Bayesian VAR with heteroscedastic innovations | Carriero et al. (2016) |
| Ridge | Ridge regression with BIC for $\lambda$ | Hoerl and Kennard (1970) |
| LASSO | Least absolute shrinkage and selection operator with BIC for $\lambda$ | Tibshirani (1996) |
| AdaLASSO | Adaptive LASSO | Zou (2006) |
| EN | Elastic Net with $\alpha = 0.5$ | Zou and Hastie (2005) |
| AdaEN | Adaptive EN | |
| CSR | Complete Subset Regressions (20C4) with hard-thresholding preselection | Elliott et al. (2013) |
| Bag | Bagging linear regressions | Inoue and Kilian (2008) |
| BBoost | Boosting linear regressions, block-wise | Bai and Ng (2009) |
| CBoost | Boosting linear regressions, component-wise | Buehlmann (2006) |
| BTree | Boosting regression trees | Friedman (2001) |
| RF | Random forests | Breiman (2001) |
| SVR | Support vector machine regression with Gaussian Kernel function | Drucker et al. (1996) |
| LSTM | Long-short-term memory RNN with 3-hidden layers | Hochreiter and Schmidhuber (1997) |
| SgLASSO-MIDAS | Sparse-group LASSO-MIDAS with block-K-fold CV for $\lambda$ & $\gamma$ | Babii et al. (2022) |

$$\mathbf{y}_t = \mathbf{A}_0 + \mathbf{A}(L)\mathbf{y}_{t-1} + \mathbf{u}_t \tag{4}$$

where $\mathbf{A}_0$ is an $N \times 1$ vector of intercepts, $\mathbf{A}(L) = \sum_{i=1}^{p} \mathbf{A}_i L^{i-1}$ represents a $p$-th order lag polynomial of VAR coefficients with each $\mathbf{A}_i$ being an $N \times N$ coefficient matrix, and $\mathbf{u}_t$ denotes the residuals. This study considers two commonly adopted variants. The first is the standard VAR in which innovations are assumed to be homoscedastic, and the second considers a flexible covariance structure allowing for heteroscedastic errors. Formally, the homoscedastic VAR model is given by assuming errors to be independent and identically distributed (iid):

$$\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}).$$

To allow for heteroscedastic innovations, this study adopts the common stochastic volatility (CSV) model proposed by Carriero et al. (2016), where the covariance matrix is scaled by a time-varying common factor, $e^{h_t}$, such that:

$$\mathbf{u}_t \sim \mathcal{N}\big(\mathbf{0}, e^{h_t}\mathbf{\Sigma}\big),$$

where the log volatility follows a stationary AR(1) process, $h_t = \rho h_{t-1} + \varepsilon_t^h$, with $|\rho| < 1$ and $\varepsilon_t^h \sim \mathcal{N}(0, \sigma_h^2)$. The two systems are estimated using Bayesian methods on a dataset of 20 quarterly variables. A description of the data and the priors used in estimating the two large BVAR systems is provided in Section **??**.

## 2.2 Factor Models

### 2.2.1 Autoregressive Diffusion Index Model

Stock and Watson (2002) introduced a framework, known as the factor-augmented regressions (FAR), that utilizes common factors, estimated through principal components, to forecast a target variable. Specifically, the predictors are assumed to follow an underlying factor structure, represented by

$$X_t = \Lambda F_t + e_t \tag{5}$$

where $F_t$ denotes an $r \times 1$ vector of common factors, $\Lambda = (\lambda_1, \dots \lambda_N)'$ is the matrix of factor loadings, and $e_t$ the idiosyncratic component, which captures the part of $X_t$ not explained by the factors. The FAR forecasting equation takes the form

$$y_{t+h} = \gamma_h + \alpha_h(L)y_t + \beta_h(L)\tilde{f}_t + \varepsilon_{t+h} \tag{6}$$

where $\tilde{f}_t \subset \tilde{F}_t$ is an $k \times 1$ vector, and $\tilde{F}_t$ is the $r \times 1$ vector of principal component estimates of $F_t$ extracted from the set of the $N$ observed predictors. $\alpha_h(L)$ and $\beta_h(L)$ represent finite order lag polynomials with dimensions $p$ and $V_f$, respectively. This study estimates static factors using the standard PCA algorithm with the expectation-maximization (EM) modification proposed by Stock and Watson (2002) to address the unbalancedness of $X_t$.

### 2.2.2 Autoregressive Diffusion Index Model with Targeted Predictors

Bai and Ng (2008) proposed modifying the standard factor-augmented regression (FAR) framework to use factors tailored to forecast a specific target variable. They introduce a preselection procedure to identify the subset of predictors most relevant to the target variable, and then apply principal components analysis to extract common factors from the refined set containing the *targeted predictors*. They propose various hard- and soft-thresholding rules that filter predictors based on their predictive power for the target series. In this study, targeted-DI forecasts are generated using a hard-thresholding preselection method. The procedure for modifying the standard FAR model is as follows:

(i) For predictor $i = 1, \ldots, N$ in $X_t$, run an OLS regression of $y_{t+h}$ on $x_{i,t}$ along with a set of control variables $W_t$, typically a constant and lags of the dependent variable $y_t$. Conduct a two-sided test for the null hypothesis that the parameter associated with $x_{i,t}$ is zero, and let $t_i$ represent the resulting $t$-statistic.

(ii) The $N_\alpha^*$-dimensional vector of *targeted* predictors is given by: $X_t^* = \{x_i \in X_t \mid |t_i| > c_\alpha\}$, where $c_\alpha$ is the critical value at significance level $\alpha$.

(iii) Principal components analysis is then applied on the set of targeted predictors $X_t^*$, and $h$-period-ahead forecasts are generated from the standard FAR framework predictive regression (Equation 6)

## 2.3 Linear ML Methods

Penalized linear regressions are the natural choice within the family of linear models, in settings where the number of predictors exceeds the number of observations. For linear models, the target function in Equation 1 takes the form $f_h(\mathbf{X}_t) = \boldsymbol{\beta}_h' \mathbf{X}_t$, and the general framework for the shrinkage estimator is given by

$$\widehat{\boldsymbol{\beta}}_h = \arg\min_{\boldsymbol{\beta}_h} \left[ \sum_{t=1}^{T-h} (y_{t+h} - \alpha_h - \boldsymbol{\beta}_h' \mathbf{X}_t)^2 + \sum_{i=1}^{N} p(\beta_{h,i}) \right] \tag{7}$$

where $p(\beta_{h,i})$ denotes the penalty function that depends on the tuning parameter $\lambda \geq 0$ which determines the balance between model complexity and in-sample fit. I consider several

popular alternatives for the penalty function.

### 2.3.1 Ridge Regression

Ridge regression (RR) was proposed by Hoerl and Kennard (1970). The penalty is given by:

$$\sum_{i=1}^{N} p(\beta_{h,i}) := \lambda \|\beta\|_2^2 = \lambda \sum_{i=1}^{N} \beta_{h,i}^2. \tag{8}$$

Ridge regression shrinks the coefficients of less relevant variables towards zero but retains all predictors in the model, which means it does not perform variable selection.

### 2.3.2 Least Absolute Shrinkage and Selection Operator

The LASSO regression was introduced by Tibshirani (1996) who added the $\ell_1$ penalty to the loss function of a linear regression model. The penalty is given by:

$$\sum_{i=1}^{N} p(\beta_{h,i}) := \lambda \|\beta\|_1 = \lambda \sum_{i=1}^{N} |\beta_{h,i}|. \tag{9}$$

The $\ell_1$ penalty shrinks the coefficients, and unlike RR, it can set the coefficients of less relevant predictors exactly to zero, effectively performing variable selection and producing sparse models.

### 2.3.3 Adaptive LASSO

The adaptive LASSO (adaLASSO), proposed by Zou (2006), uses a weighted version of the $\ell_1$ penalty term based on initial estimates of the coefficients obtained from a first-step regression. The penalty is defined as:

$$\sum_{i=1}^{N} p(\beta_{h,i}) := \lambda \sum_{i=1}^{N} \omega_i |\beta_{h,i}| \tag{10}$$

where $\omega_i = \left|\beta_{h,i}^*\right|^{-1}$ are the adaptive weights, and $\beta_{h,i}^*$ is the coefficient from the first-step estimation.

### 2.3.4 Elastic Net

Zou and Hastie (2005) propose the elastic net (EN) which combines the $\ell_1$ (LASSO) and $\ell_2$ (Ridge) penalties to address their individual limitations. The penalty is given by:

$$\sum_{i=1}^{N} p(\beta_{h,i}) := \lambda\alpha \sum_{i=1}^{N} \beta_{h,i}^2 + \lambda(1-\alpha) \sum_{i=1}^{N} |\beta_{h,i}| \tag{11}$$

where $\alpha \in [0, 1]$ is a tuning parameter that determines the relative contributions of the two penalties, with $\alpha = 0$ giving the LASSO solution, while $\alpha = 1$ providing the Ridge estimator.

### 2.3.5 Adaptive EN

Finally, from the family of shrinkage estimators, I also consider the adaptive modification of the EN model (adaEN) proposed by Zou and Zhang (2009). It is defined similarly to the adaLASSO, and it's estimation involves a two-step procedure with the weights defined by a first-round estimation of the EN model.

## 2.4 Ensemble Methods

### 2.4.1 Complete Subset Regressions

In settings where the number of potential predictors is large, an exhaustive forecast combination approach, which involves forming linear regressions with all possible variable combinations, becomes practically infeasible, as the number of models to be pooled together grows prohibitively large. To limit the number of models to be pooled and render the problem computationally feasible, Elliott et al. (2013, 2015) propose generating forecast combinations from all possible linear regression models formed by selecting a fixed number of $k$ variables from the total set of $N$ predictors. They refer to this collection of $k$-variate models as a *complete subset* and advocate using equal-weighted combinations. The predictive regression for the $m$-th model in the complete subset is given by:

$$y_{m,t+h} = \gamma_m + \alpha_m' W_t + (X_t' S_m)\beta_m + \epsilon_{m,t+h} \tag{12}$$

where $S_m$ is a $N \times N$ diagonal selector matrix with $k$ of its diagonal entries set to 1, while all other diagonal elements set to 0. The $k$ unity elements indicate which variables are included in the $m$-th $k$-variate model. With a dataset containing $N$ variables, the number of possible models generated from combining $k$ regressors is given by $c_{N,k} = \frac{N!}{k!(N-k)!}$. Let $\mathcal{M} = \left[m_1, m_2, \ldots., m_{c_{N,k}}\right]$ denote the model space of all the possible $k$-variate combinations. The CSR forecast is then the equal-weighted average:

$$\hat{y}_{t+h|t} = \frac{1}{M} \sum_{m \in \mathcal{M}} \hat{y}_{m,t+h|t} \tag{13}$$

where $M = c_{N,k}$. In situations with many predictors, even when $k$ is small, the number of different $k$-variate models in the complete subset can still be significantly large.[12] To make computing CSR forecasts feasible, in this study, I adopt a strategy similar to that of Kotchoni et al. (2019) and Medeiros et al. (2021) who limit the number of combinations by introducing a preselection step à la Bai and Ng (2008). Specifically, a variant of the hard-thresholding algorithm outlined in Section 2.2.2 is employed, where only the $N^* < N$ variables with the highest absolute $t$-statistics are retained, and predictive regressions are formed considering combinations of $k$ variables from the updated set containing the $N*$ targeted predictors.

### 2.4.2 Bagging Linear Regressions

Bagging or bootstrap aggregation, proposed by Breiman (1996), is an ensemble technique designed to reduce the out-of-sample prediction error by pooling together predictions from multiple unstable models.[13] The bagging predictor is obtained by generating a large number of bootstrap resamples of the original data, applying a pretest model selection rule to each of these resamples, and subsequently averaging the predictions from the models selected by the pretest on each bootstrap sample. Inoue and Kilian (2008) popularized its application to time-series models by examining its effectiveness in predicting U.S. CPI inflation. The modified bagging algorithm for high-dimensional time-series proceeds as follows:

(i) For each bootstrap sample $b = 1, \ldots, B$:

   (a) Run a pre-selection step by conducting two-sided tests on each slope parameter from an OLS regression fitted on the $b$-th sample using all $N$ potential predictors. Identify the subset of predictors that are statistically significant at a specified level $\tilde{\alpha}$: $X_t^{*(b)} = \{x_i \in X_t \mid \left|t_i^{(b)}\right| > c_{\tilde{\alpha}}\}$.

   (b) Run an OLS regression on the $b$-th bootstrap replica containing only the $N^{(b)}$ significant variables from the previous step, and calculate the $h$-step-ahead forecast, on the *original data*: $\hat{y}_{t+h|t}^{*(b)} = \hat{\gamma}^{*(b)\prime} + \hat{\alpha}^{*(b)\prime} W_t + \hat{\beta}^{*(b)\prime} X_t^{*(b)}$.

(ii) The bagged forecast is the average of all forecasts across the $B$ bootstrap samples:

$$\hat{y}_{t+h|t} = \frac{1}{B} \sum_{b=1}^{B} \hat{y}_{t+h|t}^{*(b)}. \tag{14}$$

As outlined above, the bagging algorithm requires that $t$-statistics are obtained from a regression that jointly considers the full set of potential predictors. In data-rich environments,

[12] For instance, with $N = 257$, even for $k = 4$ there are 177.556.160 different models.

[13] As Breiman noted in his seminal paper (Breiman, 1996), 'if perturbing the training set leads to significant changes in the predictions, then bagging can improve accuracy.'

this is likely to be infeasible, as $T << N$. To address this, Medeiros et al. (2021) suggest modifying the screening step by randomly dividing all variables into equal-sized groups and gathering the $t$-statistics obtained from estimating a regression for each group. Nevertheless, when the target variable is a low-frequency macroeconomic aggregate, the OLS infeasibility issue may persist, as observed in our case. To address the persistent dimensionality problem, I introduce an additional randomized variable-selection step following the group-based pretesting stage, by uniformly drawing a subset of $N_s^{(b)}$ variables without replacement from the set of $N^{(b)}$ predictors identified by the group-based preselection procedure.[14]

### 2.4.3 Boosting Linear Regressions

Boosting, originally introduced by Schapire (1990), and later Friedman (2001) formalized it as a functional gradient descent algorithm, is an ensemble technique for approximating an unknown nonlinear function $\Phi(.)$ by sequentially estimating multiple weak learners.[15] Assuming that the *quadratic-error loss function* is used to penalize deviations of $\Phi(X_t)$ from $y_t$, the boosting algorithm solves the problem:

$$\hat{\Phi} = \arg\min_{\Phi} \frac{1}{2T} \sum_{t=1}^{T} (y_t - \Phi(X_t))^2. \tag{15}$$

Under quadratic loss, the algorithm approximates $\Phi(x) = \mathbb{E}(y_t|X_t = x)$, and the boosting algorithm reduces to an iterative least-squares refitting of the residuals (Friedman, 2001).[16]

The iterative procedure for the generic $L_2$ boosting algorithm, under certain base procedures that ensure the additiveness of the model, can be described as follows. At each iteration, $m = 1, ..., M$, the algorithm fits a new learner to the *current residuals*, defined as the difference between the observed response and the aggregated function estimates from all previously trained learners. Formally, the residuals at iteration $m$ are given by

[14] This additional step mirrors Breiman's double randomization technique in random forests, where each tree is grown by randomly selecting a subset of predictors at every split node. This feature of the random forests is essential in reducing the correlation among base learners (individual trees), leading to their superior out-of-sample performance relative to a simple ensemble of bagged trees. In a similar spirit, the introduced randomization step not only ensures the feasibility of the estimates, but also facilitates the pooling of signals from base learners that carry distinct information, potentially enhancing the forecast accuracy of the aggregated signal.

[15] *Weak learner* is a particular type of *base learner*. Although the two terms are often used interchangeably, they carry distinct meanings. Base learner is a general term that describes the model that forms the building block of the ensemble, regardless of its individual strength. In the context of boosting ensembles, the building blocks are typically referred to as weak learners, since they are deliberately designed to have weak predictive power.

[16] The squared-error loss functions, was formalized in the context of boosting by Bühlmann and Yu (2003), who also introduced the term $L_2$-Boost to describe the boosting algorithm minimizing this loss. Gradient boosting algorithms that minimize the quadratic error loss function are also commonly referred to as least-squares boosting (LS Boost).

$u_{t,m} = y_t - \nu \sum_{j=1}^{m-1} \hat{\phi}_j(X_t)$, where $\hat{\phi}_j(X_t)$ denotes the prediction from the weak learner at the $j$-th iteration. The newly fitted learner from the $m$-th round, $\hat{\phi}_m$, is shrunk by a factor $\nu \in (0, 1]$ and added to the overall fit, arriving after $M$ iterations to the final estimate $\hat{\Phi}_M(.) = \hat{\Phi}_0(.) + \nu \sum_{m=1}^{M} \hat{\phi}_m(.)$. This final step highlights that boosting operates as an ensemble technique, where the resulting estimate $\hat{\Phi}_M$ is the sum of $M$ weak learners $\hat{\phi}_m$, each fitted to the re-weighted versions of the data, also known as *pseudo residuals*.

A key to avoiding overfitting in the boosting framework is to ensure that the learner remains weak, meaning it exhibits high bias and low variance. To that end, a particularly effective approach for carrying out boosting in high-dimensional problems, is to introduce learners that select only one variable at each iteration. This strategy, known as *componentwise* boosting, was introduced by Bühlmann and Yu (2003). Following Bai and Ng (2009), this study employs two methods for incorporating the lags of different covariates. The first approach follows naturally from the idea of componentwise boosting, and treats each variable and its lags as distinct predictors, while the second approach modifies the base learner, to treat lags of the same variable as a group, allowing for a structured inclusion of predictor dynamics. Below, I formally introduce the $L_2$ boosting procedure for each of the two alternative lag treatments, referred to as *component-wise* and *block-wise* boosting algorithms:

**Component-wise Boosting**

(i) Let $\hat{\Phi}_{t,0} = \bar{y}$ for each $t$, with $\bar{y} = \frac{1}{t} \sum_{s=1}^{t} y_s$

(ii) For iteration $m = 1, \ldots, M$:

a) Calculate the *current residuals* $\hat{u}_t = y_t - \hat{\Phi}_{t,m-1}$

b) For each variable $i = 1, \ldots, N$ regress the *current residuals* $\hat{u}$ on the $i$-th regressor to obtain $\hat{b}_i$. Compute $\hat{e}_{t,i} = \hat{u}_t - x_{t,i}\hat{b}_i$ and the corresponding $SSR_i = \hat{e}_i'\hat{e}_i$.

c) Let $i_m^*$ denote the index of the predictor selected at the $m$-th iteration, corresponding to that delivering the smallest $SSR$:

$$SSR_{i_m^*} = \min_{i \in [1,\ldots,N]} SSR_i = \min_{i=1,\ldots,N} \sum_{s=1}^{t} \Big( \hat{u}_s - \hat{\phi}_m(x_{s,i}) \Big)^2.$$

d) Let $\hat{\phi}_{t,m} = x_{t,i_m^*} \hat{b}_{i_m^*}$.

e) Update $\hat{\Phi}_{t,m} = \hat{\Phi}_{t,m-1} + \nu \hat{\phi}_{t,m}$ where $0 < \nu \leq 1$ is the step length.

**Block-wise Boosting**

(i) Let $\hat{\Phi}_{t,0} = \bar{y}$ for each $t$.

(ii) For $m = 1, \ldots, M$:

a) Calculate the *current residuals* $\hat{u}_t = y_t - \hat{\Phi}_{t,m-1}$

b) For each variable $i = 1, \ldots, N$ estimate the model:

$$\hat{u}_t = \sum_{p=1}^{p_i^*} \alpha_p y_{t-p} + \sum_{q=1}^{q_i^*} \beta_q x_{t-(q-1),i} + v_t$$

where lag orders $(p_i^*, q_i^*)$ for the $i$-th regressor are selected via BIC.

c) Let $(p_i^*, q_i^*) = \arg\min_{p,\,q} \mathrm{BIC}(p, q)$, and $\hat{b}_i$ the OLS estimator obtained by regressing $\hat{u}$ on $z_{t,i}$ where $z_{t,i} = (y_{t-1}, \ldots, y_{t-p_i^*}, x_{t,i}, \ldots, x_{t-(q_i^*-1),i})'$.
Compute $\hat{e}_{t,i} = \hat{u}_t - z_{t,i}'\hat{b}_i$ and the corresponding $SSR_i = \hat{e}_i'\hat{e}_i$.

d) Let $i_m^*$ be such that $SSR_{i_m^*} = \min_{i \in [1,\ldots,N]} SSR_i$.

e) Let $\hat{\phi}_{t,m} = z_{t,i_m^*}\hat{b}_{i_m^*}$.

f) Update $\hat{\Phi}_{t,m} = \hat{\Phi}_{t,m-1} + \nu\hat{\phi}_{t,m}$.

To tune the two hyperparameters of the model, namely the learning rate $\nu$, and the number of iterations $M$, Bühlmann and Yu (2003) recommend using a small value of $\nu$, leaving only the need for a stopping rule in order to determine $M$. In this study, I follow Bai and Ng (2009) and use information criteria to determine the stopping rule.

## 2.5 Nonlinear ML Methods

### 2.5.1 Random Forests

One can create tree ensembles using bagging applied to a set of trees by generating many bootstrap replicas of the original data and growing individual trees to each replica. A prominent example is the random forests model, proposed by Breiman (2001), which is a special case of a bagged trees ensemble, that incorporates an extra layer of randomness in the splitting stage, which helps control overfitting and strengthen prediction accuracy. Before delving into the specifics of random forests, it is instructive to examine its foundational building block, the regression tree. A regression tree is a nonparametric model that estimates the relationship between a target variable and its predictors by recursively partitioning the covariate space into a series of regions, allowing it to capture complex and potentially nonlinear relationships (Breiman et al., 1984). A tree is grown by recursively performing a binary split in each branch node using one of the variables, creating two (left and right) child nodes at each time. At each node, a set of rules determines the best variable to split, and at what value to perform the split (see figure 2). The random forests ensemble averages predictions from multiple regression trees. The algorithm is implemented as follows:

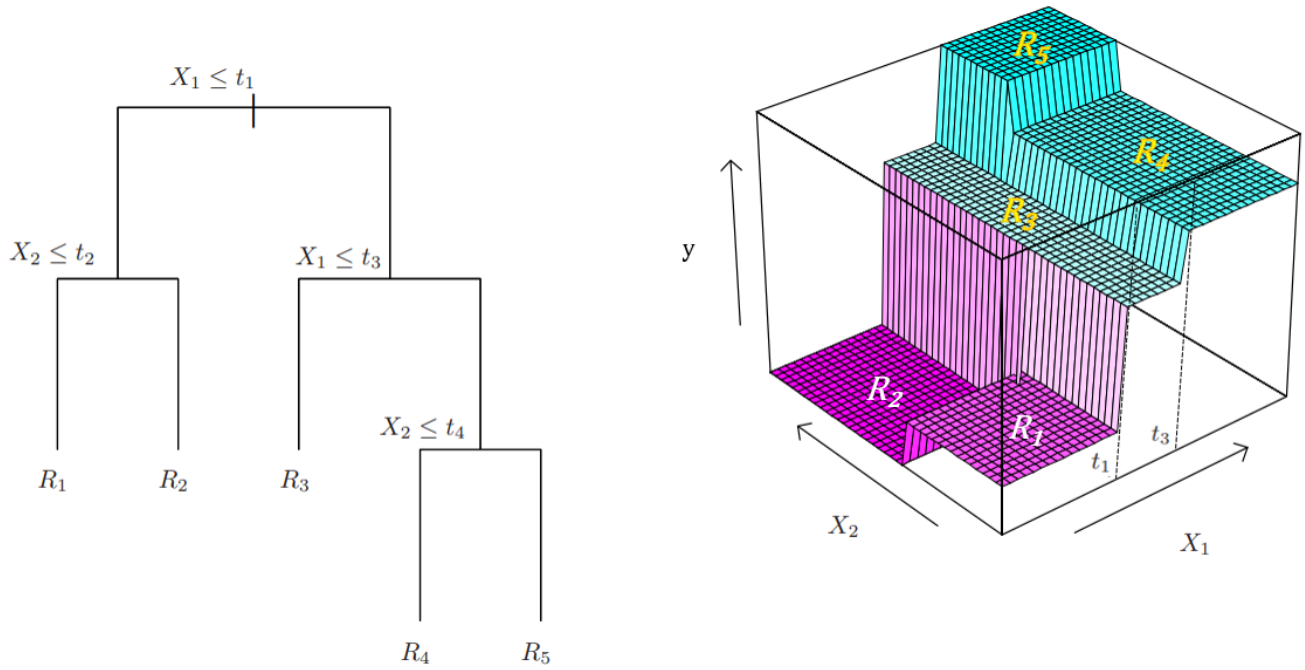


Figure 2: Illustration of regression tree with two variables (left) and the corresponding regression surface (right). Adapted from Efron and Hastie (2021), Figure 8.6.

(i) Generate $B$ bootstrap versions of the original dataset $(y_{t+h}, X_t')$.

(ii) Specify the minimum leaf size which will determine the number of regions, $K_b$. For each bootstrap resample $b = 1, \ldots, B$, grow a tree by sampling a random subset $N^*$ of the $N$ predictors prior to each split.

(iii) The random forests forecast is obtained by averaging the predictions from the $B$ regression trees, calculated using the original data:

$$\widehat{y}_{t+h} = \frac{1}{B} \sum_{b=1}^{B} \left[ \sum_{k=1}^{K_b} \widehat{\beta}_{k,b} \mathrm{I}_{k,b}\left(X_t; \widehat{\theta}_{k,b}\right) \right].$$

where $\mathrm{I}_k(X_t; \theta_k)$ denotes an indicator function determining membership in each region (corresponding to a single terminal node), such as:

$$\mathrm{I}_k(X_t; \theta_k) = \begin{cases} 1 & \text{if } X_t \in R_k(\theta_k) \\ 0 & \text{otherwise,} \end{cases}$$

with $\theta_k$ the set of parameters of the $b$-th tree that define the $k$-th region (i.e. the optimal variable and splitting point in each parent node within the path to terminal node $k$).

### 2.5.2 Boosting Regression Trees

In addition to the previously introduced boosting framework with linear regressions, the gradient boosting algorithm can also be implemented using trees as base learners. The least-squares gradient boosting algorithm modified to use regression trees, proceeds as follows:

(i) Set the number of steps $M$ and the shrinkage factor $\nu \in (0, 1]$, and initialize $\hat{\Phi}_{t,0} = \bar{y}$.

(ii) For $m = 1, ..., M$ repeat:

a) Compute the *current residuals* $\hat{u}_t = y_t - \hat{\Phi}_{t,m-1}$

b) Fit a shallow regression tree to the data $(\hat{u}_s, X_s, )_{s=1}^{t}$, and obtain the estimate for $\phi_{t,m}$

c) Update the fitted model by adding the shrunken version of $\hat{\phi}_{t,m}$: $\hat{\Phi}_{t,m} = \hat{\Phi}_{t,m-1} + \nu\hat{\phi}_{t,m}$.

(iii) The final fitted value is given by $\hat{y}_{t+h} = \bar{y} + \nu \sum_{m=1}^{M} \hat{\phi}_{t,m}$.

### 2.5.3 Support Vector Regression

Support-vector machines (SVM) were initially invented by Vapnik (1995) as a classification approach, and were later expanded to handle continuous response variables by Drucker et al. (1996) who introduced support-vector regressions (SVR). The objective in the $\varepsilon$-insensitive support-vector ($\varepsilon$-SV) regression, which takes its name from the underlying ($\varepsilon$-insensitive) loss function that is defined below, is to find a function $f(x)$ that deviates at most $\varepsilon$ from the observed $y_t$ for all the training data, and at the same time is as flat as possible. The linear function we are seeking to find, takes the form $f(x) = \langle w, x\rangle + b$ with $b \in \mathbb{R}$, where $\langle \cdot, \cdot \rangle$ denotes the dot product. The above problem can be formally defined as a minimization of a loss function plus a penalty (referred to as regularized risk):

$$f_0 := \text{argmin}_{f \in H} := \frac{1}{T} \sum_{i=1}^{T} L(x_i, y_i, f(x_i)) + \frac{\lambda}{2}\|w\|^2 \tag{16}$$

where $H$ is some function class, $\lambda > 0$ denotes the regularization constant, and $L(.)$ is the $\varepsilon$-insensitive loss function, described by:

$$L(x_i, y_i, f(x_i), w) = \begin{cases} 0 & \text{if } |y_i - f(x_i, w)| \leq \varepsilon \\ |y_i - f(x_i, w)| - \varepsilon & \text{otherwise.} \end{cases}$$

The problem can be rewritten in its Lagrangian form with the help of a dual set of variables. The reformulation of the problem to its dual form, further allows us to extend the SV machine to nonlinear functions, by replacing the dot product with a nonlinear kernel

function $k(x, x') := \langle \Phi(x), \Phi(x') \rangle$, where $\Phi(x)$ is a transformation that maps the training data $x_i$ to a (potentially) high-dimensional feature space. Applying the optimality conditions and following some derivations that can be tracked in Smola and Schölkopf (2004), we can restate the optimization problem of the nonlinear regression SVM in its dual Lagrangian form, where one has to find coefficients $\alpha_i$, $\alpha_i^*$, $i = 1, \ldots, T$ that maximize:

$$W = -\frac{1}{2} \sum_{i,j=1}^{T} (\alpha_i - \alpha_i^*)\big(\alpha_j - \alpha_j^*\big) k(x_i, x_j) - \varepsilon \sum_{i=1}^{T} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{T} y_i (\alpha_i - \alpha_i^*) \tag{17}$$

subject to $\sum_{i=1}^{T} (\alpha_i - \alpha_i^*) = 0$ and $\alpha_i, \alpha_i^* \in [0, C]$, where $\alpha_i, \alpha_i^*$ are Lagrange multipliers. To obtain the optimal solution, the Karush–Kuhn–Tucker (KKT) complementarity conditions are required. From the optimality conditions, we retrieve the $w$ parameter, $w = \sum_{i=1}^{T} (\alpha_i - \alpha_i^*) \Phi(x_i)$, yielding the following optimal solution for the function:

$$f(x) = \sum_{i=1}^{T} (\alpha_i - \alpha_i^*) k(x_i, x) + b.$$

### 2.5.4 Neural Networks

Long Short-Term Memory (LSTM) networks are a particular class of recurrent neural networks (RNNs) introduced by Hochreiter and Schmidhuber (1997), and further refined by Graves and Schmidhuber (2005). They employ units known as Constant Error Carousels (CECs), which facilitate stable gradient propagation and enable the network to capture long-term dependencies. This architecture makes LSTMs particularly well suited for modelling time series and other forms of sequential data.

The formulas that describe the components for a vanilla LSTM layer at time $t$, are given by:

$$\begin{aligned}
g_t &= \sigma_c(\boldsymbol{W}_g \mathbf{x}_t + \mathbf{R}_g \mathbf{h}_{t-1} + \boldsymbol{b}_g) && \text{cell candidate} \\
i_t &= \sigma_g(\boldsymbol{W}_i \mathbf{x}_t + \mathbf{R}_i \mathbf{h}_{t-1} + \boldsymbol{b}_i) && \text{input gate} \\
f_t &= \sigma_g(\boldsymbol{W}_f \mathbf{x}_t + \mathbf{R}_f \mathbf{h}_{t-1} + \boldsymbol{b}_f) && \text{forget gate} \\
\mathbf{c}_t &= f_t \odot \mathbf{c}_{t-1} + i_t \odot g_t && \text{cell state} \\
o_t &= \sigma_g(\boldsymbol{W}_o \mathbf{x}_t + \mathbf{R}_o \mathbf{h}_{t-1} + \boldsymbol{b}_o) && \text{output gate} \\
\mathbf{h}_t &= o_t \odot h(\mathbf{c}_t) && \text{hidden state}
\end{aligned}$$

$$\text{and } \hat{y}_{t+h/t} = \boldsymbol{U}_y \mathbf{h}_t + \boldsymbol{b}_y$$

where $\mathbf{c}_t$ denotes the cell state at time $t$, and $\mathbf{h}_t$ the hidden state. The hidden state is also referred to as the block or cell 'output', and the cell candidate ($g_t$) as the 'input' of the cell.

$\sigma_c, \sigma_g$ and $h$ are point-wise non-linear activation functions for the cell input, the gates and the cell state, respectively. The logistic sigmoid $\left(\sigma_g(x) = (1 + e^{-x})^{-1}\right)$ is used as the activation function for the gates, while the hyperbolic tangent ($\sigma_c(x) = h(x) = \tanh(x)$) is usually used as the cell input and output activation function. $\odot$ denotes element-wise multiplication of two vectors (Hadamard product). $\boldsymbol{W}, \mathbf{R}$, and $\boldsymbol{b}$ are the weights to be estimated. Specifically, matrices $\boldsymbol{W}$ and $\mathbf{R}$ contain the input and the recurrent weights, respectively, while $\boldsymbol{b}$ denotes the vector of the bias weights. The number of blocks (neurons) in each LSTM layer determines the dimension of hidden state in the corresponding layer. Assuming $N$ inputs and $H$ LSTM blocks, then the weights for the LSTM layer are: $\boldsymbol{W}_g, \boldsymbol{W}_i, \boldsymbol{W}_f, \boldsymbol{W}_o \in \mathbb{R}^{H \times N}$; $\mathbf{R}_g, \mathbf{R}_i, \mathbf{R}_f, \mathbf{R}_o \in \mathbb{R}^{H \times H}$; and $\mathbf{b}_g, \mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o \in \mathbb{R}^H$. At the final step, the prediction, $\hat{y}_{t+h}$, is calculated as the linear combination of the hidden states.

## 2.6 Mixed-Frequency Models

### 2.6.1 Sparse-group-LASSO-MIDAS

Babii et al. (2022) introduced a penalized regression model specifically tailored for high-dimensional settings involving mixed-frequency data. They proposed utilizing the sparse-group LASSO (sg-LASSO) estimator of Simon et al. (2013), which solves the following penalized least-squares problem:

$$\min_{b \in \mathbf{R}^p} \|\mathbf{y} - \mathbf{X}b\|_T^2 + \lambda \Omega(b) \tag{18}$$

where $\lambda \geq 0$ is the regularization parameter, and $\Omega$ the sg-LASSO penalty function:

$$\Omega(b) = \gamma |b|_1 + (1 - \gamma) \|b\|_{2,1},$$

where weight parameter $\gamma \in [0, 1]$ balances between the $\ell_1$ LASSO penalty ($\gamma = 1$) and the group LASSO norm ($\gamma = 0$; see, Yuan and Lin, 2006), $\|b\|_{2,1} = \sum_{G \in \mathcal{G}} |b_G|_2$ with $\mathcal{G}$ denoting the group structure. In a time series forecasting context the authors propose forming groups consisting of each covariate's lags. More formally, their suggested approach utilizes a *dictionary* which is a collection of functions used to define alternative sets of weights for the lag polynomials of the various covariates. When the forecasting problem involves data sampled at different frequencies, high-frequency information is incorporated into the model in a MIDAS fashion through high-frequency lag polynomials. Given these group structures, the sparse-group LASSO regularization enables a structured sparsity approach, allowing for variable selection both between groups, using the group LASSO penalty to identify the most relevant covariates, and within groups, using the standard LASSO penalty to determine the

shape of the (MIDAS) weight function.

# 3 Methods for Handling Mixed-Frequency Data

In empirical macroeconomic analyses involving variables measured at different frequencies, a key challenge is how to address the frequency mismatch to analyze relationships between these variables. To incorporate information from large mixed-frequency datasets into the ML models introduced above, I draw on methods from the distributed lag literature, which allow the integration of high-frequency time series (and their lags) into low-frequency regression equations. Table 2 summarizes the three alternative methods, employed in this study, for harmonizing uneven frequencies and handling the lags of covariates sampled at the same or higher frequencies relative to the target, and provides the labels for the respective information set used to train the competing ML models.

Table 2: List of Information Sets

| Acronym | Description |
|---|---|
| D1 | Equal-weighted Temporal Aggregation |
| D2 | Unrestricted Lag Polynomials |
| D3 | Legendre Polynomial Weights (3rd degree) |

Before introducing each method, it is essential to define certain terms to facilitate the subsequent discussion. I use the term *leading observations* to refer to any observations in a series that correspond to periods following the last available observation of the target variable, $t$. Let $q_i$ denote the number of available leading observations in series $i$, potentially sampled at high-frequency, and $t + w_i$ denote the final period for which data is available for predictor $i$. We define the term *leads* as the set of $q_i$ variables extracted, after applying the lag operator to the series containing the leading observations. For a high-frequency predictor $i$, the lag operator is defined as $L^{1/m}x_{t,i} = x_{t-1/m,i}$ where $m$ represents the frequency mismatch between the target and the predictor, The set of lagged variables referred to as leads is then obtained from $L^{j/m}x_{t+w_i,i}$ for $j = 0, 1, .., q_i - 1$. Given these definitions, $q_i > 0$ implies that covariate $i$ is released more timely than the GDP, with the first $q_i$ months of quarter $t + 1$ being available for that series. The notation for time periods adopted here references the low (aggregate) period, so both $t$ and $t + w$ correspond to quarters.

## 3.1 Single-frequency Information Set

**Deterministic Temporal Aggregation** The first information set (D1) evaluated in this study is constructed by combining predictors originally available at a quarterly frequency with the downsampled version of the high-frequency series, which are converted to quarterly by applying an equal-weighted average after any preprocessing steps. Formally, following the temporal aggregation literature (e.g., Chow and Lin, 1971), the conversion of high-frequency indicator data to aggregated low-frequency observations is achieved through a deterministic aggregator function $\psi(L^{1/m})$, applied in the lag operator $L^{1/m}$. In the context of a monthly-quarterly conversion, the expression for downsampling the monthly predictor $x_t^M$ to $x_t^Q$ takes the form:

$$x_t^Q = \psi(L^{1/3})x_t^M = \sum_{j=0}^{2} \omega_j L^{j/3} x_t^M,$$

with $\omega_j = 1/3$ providing the uniformly weighted average.

## 3.2 Mixed-frequency Information Sets

While ex-ante temporal aggregation is a straightforward and simple approach for forecasting with predictors sampled at different frequencies, deterministic aggregation of time series, can lead to loss of potentially valuable high-frequency information (see Marcellino, 1999). An alternative approach to handle mixed-frequency data, without resorting to resampling high-frequency series, is the mixed-frequency data sampling (MIDAS) regression framework proposed by Ghysels et al. (2005, 2006, 2007) and Andreou et al. (2010), which enables the estimation of dynamic equations in which high-frequency predictors are projected directly onto a low-frequency target. MIDAS regressions employ distributed lag structures specifically designed for high-frequency variables, allowing the weights for the aggregation of lags to be determined empirically. The MIDAS regression for forecasting a low-frequency target $h$ periods ahead using $N$ covariates $x_i$ potentially sampled at different frequencies, is given by:

$$y_{t+h} = \alpha + \sum_{p=0}^{P} \phi_p y_{t-p} + \sum_{i=1}^{N} \beta_i \psi\big(L^{1/m}; \omega_i\big) x_{t+w_i,i} + u_{t+h}, \tag{19}$$

where $\psi(L^{1/m}; \omega_i) = \sum_{j=0}^{V_i-1} \omega_i(\theta, j) L^{j/m}$ is the (high-frequency) lag polynomial, and $V_i$ is total number of (leads and) lags for $i$-th predictor. In the standard MIDAS framework, in order to ensure a parsimonious specification, the lag polynomial is typically specified as an exponential Almon lag polynomial (Lütkepohl, 1981), where the lag weights $\omega_j^\theta$ are defined by a functional form. Next, I describe two alternative methods from the recent MIDAS

literature for parameterizing polynomial weights, that provide suitable options for use with machine learning algorithms. Note that, although most ML models introduced earlier do not explicitly involve lag polynomials (with a few exceptions), however, the underlying sets of variables involved in the parameterization of the lag polynomials, as suggested by the two methods, can serve as information sets to train ML methods, 'framing' them as lag polynomials.

### 3.2.1 Unrestricted Lag Polynomials

An appealing MIDAS variant, suitable when the frequency mismatch is small, is the Unrestricted MIDAS (U-MIDAS) approach, first proposed by Koenig et al. (2003), and later formalized by Foroni et al. (2015), which parameterizes MIDAS polynomial weights without relying on functional distributed lag polynomials. The unrestricted model is obtained from Equation 19 by relaxing the functional restriction $\beta_i \psi\left(L^{1/m}; \omega_i\right)$, and replacing it with $\delta_i\left(L^{1/m}\right)$, arriving at

$$y_{t+h} = \alpha + \sum_{p=0}^{P} \phi_p y_{t-p} + \sum_{i=1}^{N} \sum_{j=0}^{V_i - 1} \delta_{j,i} L^{j/m} x_{t+w_i,i} + u_{t+h}. \tag{20}$$

where in the standard linear regression setup, Equation 20 can be estimated via OLS. Inspired by the U-MIDAS approach, this study adopts the idea and uses the information set obtained by gathering the leads and lags for each high and same frequency predictor to train the ML specifications introduced earlier. As such, the U-MIDAS information set (D2) is given by $(Z_1, \ldots, Z_N)$, where $Z_{t,i} = \left(L^{j/m} x_{t+w_i,i}\right)_{j \in [0,1,\ldots,V_i-1]}$.

### 3.2.2 Aggregation with Legendre Dictionaries

Babii et al. (2022) propose parameterizing the lag coefficients by expressing the MIDAS weight function as a linear combination of a collection of approximating functions $w_l(u)$ with $u \in [0, 1]$, referred to as *dictionary*. Using their approach, the linear MIDAS regression (Eq. 19), can be rewritten as:

$$y_{t+h} = \alpha + \sum_{p=0}^{P} \phi_p y_{t-p} + \sum_{i=1}^{N} \sum_{l=1}^{L} \beta_{i,l} \frac{1}{V_i} \sum_{j=0}^{V_i - 1} w_l\left(\frac{j}{m}\right) L^{j/m} x_{t+w_i,i} + u_{t+h}, \tag{21}$$

with $\{w_l : l = 1, \ldots, L\}$, and $L \leq V_i$ the dictionary size, which is determined by the polynomial degree. The authors recommend using orthogonal polynomials for the dictionary, such as Legendre polynomials that can reduce multicollinearity. Notably, $L \leq V_i$ leads to dimensionality reductions, which is especially advantageous for covariates sampled at very

high frequencies, like weekly or daily data.[17] When dimensionality is not an issue, such as when the forecasting problem involves only a few covariates, Equation 21 can be estimated using OLS. However, as discussed in Section 2.6.1, in high-dimensional time-series settings, Babii et al. (2022) propose using the sg-LASSO estimator. Given the dictionary choice, Equation 21 is simply a linear regression, estimated using an information set obtained by weighting the lagged values of each covariate with a matrix generated from the weight function, yielding various temporally aggregated versions of each predictor. This set serves as the second information set proposed in this study for training the various ML algorithms introduced in the previous section. Formally, let $W = (w_l(j/m)/V_i)_{j\in[0,V_i-1],l\in[1,L]}$ denote the $V_i \times L$ matrix of weights, and $Z_i$ the $T \times V_i$ matrix of the $i$-th covariate and its lags. The Legendre-formulated information set (D3) is then obtained by aggregating each row of $Z_i$ using dictionary $W$, as follows: $(Z_1W, \ldots, Z_NW)$.[18,19]

# 4 Data

To construct the mixed-frequency dataset, indicators were combined from two comprehensive macroeconomic datasets (McCracken and Ng, 2020, 2016) that replicate the coverage in the 'Stock-Watson' datasets (Stock and Watson, 2005, 2012): the FRED-MD, which includes variables sampled at a monthly frequency, and the FRED-QD, which contains a collection of quarterly series. Although the two datasets have been widely used in the literature to evaluate the performance of alternative forecasting methodologies in data-rich environments, this study is the first attempt to use the two in tandem and leverage the combined informational content to guide the model selection process within a data-rich mixed-frequency framework.

Since the FRED-MD and FRED-QD datasets were not originally designed to be used together as a bundle of mixed-frequency panels, numerous indicators in the FRED-QD vintages are temporally-aggregated versions of the series found in FRED-MD. To isolate the unique information from both datasets and create a coherent comprehensive set of macroeconomic variables sampled at different frequencies, the following steps were undertaken. First,

[17] For instance, if considering a quarter's worth of past information for the lagged terms of the predictors, a daily financial indicator would require approximately 66 lags (one for each trading day), while a weekly series would require about 13 lags.

[18] Although flexible, aggregation based on Legendre polynomials still imposes predefined parametric restrictions on the shape of the polynomial, making it more restrictive compared to the U-MIDAS approach. Nonetheless, when differences in sampling frequencies are large, the dimensionality reduction and mitigation of multicollinearity can offer potential benefits.

[19] To get an idea of the flexibility of Legendre polynomials in approximating different lag structures, Appendix A provides an example of the various sets of weights generated by Legendre polynomials when the degree is set to 3.

the FRED-MD and FRED-QD vintages for April 2021 were downloaded.[20] The datasets initially consisted of unbalanced panels featuring 135 monthly and 248 quarterly indicators, respectively. Subsequently, overlapping variables were removed from the corresponding panels, and series available at a higher sampling frequency on the FRED data server (https://fred.stlouisfed.org) were downloaded at their highest available frequency. These steps resulted in a mixed-frequency dataset with a quarterly and a monthly panel consisting of 87 and 171 series, respectively, including 32 financial market indicators.

### 4.1 Pseudo Real-Time Vintages

In the absence of a set of actual vintages suitable for contexts requiring the use of rich mixed-frequency datasets, in order to recreate an experimental setup that captures how economic activity is monitored today, publication release delays for each series in the two unbalanced panels were inferred from the online FRED metadata and then applied to the individual predictors. This resulted in the creation of 219 monthly pseudo real-time vintages spanning the period January 2003 to March 2021. Each vintage consists of a pair of unbalanced monthly and quarterly panels that replicate the availability of economic statistics at the last day of each month. The vintage were designed to reflect the ragged-edge structure that forecasters encounter in practice, closely mimicking the real-time informational inflow.

Monitoring economic activity in real-time implies nowcasting and forecasting multiple times within the quarter, frequently updating the predictions to incorporate newly released information. Given the monthly periodicity of the constructed vintages, this article assumes that the forecaster tracks GDP progress on a monthly basis. This translates into three prediction and update exercises for GDP nowcast and forecasts, each assumed to be conducted at the end of each month. Table 3 provides an illustration of the real-time data inflow involved in this process. Specifically, it presents the month-to-month evolution across three consecutive vintages (January to March 2021) for four quarterly and four monthly indicators from the two panels compiled for this study.

## 5 Out-Of-Sample Forecasting Setup

To compare the ML algorithms and the various information sets, the 219 monthly real-time vintages, are utilized to perform two pseudo out-of-sample (POOS) evaluation experiments. The first experiment replicates quarterly GDP monitoring, assuming GDP is tracked once at each quarter, using only the vintages corresponding to the third month of the quarter

[20] https://research.stlouisfed.org/econ/mccracken/fred-databases/.

Table 3: Real-time Data Inflow in Monthly Vintages

| | GDPC1 | OUTBS | NWPIx | AHETPIx | MZMSL | UNRATE | CMRMTSPLx | UMCSENTx |
|---|---|---|---|---|---|---|---|---|
| **Vintage 31/01/2021** | | | | | | | | |
| 30/09/2020 | 18597 | 118 | 683 | 22 | 21250 | 8 | 1564146 | 80 |
| 31/10/2020 | - | - | - | - | - | 7 | 1572500 | 82 |
| 30/11/2020 | - | - | - | - | - | 7 | 1569672 | 77 |
| 31/12/2020 | - | - | - | 22 | - | 7 | - | 81 |
| 31/01/2021 | - | - | - | - | - | - | - | 79 |
| 28/02/2021 | - | - | - | - | - | - | - | - |
| 31/03/2021 | - | - | - | - | - | - | - | - |
| **Vintage 28/02/2021** | | | | | | | | |
| 30/09/2020 | 18597 | 118 | 683 | 22 | 21250 | 8 | 1564146 | 80 |
| 31/10/2020 | - | - | - | - | 21369 | 7 | 1572500 | 82 |
| 30/11/2020 | - | - | - | - | - | 7 | 1569672 | 77 |
| 31/12/2020 | - | 120 | - | 22 | - | 7 | - | 81 |
| 31/01/2021 | - | - | - | - | - | 6 | - | 79 |
| 28/02/2021 | - | - | - | - | - | - | - | 77 |
| 31/03/2021 | - | - | - | - | - | - | - | - |
| **Vintage 31/03/2021** | | | | | | | | |
| 30/09/2020 | 18597 | 118 | 683 | 22 | 21250 | 8 | 1564146 | 80 |
| 31/10/2020 | - | - | - | - | 21369 | 7 | 1572500 | 82 |
| 30/11/2020 | - | - | - | - | 21565 | 7 | 1569672 | 77 |
| 31/12/2020 | 18794 | 120 | - | 22 | - | 7 | 1566283 | 81 |
| 31/01/2021 | - | - | - | - | - | 6 | - | 79 |
| 28/02/2021 | - | - | - | - | - | 6 | - | 77 |
| 31/03/2021 | - | - | - | - | - | - | - | 85 |

NOTES: The table presents the real-time data inflow across three consecutive monthly vintages. Four quarterly variables (including the target variable) and four monthly variables with varying release delays, have been selected. The highlighted row marks the last available observation of the target variable (GDPC1) in each vintage. Observations below that row correspond to leading information. The availability of leading information differs across vintages, with some months offering timely data corresponding to one quarter ahead of GDP, and others two quarters ahead.

(referred to hereafter as 'EoM3,' i.e., end-of-month 3), namely March, June, September, and December. The second OOS evaluation exercise assumes that tracking of GDP occurs at higher observation frequency by employing the full set of 219 month-end vintages, thereby replicating monthly monitoring, where GDP nowcasts and forecasts are (re)calculated three times per quarter. Out-of-sample nowcasts and forecasts are generated and evaluated for the periods 2003:M01 to 2021:M03, encompassing 219 vintages for the monthly experiment, and 2003:Q1 to 2021:Q1, covering 73 vintages for the quarterly evaluation. Model performance is compared for the nowcast and for forecasts up to four quarters ahead. Both out-of-sample experiments are conducted using a rolling-window estimation approach. Regarding the effective sample size, estimation of direct multi-step forecasting models is based on a window of $R_d = 132 - h$ quarterly observations. For iterated specifications, the training sample size does not depend on the forecast horizon, but varies with the number of lags included in the model. Specifically, the window length for the AR(P) and BVAR(P) models is given by $R_{it} = 132 - P - 1$, where $P$ represents the number of lags.

The accuracy of point forecasts obtained from the two pseudo out-of-sample (POOS) exercises is assessed using two error measures. First, the root mean squared error (RMSE) is employed as the primary metric to compare the performance between different models and alternative information sets. Second, to ensure the robustness of the results, the mean absolute error (MAE) is reported alongside the RMSE, given its reduced sensitivity to large forecast errors. The two error measures are defined as follows:

$$\text{RMSE}_{n,m} = \sqrt{\frac{1}{v_T - v_0 + 1} \sum_{v=v_0}^{v_T} \hat{e}_{v,n,m}^2}, \tag{22}$$

$$\text{MAE}_{n,m} = \frac{1}{v_T - v_0 + 1} \sum_{v=v_0}^{v_T} |\hat{e}_{v,n,m}|, \tag{23}$$

where $\hat{e}_{v,n,m} = y_{v,n} - \hat{y}_{v,n,m}$, and $\hat{y}_{v,n,m}$ is the $n$-quarters ahead prediction for GDP growth computed in period $v$, obtained by model $m$. $v_0$ and $v_T$ represent the first and last vintages, respectively, for which the $n$-quarters ahead predictions were generated in each pseudo out-of-sample experiment. For the monthly evaluation, $v_0 =$ 2003:M01 and $v_T =$ 2021:M03$-3n$, while for the quarterly OOS experiment, which is based on the vintages of the 3rd month of each quarter, $v_0 =$ 2003:Q1 and $v_T =$ 2021:Q1$-n$.[21] In the out-of-sample evaluation that uses all the monthly vintages, since the target is observed at a quarterly frequency, the

[21] Given that the last realized GDP observation in the sample corresponds to 2021Q1, the number of out-of-sample nowcasts and forecasts used to calculate RMSE and MAE in each OOS experiment depends on the forecast horizon ($n$). Specifically, for the monthly out-of-sample evaluation, the metrics are calculated over $219 - 3n$ predictions, while for the quarterly evaluation, they are based on $73 - n$ predictions.

monthly forecasts are evaluated against the same realized quarterly value. Consequently, the forecast errors used in the calculation of the two error measures will share the same realized GDP value for all months within the same quarter.

Evaluating and comparing the performance of multiple candidate models necessitates additional testing procedures to facilitate comparisons and ascertain that observed differences in predictive accuracy across models are statistically significant, in such settings. To elaborate, in the quest for the best model, the empirical forecaster inevitably conducts repeated searches over the same set of historical time-series data, raising the issue of 'data snooping.' This occurs when the repeated search leads to a statistically significant outperformance of a single model that is solely the outcome of luck.[22] The solution to data snooping turns to the concept of 'multiple-testing,' where the null hypothesis is formulated to involve all models under consideration, rather than strictly focusing on a pair of models (White, 2000; Hansen, 2005). A different approach to multiple testing advocated by Hansen et al. (2011) focuses on constructing the set that contains the "best" model(s) with a given level of confidence $(1-\alpha)$, allowing for the possibility that more than one models can be the "best" (i.e., statistically indistinguishable from the best-performing model(s)). The set that contains the superior models with equivalent performance is referred to as the 'model confidence set' (MCS) and is denoted by $M^*_{1-\alpha}$. To address the inherent multiple-testing problem, whenever possible, I employ the MCS test to identify the subset of models whose performance is statistically indistinguishable in each forecast horizon. The distribution of the test statistic for the MCS is generated using a circular block bootstrap procedure with the number of resamples set to 1000, and the block length to 8. The significance level in the MCS test is set to $\alpha = 40\%$, implying that the estimated model confidence set, $M^*_{60}$, contains the true superior models (i.e., it is the true MCS) with at least 60% probability.

# 6 Results

## 6.1 Baseline Models

Tables 4 and 5 present the performance of the models drawn from the 13 distinct machine learning model classes trained using the quarterly information set D1, along with the Bayesian VARs, the standard and target ARDI models, and the univariate benchmarks. The figures reported in Tables 4 and 5 correspond to RMSE and MAE ratios, respectively, relative to the AR(1) benchmark, derived from the quarterly OOS evaluation based on the

[22]Data snooping arises even when attempting to improve upon the most promising techniques identified in previous research conducted on the same dataset. As Abu-Mostafa et al. (2012, p. 175) put it, "Although you haven't even seen the data yet, you are already guilty of data snooping."

73 vintages spanning the period from Mar-2003 to Mar-2021. Specifically, each table shows the error metrics for the $n$-quarters-ahead predictions (with $n = 0$ denoting the nowcast), along with the average relative error across all five horizons, displayed in column 6. Cells highlighted in grey indicate the models included in the 60% MCS for each horizon, computed based on the quadratic and absolute losses, in Tables 4 and 5, respectively. The final column in both tables shows the average p-value of the MCS test over the five horizons. Higher p-values signify weaker evidence of inferior predictive performance relative to other models in the set. Consequently, models with higher MCS p-values are stronger candidates. Bold entries in columns 1 to 6 denote the models with the lowest relative error for each step horizon and on average, while the bold figure in the last column corresponds to the largest average MCS p-value across models. To provide a sense of the magnitude of forecast errors, the absolute RMSE and MAE values are reported for the AR(1) model. Finally, note that all models presented in the tables of this subsection have been estimated only using quarterly data, with the exception of the sg-LASSO, which was trained on the mixed-frequency Legendre-aggregated set D3, following the recommendation of Babii et al. (2022).

I start with a sanity check, confirming that key results in the two tables align with findings from past studies. For instance, block-wise boosting outperforms its component-wise counterpart both overall and across individual horizons, consistent with Bai and Ng (2009). Similarly, the BVAR with the flexible covariance structure outperforms its homoscedastic counterpart, corroborating evidence from multiple studies (e.g., Carriero et al., 2016; Chan, 2020). Additionally, incorporating a pre-selection step prior to factor extraction improves the performance of standard diffusion index models, as suggested in Bai and Ng (2008). Notably, all these findings are robust to the choice of error criterion, whether RMSE or MAE is used. Shifting attention to the univariate benchmarks, the AR(1) emerges as the strongest candidate among the autoregressive specifications. However, assuming that GDP in log-levels follows a random walk process provides a significant improvement in predicting GDP growth compared to the rest univariate models, particularly for the nowcast and the first two quarters ahead forecasts. The strong performance of the RW model relative to other univariate specifications (and more broadly) aligns with the findings of D'Agostino et al. (2007), whose conclusions remain relevant today. This outcome is largely attributable to the fact that U.S. output growth has remained relatively stable throughout most of the post-1985 era, with the exception of two major episodes: the 2007-08 financial crisis, and the 2020 economic disruption caused by the COVID-19 pandemic.

Turning to the comparative performance between models, when the objective is to nowcast, boosting diffusion indices with a linear base procedure and block-wise treatment for incorporating factor dynamics, outperforms all other ML candidates and benchmarks, achiev-

ing a 40% reduction in RMSE relative to the AR(1) and 19% relative to the RW. Following closely in terms of nowcasting performance is the class of ARDI models, with the single targeted-factor ARDI model exhibiting an RMSE only marginally smaller than that of BBoost-D1F. While when predicting the current quarter, all ML methods, except adaLASSO, LSTM, and sg-LASSO, produce more precise estimates compared to the RW; however, when forecasting the next quarter, the majority of models including standard benchmarks find it challenging to beat the RW, with only a few exceptions of ML algorithms.[23] Specifically, for $n = 1$ only 7 out of the 20 non-univariate models manage to outperform the RW. When forecasting two quarters ahead, most models achieve similar accuracy, with only the elastic net, the linear bagging ensemble, and the ridge regression delivering somewhat more accurate forecasts. Beyond the 2-quarters-ahead prediction, the majority of models are at most as accurate as the AR(1) benchmark and the rolling 30-year average growth rate, with almost none achieving a ratio smaller than one.This finding is supported by the fact that the same pattern is observed across both error metrics and also because most models are included in the MCS, indicating statistically indistinguishable predictive accuracy. Examining the across-horizon statistics, the block boosting and ridge regression models achieve the best and second-best performance, respectively, in terms of both average 5-horizon RMSE and average MCS p-value. Similar observations emerge when comparing model performance using the MAE criterion. While block boosting displays somewhat higher MAE compared to its relative performance as captured by the RMSE, it nevertheless maintains its status as the best performer in terms of average 5-horizon performance and average MCS p-value. The models in the ARDI class also demonstrate notable and consistent performance, with the single-factor target ARDI achieving the lowest MAE for the nowcast horizon, and the two-factor target ARDI delivering the second lowest average MAE, and the second highest MCS p-value.

The comparative performance results presented in Tables 4 and 5 indicate that machine learning methods, when combined with a large set of covariates held at the same frequency with the target variable, offer notable gains in predicting both current and future GDP growth compared to standard benchmarks and state-of-the-art models. While the primary

[23] One possible explanation for the poor performance of the sg-LASSO is that predictive relationships are not adequately captured by sparse representations. This hypothesis is in line with the evidence presented in Giannone et al. (2021) who studied the relevance of sparsity across several economic prediction problems. This is also supported by the observation that several best-performing specifications correspond to models that do not enforce sparsity, including the factor-boosting model. To give an idea of the degree of sparsity imposed by sg-LASSO, Appendix B presents sparsity patterns for different sparsity-inducing algorithms. A comparison between the variables selected by sg-LASSO-MIDAS (Fig. 9) and those selected by applying hard-thresholding preselection on information set D3 (Fig. 8) reveals a significant reduction in the relative number of predictors included in sg-LASSO.

gains of utilizing machine learning are observed mainly in nowcasting, some benefits also extend to short-term forecasting up to two quarters ahead.

## 6.2 Horse Race: The full picture

In this section I turn to assess the potential gains in forecasting accuracy coming from two modifications: (1) the different schemes for incorporating predictors sampled at different frequencies, and (2) forming parsimonious ML specifications using a reduced set of predictors comprised of only factors on the RHS. The various components and algorithms are combined in an extended horse race which ranks the performance of all specifications resulting from combining the distinct ML methodologies with the three information sets D1, D2, and D3 that correspond to alternative methods for handling the mixed-frequency nature of the setup, along with their factor-only counterparts, denoted D1F, D2F, and D3F. A total of six information sets are formed which are then combined with the 13 ML algorithms examined in this article. The combined ML specifications together with the standard econometric techniques and workhorse benchmarks considered herein, result in a ranking containing a total of 85 specifications. The full set of models is evaluated in two out-of-sample experiments. The first assumes that monitoring of economic activity occurs once at the end of each quarter, while the second considers tracking GDP at a monthly frequency, i.e., three times within the quarter.

### 6.2.1 Real-Time Quarterly Monitoring

Tables 6 and 7 present the 20 best performing specifications ranked according to their average relative RMSE and MAE, respectively, derived from the horse race based on the out-of-sample evaluation that utilises the real-time quarter-end vintages. Models have been ranked with respect to column labelled 'avg', which corresponds to the average relative error over all five horizons.[24] In order to retain conciseness, the tables present the 20 models most relevant to each ranking, corresponding to approximately the upper 25% distribution of models. Nevertheless, the analysis in this section also draws conclusions based on the full rankings, which are available upon request. For brevity, in what follows, parentheses are used in the notation of the information sets to collectively refer to any of the three information sets and its factor-only counterpart. For example, D1(F) denotes both D1 and D1F.

[24]Since the tables rank models based on their 5-horizon average performance, bold entries may be absent in some columns. This occurs because the best-performing specification might appear further down the ranking. For example, the models with the lowest RMSEs for the 2- and 4-quarter ahead forecasts are not included in the condensed ranking shown in Table 6.

Table 4: Forecasting Errors: RMSE, Mar-2003 to Mar-2021

| Models | n=0 | n=1 | n=2 | n=3 | n=4 | avg | avgMCS |
|---|---|---|---|---|---|---|---|
| AR(1) | 2.08 | 1.73 | 1.68 | 1.58 | 1.59 | - | 0.55 |
| AR(4) | 1.06 | 1.04 | 1.01 | 1.00 | 1.00 | 1.02 | 0.52 |
| AR(BIC) | 1.00 | 1.01 | 1.00 | 1.00 | 1.00 | 1.00 | 0.49 |
| AR(CV) | 1.03 | 1.01 | 1.00 | 1.00 | 1.00 | 1.01 | 0.45 |
| RW | 0.74 | 0.90 | 0.93 | 1.00 | 1.00 | 0.92 | 0.55 |
| ARDI(1) | 0.62 | 0.94 | 0.93 | 1.00 | 1.00 | 0.90 | 0.56 |
| ARDI(2) | 0.64 | 0.94 | 0.93 | 1.00 | 1.00 | 0.90 | 0.56 |
| T.ARDI(1) | 0.60 | 0.96 | 0.93 | 1.00 | 1.00 | 0.90 | 0.56 |
| T.ARDI(2) | 0.61 | 0.91 | 0.93 | 1.00 | 0.99 | 0.89 | 0.57 |
| BVAR-Minn | 0.70 | 0.95 | 0.93 | 1.00 | 1.02 | 0.92 | 0.52 |
| BVAR-CSV | 0.65 | 0.91 | 0.93 | 1.01 | 1.01 | 0.90 | 0.43 |
| BBoost-D1F | **0.60** | **0.88** | 0.93 | 1.00 | 1.00 | **0.88** | **0.78** |
| CBoost-D1F | 0.66 | 0.89 | 0.93 | 1.02 | 1.00 | 0.90 | 0.46 |
| CSR-D1 | 0.66 | 0.92 | 0.93 | 1.01 | 1.01 | 0.91 | 0.43 |
| Bag-D1 | 0.69 | 0.90 | 0.91 | 0.99 | 1.00 | 0.90 | 0.66 |
| BTree-D1 | 0.72 | 0.90 | 0.94 | 1.01 | 1.03 | 0.92 | 0.31 |
| RF-D1 | 0.69 | 0.90 | 0.93 | 1.00 | 1.00 | 0.91 | 0.48 |
| SVR-D1 | 0.73 | 0.90 | 0.92 | **0.99** | **0.99** | 0.91 | 0.63 |
| Ridge-D1 | 0.66 | 0.91 | 0.91 | 1.00 | 1.00 | 0.89 | 0.67 |
| LASSO-D1 | 0.67 | 0.98 | 0.92 | 1.10 | 1.07 | 0.95 | 0.19 |
| AdaLASSO-D1 | 0.77 | 0.92 | 0.98 | 1.06 | 1.01 | 0.95 | 0.26 |
| EN-D1 | 0.68 | 0.90 | **0.88** | 1.05 | 1.04 | 0.91 | 0.43 |
| AdaEN-D1 | 0.73 | 0.91 | 0.94 | 1.01 | 1.00 | 0.92 | 0.46 |
| LSTM-D1 | 0.76 | 0.91 | 0.94 | 1.00 | 1.02 | 0.93 | 0.48 |
| SgLASSO-D3 | 0.75 | 0.91 | 0.93 | 1.00 | 1.00 | 0.92 | 0.40 |

NOTES: The table reports the relative RMSE for the n-steps ahead prediction, with $n = 0$ reflecting the nowcast. Evaluation exercise conducted as if predictions were calculated every end of the 3rd month of each quarter. The average relative RMSE over all horizons is also given. The last column reports the average p-value of the model confidence set (MCS) test, proposed by Hansen et al. (2011), over all horizons and for squared losses. Highlighted cells denote the models that belong to the 60% MCS (i.e. the set of superior models with equal performance). A small p-value suggests that the row model is unlikely to be a member of the MCS. Error measures are reported relative to the AR(1), while for the AR(1) benchmark the absolute RMSEs are given. Figures in bold show the model with the best statistic, that is, lowest relative error measure, and largest average MCS p-value.

Table 5: Forecasting Errors: MAE, Mar-2003 to Mar-2021

| Models | n=0 | n=1 | n=2 | n=3 | n=4 | avg | avgMCS |
|---|---|---|---|---|---|---|---|
| AR(1) | 0.74 | 0.72 | 0.71 | 0.66 | 0.66 | - | 0.42 |
| AR(4) | 1.11 | 1.04 | 1.01 | 1.01 | 1.00 | 1.03 | 0.46 |
| AR(BIC) | 1.00 | 1.01 | 1.01 | 1.01 | 1.00 | 1.01 | 0.42 |
| AR(CV) | 1.06 | 1.01 | 1.00 | 1.01 | 1.00 | 1.02 | 0.45 |
| RW | 0.87 | 0.90 | 0.92 | **0.99** | **1.00** | 0.94 | 0.69 |
| ARDI(1) | 0.80 | 0.93 | 0.92 | 1.00 | 1.00 | 0.93 | 0.62 |
| ARDI(2) | 0.79 | 0.93 | 0.94 | 1.02 | 1.00 | 0.93 | 0.58 |
| T.ARDI(1) | **0.75** | 0.94 | 0.93 | 1.01 | 1.06 | 0.94 | 0.68 |
| T.ARDI(2) | 0.78 | 0.87 | 0.91 | 1.01 | 1.00 | 0.91 | 0.71 |
| BVAR-Minn | 0.80 | 1.01 | 0.94 | 1.05 | 1.10 | 0.98 | 0.65 |
| BVAR-CSV | 0.76 | 0.93 | 0.97 | 1.05 | 1.09 | 0.96 | 0.52 |
| BBoost-D1F | 0.78 | **0.81** | 0.93 | 1.01 | 1.01 | **0.91** | **0.79** |
| CBoost-D1F | 0.80 | 0.87 | 0.95 | 1.06 | 1.01 | 0.94 | 0.51 |
| CSR-D1 | 0.75 | 0.93 | 0.98 | 1.05 | 1.08 | 0.96 | 0.52 |
| Bag-D1 | 0.77 | 0.87 | **0.90** | 1.02 | 1.06 | 0.93 | 0.68 |
| BTree-D1 | 0.85 | 0.95 | 1.00 | 1.04 | 1.13 | 1.00 | 0.21 |
| RF-D1 | 0.78 | 0.90 | 0.93 | 1.02 | 1.02 | 0.93 | 0.60 |
| SVR-D1 | 0.85 | 0.91 | 0.92 | 1.00 | 1.01 | 0.94 | 0.56 |
| Ridge-D1 | 0.78 | 0.97 | 0.96 | 1.10 | 1.14 | 0.99 | 0.31 |
| LASSO-D1 | 0.76 | 1.26 | 1.16 | 1.38 | 1.27 | 1.17 | 0.20 |
| AdaLASSO-D1 | 0.90 | 1.02 | 1.13 | 1.18 | 1.09 | 1.06 | 0.11 |
| EN-D1 | 0.77 | 0.87 | 0.98 | 1.19 | 1.13 | 0.99 | 0.27 |
| AdaEN-D1 | 0.79 | 0.88 | 0.96 | 1.05 | 1.00 | 0.94 | 0.51 |
| LSTM-D1 | 0.92 | 0.95 | 1.02 | 1.09 | 1.09 | 1.02 | 0.33 |
| SgLASSO-D3 | 0.91 | 0.95 | 0.94 | 1.02 | 1.02 | 0.97 | 0.47 |

NOTES: The table reports the relative MAE for the n-steps ahead prediction, with $n = 0$ reflecting the nowcast. Evaluation exercise conducted as if predictions were calculated every end of the 3rd month of each quarter. The average relative MAE over all horizons is also given. The last column reports the average p-value of the model confidence set (MCS) test, proposed by Hansen et al. (2011), over all horizons and for absolute losses. Highlighted cells denote the models that belong to the 60% MCS (i.e. the set of superior models with equal performance). A small p-value suggests that the row model is unlikely to be a member of the MCS. Error measures are reported relative to the AR(1), while for the AR(1) benchmark the absolute MAEs are given. Figures in bold show the model with the best statistic, that is, lowest relative error measure, and largest average MCS p-value.

Starting with the ranking based on the average RMSE, at the top of the list is the block-wise boosting algorithm trained on the set of quarterly factors, D1F. The BBoost-D1F specification ranks first based on its 5-horizon average performance, achieves the lowest RMSE both for the nowcast and the 1-quarter ahead forecast, and holds the second largest MCS p-value among the 85 models, all of which collectively underscore its strong performance. Overall, within the first 20 specifications that deliver the lowest forecast errors over the 5 horizons, there is a significance presence of linear ML methodologies. Specifically, 16 out of the 20 best performers involve some form of dynamic linear regression model such as factor-augmented ARs, bagging of linear regressions, ridge regressions, CSR, LASSO, EN and the LASSO/EN adaptive variants. With the exception of the two L2 boosting algorithms that are based on linear base procedures, only two of the remaining specifications that make it to the condensed ranking with the 20 best performers, correspond to nonlinear ML models. The two specifications are boosting trees and random forests, and are found at the bottom of the list, as they rank 19th and 20th, respectively. Examining the representation of different information sets in the top 25% of models ranked by 5-horizon average RMSE, the majority of models are trained on the quarterly factor, D1F. Specifically, 13 models, including the 4 ARDI specifications, are trained on D1F, 2 on D1, and 3 on D2, while the remaining two models in the top 20 each corresponds to a linear bagging specification, one trained on D2F and the other on D3. Moreover, reviewing the full ranking, the constant growth model is found on the 44th position, while the two Bayesian VAR alternatives, the common stochastic volatility and the homoscedastic VAR, rank 22nd and 49th, respectively. It is noteworthy that the BVAR-CSV makes it among the top 10 specifications when models are ranked by their nowcasting performance, instead of the 5-horizons average.

Turning to the top quartile of models as ranked in Tables 7 and **??** that present the ranking with respect to MAE alternative, one notable observation is that the prediction given by the constant growth model, goes from the 44th position in the previous ranking, up to the 21st position. This comparative improvement of the RW model (relative to models with similar RMSE values) aligns with expectations for two reasons. First, the rolling average serves as a reasonable proxy for future growth under normal economic conditions, which is what prevails during the majority of times covered in our OOS experiment. Second, while average growth generally provides a poor approximation near peaks or troughs when economies are experiencing periods of heightened growth or contraction, the MAE criterion does not penalize those large forecasts errors more heavily, unlike RMSE. Nevertheless, in terms of the structure of the top performing algorithms and information sets when using the MAE alternative, a similar picture emerges to that observed in the RMSE ranking. The strong presence of linear models among the top 20 specifications persists, with 13 specifications cor-

responding to linear models, and the remaining 7 specifications to nonlinear ML algorithms. The composition of models is also similar to that observed in Table 6, with the linear models found in the top quartile representing the following model classes: diffusion index models with and without target-factors, linear bagging, and different types of penalized regressions. Among the nonlinear models, the least-squares block boosting algorithm consistently ranks among the top performers, delivering the 2nd best performance based on average MAE, as well as the 5th smallest nowcast error, and the 5th largest MCS p-value. Additionally, RF and SVR are also present, with 5 out of the 7 nonlinear specifications corresponding to the RF algorithm trained on different information sets. Given that the random forest algorithm only appears in the top quartile in the MAE ranking, it is worth noting that this could be attributed to a few isolated large errors disproportionately affecting its RMSE performance, rather than indicating a general inability of the algorithm to provide reliable predictions. Regarding the frequency with which different information sets appear in the quartile with the best performers, as captured by the 5-horizon average MAE, 12 models are trained on D1(F), including the 4 ARDI models all of which make it to the top 25%, 5 are trained on D2(F), and 3 on D3(F). Excluding the 4 ARDI specifications, 7 of the 20 top performers are trained on factor-only information sets, with 5 of those corresponding to D1F. The frequent appearance of D1F among the top-performing specifications across both RMSE and MAE rankings, almost irrespective of the algorithm used, underscores the effectiveness of relying exclusively on quarterly factors derived from the temporal aggregation of predictors in large mixed-frequency datasets. This consistency highlights D1F as a preferred information set, delivering robust performance across multiple machine learning approaches. Finally, the BVAR model that allows for heteroscedasticity ranks 43rd in the MAE-ordered list but rises to 6th place when models are ranked in terms of their nowcasting performance.

Figure 3 presents a scatter plot summarizing the results of the horse race by combining the two error measures previously reported in the tables of this section. Each point illustrates the average performance of a model across the five horizons, with the x-axis showing the average MAE ratio and the y-axis showing the average RMSE ratio, both relative to the AR(1) benchmark. Lower values on both axes indicate better performance, making models closer to the origin the stronger candidates according to both metrics.[25] The models are categorized by the information set used for training, into D1(F), D2(F), D3(F), and Other, with each represented by a distinct colour. The 'Other' category (depicted as red dots) is reserved for the standard econometric models and benchmarks (i.e., ARs, ARDIs, BVARs,

[25]The upper bounds of the axes are restricted to display only the most relevant models out of the 85 evaluated. Specifically, the plot shows models that lie within one standard deviation from the median of the averaged RMSE and MAE ratios.

and the RW). While ARDI models could be associated with D1(F), they are nevertheless based on fewer factors than those included in D1F, which contains the full set of optimally selected factors. For this reason, they are classified within the general 'Other' category. Overall, the figure reinforces the key insight from the analysis thus far: machine learning models outperform standard workhorse econometric models, including state-of-the-art BVARs. Moreover, ARDI models estimated on quarterly data remain competitive, serving as reliable baselines against which the additional complexity of machine learning algorithms and alternative information sets should be evaluated. The concentration of blue and red dots positioned closest to the origin highlights the effectiveness of temporally aggregating predictors as a robust technique for handling mixed-frequency data, performing well with various algorithms. Meanwhile, the green dots concentrated toward the middle of the plot suggest that incorporating high-frequency lags in an unrestricted manner serves as a viable alternative technique. In contrast, models trained on D3 and its variation (e.g., D3F) are more dispersed toward higher RMSE and MAE values, indicating weaker predictive performance. Focusing on the bottom-left corner of the plot, a distinct cluster of models, that stand apart from the majority of other candidates, emerges. The group of models within the region bounded approximately by a horizontal line at 0.9 on the y-axis and a vertical line at 0.935 on the x-axis represent the models with the best performance taking a balanced view considering both average RMSE and average MAE. Examining the composition of specifications within the rectangular region, several key observations emerge. First, BBoost-D1F consistently outperforms other models, achieving superior performance across both error measures with a strong margin. Additionally, the two ARDI models with targeted predictors deliver strong results, particularly T.ARDI(2), which excels in both MAE and RMSE metrics. Linear bagging-based models also demonstrate robust and consistent performance across both measures and for most information sets. Notably, all Bag-D1, Bag-D2, Bag-D3, and Bag-D1F models are positioned close to the origin, highlighting the efficacy of linear ensemble methods for nowcasting and forecasting macroeconomic aggregates. Additionally, reclassifying ARDI models within the D1(F) category, the vast majority of models in the rectangular region containing the top performers is based on the quarterly information set D1(F). This, once again, underscores the effectiveness of D1(F) as a reliable information set for achieving superior predictive accuracy.

Finally, it is noteworthy that while machine learning algorithms are inherently capable of handling high-dimensional datasets, the robust and consistent performance of linear bagging models, that are built on a parsimonious base learner using only a handful of predictors, suggests that parsimony may play a crucial role in model success. Similarly, the strong results of other algorithms relying on information sets with fewer predictors, such as ARDIs

and boosted-ARDIs as well as the majority of models trained on factor-only set D1F found at the top of both rankings, further emphasize the importance of simplicity as a key ingredient for reliable forecasting performance across horizons and metrics. While this outcome could potentially be attributed to the limited sample size associated with the target variable under consideration; however, further investigation is necessary to validate this hypothesis.

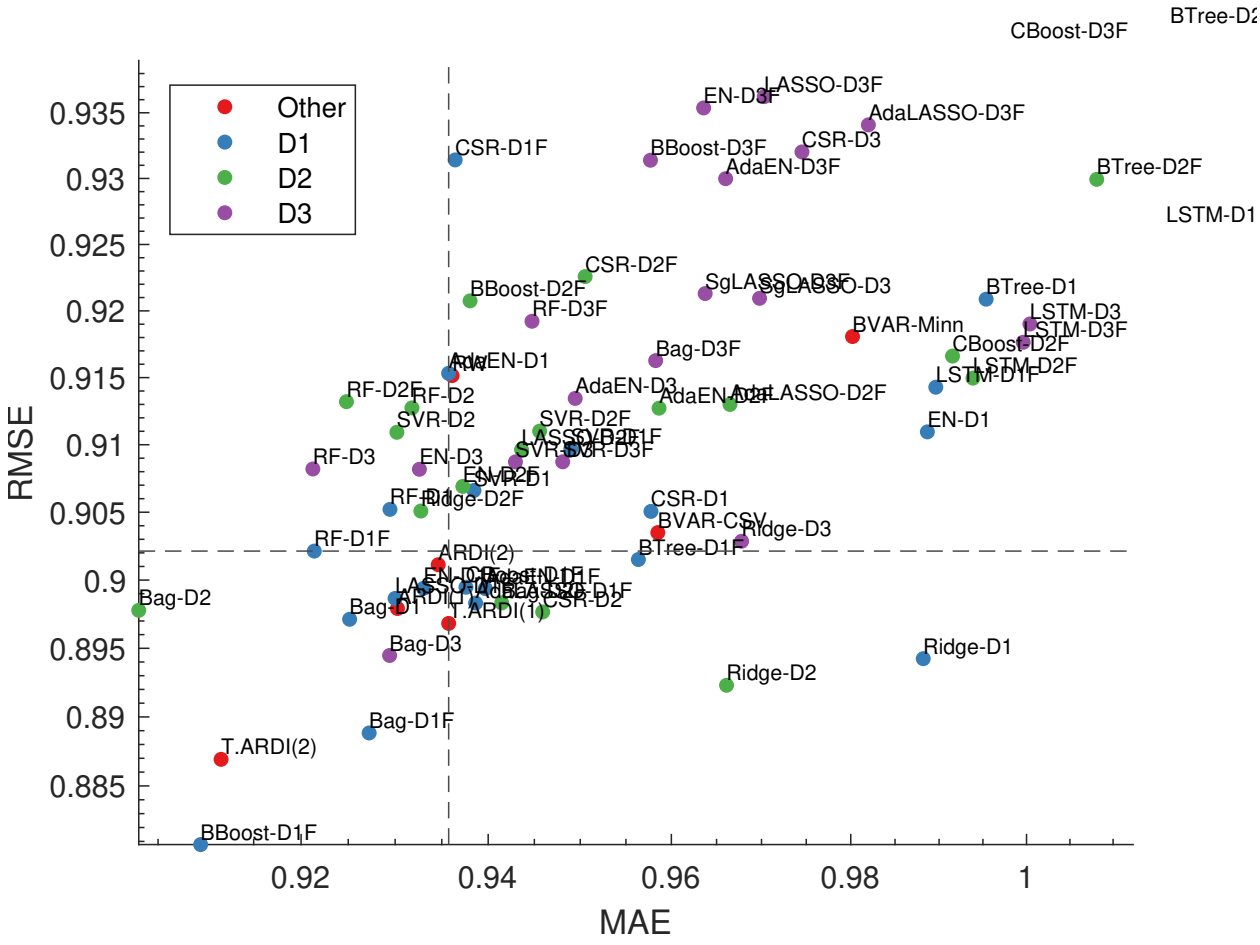


Figure 3: Forecasting performance of all models by information set category: D1(F), D2(F), D3(F), and Other. Evaluation exercise replicating quarterly monitoring using end-of-quarter vintages. The axes show average relative RMSE and MAE achieved by each model over all horizons. Error measures are relative the AR(1) benchmark. The dashed lines represent the boundaries of the upper quartile (enclosing the top 20 models) for each error metric.

### 6.2.2 Real-Time High-Frequency Monitoring

Tables 8 and 9 present the condensed rankings from the horse race for the monthly out-of-sample evaluation experiment, based on the 219 vintages spanning the period from Jan-2003

Table 6: Forecasting Errors: RMSE, 20 Best Performing Models, Mar-2003 to Mar-2021

| Models | n=0 | n=1 | n=2 | n=3 | n=4 | avg | avgMCS |
|---|---|---|---|---|---|---|---|
| BBoost-D1F | **0.60** | **0.88** | 0.93 | 1.00 | 1.00 | **0.88** | 0.72 |
| T.ARDI(2) | 0.61 | 0.91 | 0.93 | 1.00 | 0.99 | 0.89 | 0.64 |
| Bag-D1F | 0.62 | 0.90 | 0.93 | 1.00 | 1.00 | 0.89 | 0.53 |
| Ridge-D2 | 0.62 | 0.91 | 0.93 | 1.00 | 1.00 | 0.89 | 0.42 |
| Ridge-D1 | 0.66 | 0.91 | 0.91 | 1.00 | 1.00 | 0.89 | 0.70 |
| Bag-D3 | 0.62 | 0.92 | 0.93 | 1.01 | 1.00 | 0.89 | 0.50 |
| T.ARDI(1) | 0.60 | 0.96 | 0.93 | 1.00 | 1.00 | 0.90 | 0.59 |
| Bag-D1 | 0.69 | 0.90 | 0.91 | 0.99 | 1.00 | 0.90 | **0.75** |
| CSR-D2 | 0.62 | 0.92 | 0.95 | 1.00 | 1.01 | 0.90 | 0.53 |
| Bag-D2 | 0.66 | 0.92 | 0.92 | 0.99 | 1.00 | 0.90 | 0.53 |
| ARDI(1) | 0.62 | 0.94 | 0.93 | 1.00 | 1.00 | 0.90 | 0.46 |
| AdaLASSO-D1F | 0.64 | 0.91 | 0.94 | 1.00 | 1.00 | 0.90 | 0.48 |
| Bag-D2F | 0.69 | 0.89 | 0.92 | **0.99** | 0.99 | 0.90 | 0.61 |
| LASSO-D1F | 0.65 | 0.91 | 0.94 | 1.00 | 1.00 | 0.90 | 0.43 |
| AdaEN-D1F | 0.65 | 0.91 | 0.94 | 1.00 | 1.00 | 0.90 | 0.43 |
| EN-D1F | 0.65 | 0.91 | 0.94 | 1.00 | 1.00 | 0.90 | 0.43 |
| CBoost-D1F | 0.66 | 0.89 | 0.93 | 1.02 | 1.00 | 0.90 | 0.44 |
| ARDI(2) | 0.64 | 0.94 | 0.93 | 1.00 | 1.00 | 0.90 | 0.50 |
| BTree-D1F | 0.67 | 0.90 | 0.91 | 1.01 | 1.01 | 0.90 | 0.56 |
| RF-D1F | 0.70 | 0.90 | 0.92 | 1.00 | 0.99 | 0.90 | 0.61 |

NOTES: The table reports the relative RMSE for the n-steps ahead prediction, with n=0 reflecting the nowcast. Evaluation exercise conducted as if predictions were calculated every end of the 3rd month of each quarter. The average relative RMSE over all horizons is also given. The last column reports the average p-value of the model confidence set (MCS) test, proposed by Hansen et al. (2011), over all horizons and for squared losses. Highlighted cells denote the models that belong to the 60% MCS (i.e. the set of superior models with equal performance). A small p-value suggests that the row model is unlikely to be a member of the MCS. Error measures are reported relative to the AR(1). The models have been ranked wrt the average relative error measure over the 5 different forecasting horizons considered. Figures in bold show the model with the best statistic, that is, lowest relative error measure, and largest average MCS p-value. Model names ending in $F$ contain only factors on the RHS. The full ranking containing all the premutations of models and transformations, can be found in the appendix.

Table 7: Forecasting Errors: MAE, 20 Best Performing Models, Mar-2003 to Mar-2021

| Models | n=0 | n=1 | n=2 | n=3 | n=4 | avg | avgMCS |
|---|---|---|---|---|---|---|---|
| Bag-D2 | 0.72 | 0.88 | **0.89** | 1.00 | 1.01 | **0.90** | **0.78** |
| BBoost-D1F | 0.78 | **0.81** | 0.93 | 1.01 | 1.01 | 0.91 | 0.59 |
| T.ARDI(2) | 0.78 | 0.87 | 0.91 | 1.01 | 1.00 | 0.91 | 0.70 |
| RF-D3 | 0.79 | 0.90 | 0.92 | 0.99 | 1.00 | 0.92 | 0.62 |
| RF-D1F | 0.80 | 0.89 | 0.92 | 1.00 | 0.99 | 0.92 | 0.59 |
| RF-D2F | 0.82 | 0.90 | 0.93 | 1.00 | **0.98** | 0.92 | 0.51 |
| Bag-D1 | 0.77 | 0.87 | 0.90 | 1.02 | 1.06 | 0.93 | 0.44 |
| Bag-D1F | 0.77 | 0.88 | 0.92 | 1.02 | 1.04 | 0.93 | 0.48 |
| Bag-D3 | **0.72** | 0.89 | 0.94 | 1.05 | 1.04 | 0.93 | 0.45 |
| RF-D1 | 0.78 | 0.90 | 0.93 | 1.02 | 1.02 | 0.93 | 0.30 |
| LASSO-D1F | 0.79 | 0.89 | 0.96 | 1.01 | 1.00 | 0.93 | 0.36 |
| SVR-D2 | 0.84 | 0.90 | 0.92 | 1.00 | 1.00 | 0.93 | 0.56 |
| ARDI(1) | 0.80 | 0.93 | 0.92 | 1.00 | 1.00 | 0.93 | 0.58 |
| RF-D2 | 0.81 | 0.90 | 0.94 | 1.01 | 1.00 | 0.93 | 0.41 |
| EN-D3 | 0.82 | 0.92 | 0.93 | 1.00 | 1.00 | 0.93 | 0.59 |
| Ridge-D2F | 0.80 | 0.86 | 0.91 | 1.03 | 1.07 | 0.93 | 0.37 |
| EN-D1F | 0.80 | 0.89 | 0.97 | 1.01 | 1.00 | 0.93 | 0.38 |
| ARDI(2) | 0.79 | 0.93 | 0.94 | 1.02 | 1.00 | 0.93 | 0.43 |
| AdaEN-D1 | 0.79 | 0.88 | 0.96 | 1.05 | 1.00 | 0.94 | 0.52 |
| T.ARDI(1) | 0.75 | 0.94 | 0.93 | 1.01 | 1.06 | 0.94 | 0.57 |

NOTES: The table reports the relative MAE for the n-steps ahead prediction, with n=0 reflecting the nowcast. Evaluation exercise conducted as if predictions were calculated every end of the 3rd month of each quarter. The average relative MAE over all horizons is also given. The last column reports the average p-value of the model confidence set (MCS) test, proposed by Hansen et al. (2011), over all horizons and for absolute losses. Highlighted cells denote the models that belong to the 60% MCS (i.e. the set of superior models with equal performance). A small p-value suggests that the row model is unlikely to be a member of the MCS. Error measures are reported relative to the AR(1). The models have been ranked wrt the average relative error measure over the 5 different forecasting horizons considered. Figures in bold show the model with the best statistic, that is, lowest relative error measure, and largest average MCS p-value. Model names ending in $F$ contain only factors on the RHS. The full ranking containing all the combinations of models and transformations, can be found in the appendix.

to Mar-2021. The first table reports the results using the RMSE criterion, while the second presents the ranking based on the MAE alternative. Tables 8 and 9 share the same structure as those in the previous subsection but include one fewer column. Specifically, the last column in Tables 6 and 7, which reports the average MCS p-value, is omitted in the tables herein because the series of forecast errors for each of the $n$-quarter-ahead predictions in the monthly OOS experiment pools forecasts is derived from different horizons ($h$-steps ahead), rendering the standard MCS procedure inapplicable.

The key takeaways from these tables are summarized as follows. (1) Whereas in the quarterly out-of-sample evaluation all four ARDI specifications made it to the upper quartile in both the RMSE and MAE rankings, in the monthly evaluation exercise, only a single ARDI specification reaches the quartile of top-performing models. Specifically, the ARDI model estimated using the first two targeted-factors achieves the 8th smallest average 5-horizon RMSE, while the standard ARDI(1) is found in the 6th position when model performance is assessed in terms of the 5-horizon average MAE. This finding suggests that although ARDI remains a strong candidate for nowcasting and short-term forecasting of GDP growth, it nevertheless is less effective at capturing useful within-quarter signals compared to other models. (2) Both rankings reveal a strong presence of linear algorithms, particularly bagging and various regularized regression methods. However, certain algorithms dominate specific rankings, with ridge regressions appearing exclusively among the 20 top performers in the RMSE list, whereas random forests primarily observed in the MAE ranking, with only a few instances in the RMSE list. (3) There is an overall increased representation of D2(F) and D3(F) in the upper quartile, while the presence of top performing specifications trained exclusively on factors is less pronounced, implying that adding individual X's provides additional gains when the forecaster tracks GDP growth earlier in the quarter (as opposed to when only tracking at the end of the quarter, as is the case in the quarterly OOS evaluation). Specifically, in the upper 25% of the RMSE ranking, 12 specifications were trained on D1(F), including T.ARDI(2), 5 on D2(F), and 3 on D3(F), with half of the top-ranked specifications corresponding to factor-only information sets. Similarly, in the upper quartile of the MAE ranking, 9 specifications were trained on D1(F), including ARDI(1), 7 on D2(F), and 3 on D3(F). Moreover, regarding the inclusion of individual series, 10 specifications correspond to factor-only information sets, and 9 to composite sets, with the remaining spot occupied by the RW, which makes it to the upper quartile under the MAE criterion in the monthly OOS evaluation. (4) Examining Figure 4, which combines the two error metrics, reveals that training linear ensemble methods such as bagging and CSR with information sets that incorporate the lagged terms of high frequency variables and/or factors in an unrestricted manner is a highly effective device for tracking GDP growth on a monthly basis, consistently

outperforming other ML candidates and standard benchmarks. This finding, drawn from the figure illustrating the horse race with the combined metrics, reiterates and reinforces the two earlier takeaways observed from analysing the two rankings separately. (5) Finally, the BBoost-D1F specification stands out as a strong performer, particularly for nowcasting GDP. Among the 85 models evaluated, it ranks 3rd in the RMSE ranking for average 5-horizon errors and 2nd for nowcasting. While slightly less dominant in the MAE ranking, it still ranks 5th for nowcasting and 13th for the average 5-horizon MAE, placing it comfortably within the top quartile of models. These results position BBoost-D1F firmly within the top-performing models, highlighting its strong predictive capability across metrics and horizons.

The findings from the horse race in the monthly out-of-sample evaluation align closely with those of the standard quarterly-frequency OOS experiment. This consistency underscores the robustness of the top-performing candidates and confirms that the leading algorithms and information sets are equally effective for tracking GDP at a monthly, and possibly even higher, observational frequencies.

# 7 Conclusion

The findings of this study reveal that ML methods can produce more accurate nowcasts and short-term forecasts of GDP growth rates compared to numerous commonly used benchmarks. In terms of the components that make the successful combination, it is challenging to find unanimity when the set of competing models is extensive and multiple horizons are considered. Examining the full ranking from the comprehensive evaluation of all 85 candidate models provides crucial insights. As evidenced by the horse race results, the $L2$ boosting algorithm with a linear base procedure, estimated on the set of quarterly factors extracted from the full set of predictors and incorporating lags using the block-wise approach of Bai and Ng (2009), consistently ranks among the top performers, both overall and particularly for nowcasting. Repeatedly updating the prediction using the variables that at each iteration best fit the remaining errors (updated residuals), seems to provide a promising route for approximating the conditional expectation of the target. Overall, the upper quartile of the models' performance distribution, as captured by the RMSE, is marked by a significant presence of linear ML methodologies. Out of the 20 best performers, 16 specifications involve some form of a linear regression model: factor-augmented ARs, Ridge Regressions, Bagged Linear Regressions, CSR, LASSO, EN and the LASSO/EN adaptive variants. When ranking

Table 8: Forecasting Errors: RMSE, 20 Best Performing Models, *Monthly OOS Evaluation* over Jan-2003 to Mar-2021

| Models | n=0 | n=1 | n=2 | n=3 | n=4 | avg |
|---|---|---|---|---|---|---|
| Bag-D3 | 0.73 | 0.93 | 0.97 | 1.00 | 0.99 | **0.92** |
| CSR-D2 | **0.71** | 0.94 | 0.98 | 1.00 | 1.00 | 0.93 |
| BBoost-D1F | 0.72 | 0.92 | 0.98 | 1.01 | 1.00 | 0.93 |
| Bag-D2F | 0.77 | 0.91 | 0.97 | 0.99 | 1.00 | 0.93 |
| Ridge-D1 | 0.76 | 0.92 | 0.96 | 1.00 | 1.00 | 0.93 |
| Ridge-D2 | 0.74 | 0.93 | 0.97 | 1.00 | 1.00 | 0.93 |
| Bag-D2 | 0.77 | 0.92 | 0.97 | 0.99 | 1.00 | 0.93 |
| T.ARDI(2) | 0.75 | 0.93 | 0.98 | 1.00 | 1.01 | 0.93 |
| Bag-D1F | 0.76 | 0.93 | 0.97 | 1.00 | 1.00 | 0.93 |
| Bag-D1 | 0.78 | 0.92 | 0.96 | 1.00 | 1.00 | 0.93 |
| CSR-D1 | 0.76 | 0.93 | 0.97 | 1.00 | 1.01 | 0.94 |
| Ridge-D3 | 0.79 | 0.92 | 0.97 | 1.00 | 1.00 | 0.94 |
| RF-D1F | 0.82 | 0.92 | 0.97 | 1.00 | **0.99** | 0.94 |
| Ridge-D2F | 0.81 | 0.91 | 0.97 | 1.00 | 1.00 | 0.94 |
| CBoost-D1F | 0.77 | 0.93 | 0.98 | 1.01 | 1.00 | 0.94 |
| SVR-D1 | 0.82 | 0.92 | 0.96 | 0.99 | 1.00 | 0.94 |
| LASSO-D1F | 0.79 | 0.93 | 0.98 | 1.00 | 1.00 | 0.94 |
| AdaLASSO-D1F | 0.78 | 0.94 | 0.98 | 1.00 | 1.00 | 0.94 |
| RF-D1 | 0.80 | 0.92 | 0.97 | 1.00 | 1.00 | 0.94 |
| Bag-D3F | 0.82 | 0.92 | 0.97 | 1.00 | 0.99 | 0.94 |

NOTES: The table reports the relative RMSE for the n-steps ahead prediction, with $n = 0$ reflecting the nowcast. Evaluation exercise conducted in a *real-time* fashion, updating the predictions at the end of every month. The average relative RMSE over all horizons is also given. Since in the real-time POOS evaluation each of the forecasts comes from different h-steps ahead predictions (i.e. for Jan & Feb the nowcast, $n = 0$, comes from setting $h = 2$, while in Mar from $h = 1$), I abstract from conducting MCS testing, as it is primarily suited for single-horizon forecast errors. Error measures are reported relative to the AR(1). The models have been ranked wrt the average relative error measure over the 5 different forecasting horizons considered. Figures in bold show the model with the lowest relative error measure. Model names ending in $F$ contain only factors on the RHS. The full ranking containing all the combinations of models and transformations, can be found in the appendix.

Table 9: Forecasting Errors: MAE, 20 Best Performing Models, *Monthly OOS Evaluation* over Jan-2003 to Mar-2021

| Models | n=0 | n=1 | n=2 | n=3 | n=4 | avg |
|---|---|---|---|---|---|---|
| Bag-D2 | 0.77 | 0.89 | 0.96 | 1.01 | 1.04 | **0.94** |
| RF-D1F | 0.86 | 0.91 | 0.97 | **0.99** | 1.00 | 0.95 |
| RF-D3 | 0.84 | 0.93 | **0.96** | 0.99 | 1.02 | 0.95 |
| Bag-D3 | **0.77** | 0.91 | 0.99 | 1.03 | 1.03 | 0.95 |
| RF-D2F | 0.87 | 0.93 | 0.97 | 1.00 | 0.99 | 0.95 |
| ARDI(1) | 0.86 | 0.92 | 0.97 | 0.99 | 1.00 | 0.95 |
| SVR-D2 | 0.87 | 0.91 | 0.97 | 1.00 | 1.01 | 0.95 |
| Bag-D1 | 0.81 | 0.90 | 0.97 | 1.03 | 1.05 | 0.95 |
| RF-D1 | 0.84 | 0.92 | 0.97 | 1.01 | 1.02 | 0.95 |
| RW | 0.89 | 0.92 | 0.97 | 0.99 | 1.00 | 0.95 |
| RF-D2 | 0.85 | 0.92 | 0.97 | 1.01 | 1.01 | 0.95 |
| EN-D2F | 0.88 | 0.93 | 0.97 | 1.00 | 1.00 | 0.95 |
| BBoost-D1F | 0.81 | 0.92 | 0.99 | 1.04 | 1.02 | 0.96 |
| BBoost-D2F | 0.93 | 0.90 | 0.96 | 1.00 | 1.01 | 0.96 |
| EN-D3 | 0.90 | 0.92 | 0.97 | 1.00 | 1.00 | 0.96 |
| CSR-D1F | 0.88 | 0.91 | 0.97 | 1.01 | 1.02 | 0.96 |
| SVR-D1 | 0.88 | 0.92 | 0.97 | 1.01 | 1.02 | 0.96 |
| LASSO-D2F | 0.91 | 0.93 | 0.97 | 1.00 | 0.99 | 0.96 |
| LASSO-D1F | 0.85 | 0.93 | 1.00 | 1.02 | 1.00 | 0.96 |
| EN-D1F | 0.85 | 0.93 | 0.99 | 1.01 | 1.01 | 0.96 |

NOTES: The table reports the relative MAE for the n-steps ahead prediction, with n=0 reflecting the nowcast. Evaluation exercise conducted in a *real-time* fashion, updating the predictions at the end of every month. The average relative MAE over all horizons is also given. Since in the real-time POOS evaluation each of the forecasts comes from different h-steps ahead predictions (i.e. for Jan & Feb the nowcast, $n = 0$, comes from setting $h = 2$, while in Mar from $h = 1$), I abstract from conducting MCS testing, as it is primarily suited for single-horizon forecast errors. Error measures are reported relative to the AR(1). The models have been ranked wrt the average relative error measure over the 5 different forecasting horizons considered. Figures in bold show the model with the lowest relative error measure. Model names ending in *F* contain only factors on the RHS. The full ranking containing all the combinations of models and transformations, can be found in the appendix.

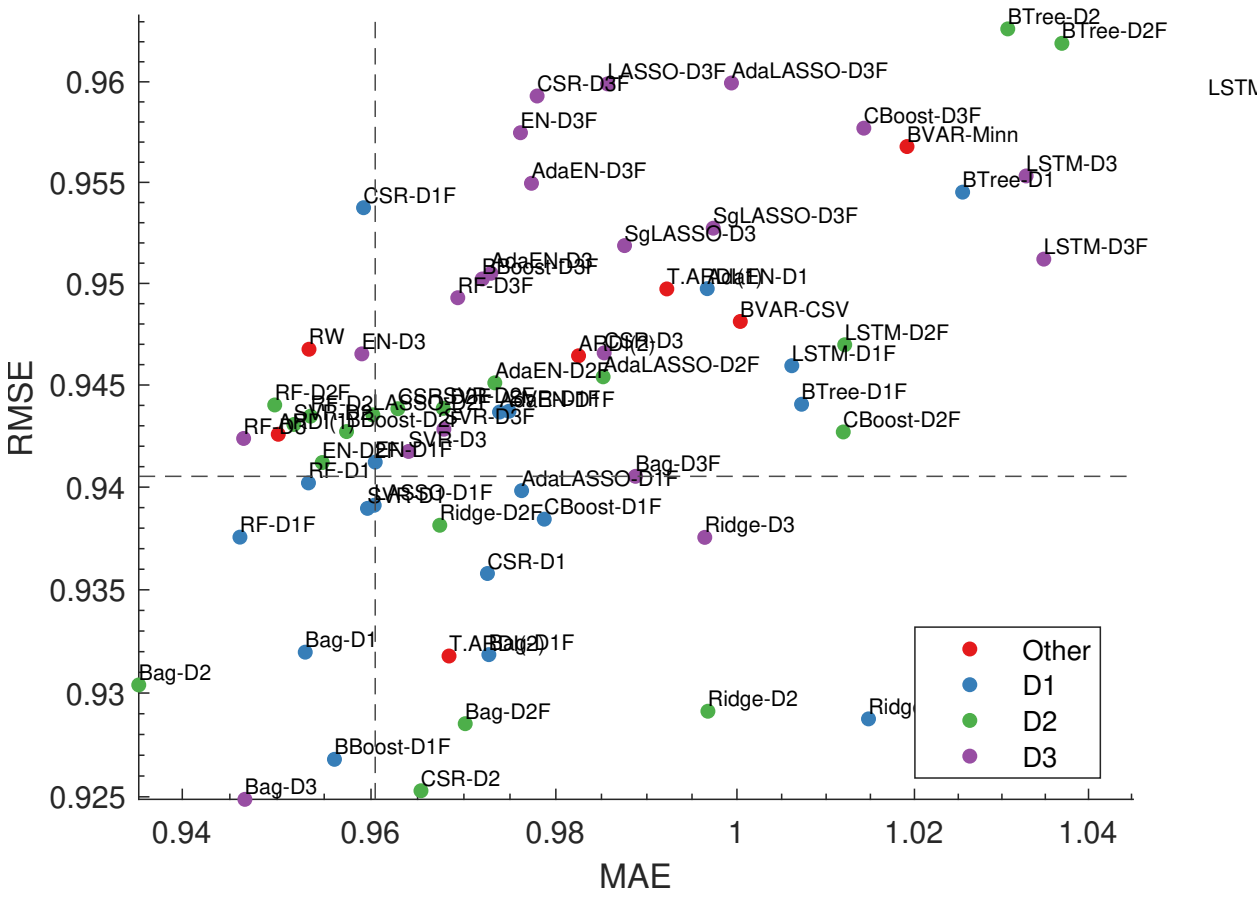


Figure 4: Forecasting performance of all models by information set category: D1(F), D2(F), D3(F), and Other. Evaluation exercise replicating monthly monitoring using end-of-month vintages. The axes show average relative RMSE and MAE achieved by each model over all horizons. Error measures are relative to the AR(1) benchmark. The dashed lines represent the boundaries of the upper quartile (enclosing the top 20 models) for each error metric.

model performance with the MAE alternative, which is less sensitive to large forecast errors, a similar picture emerges. However, the number of nonlinear models making it to the top quartile increases considerably, with the majority of the new entries corresponding to some specification that involves random forests. Among the linear specifications, a machine learning approach that warrants particular attention is the bagging ensemble with a linear base learner which consistently ranks in the top quartile of the model performance distribution for both RMSE and MAE, demonstrating strong performance regardless of the information set being used. Regarding the competing information sets, the results of the horse race identify the quarterly-factor information set (D1F) as a particularly effective companion set for numerous algorithms, as it appears frequently among the top-ranked specifications. Similar conclusions emerge from the horse race based on the monthly out-of-sample evaluation experiment, reinforcing the robustness of the top candidates identified in the standard quarterly-frequency OOS evaluation, and verifying that the prevailing algorithms and information sets are also suitable for tracking GDP at high observation frequency. However, one notable distinction between the two OOS experiments is that when GDP is assumed to be monitored on a monthly basis, there is an increased representation of information sets containing high-frequency panels (D2 and D3) in the upper quartile of both rankings, which implies that using high-frequency predictors helps capture useful within-quarter signals early in the quarter. Among the candidate models that deserve special attention, given its robust success (both across horizons and error measures) as well as its fast and easy implementation, is the diffusion-index of Stock and Watson (2002) estimated with the target-relevant factor modification of Bai and Ng (2008). Specifically, the target-ARDI containing the first 2 principal components extracted from the quarterly panel and using the BIC for selecting the optimal lag-orders, is consistently found among the top performers. Nevertheless, it is never found to have the best performance, neither when considering the 5-horizon average performance, nor the individual horizons' performances. Furthermore, these results suggest that doing preselection ahead of factor extraction is almost clearly the preferred choice when forming diffusion indices for the purposes of nowcasting and forecasting using standard factor-augmented autoregressions. This finding might be suggestive of a possible direction towards a fruitful modification of the least-squares factor-boosting algorithm that could further increase its already strong performance.

Regarding further directions for future work, past studies have demonstrated that generating forecasts at lower levels of aggregation, by setting the target variable to be each of the demand- or supply-side components of GDP and subsequently aggregating the forecasts for these subcomponents, can yield promising results.[26] For instance, Foroni and Marcellino

[26]GDP can be decomposed into subcomponents derived either from the supply- or the demand-side.

(2014) using a large unbalanced mixed-frequency dataset for the Euro area, show that there can be significant accuracy gains from separately nowcasting each supply-side GDP component using factor-augmented MIDAS (F-MIDAS) regressions and then aggregating the predictions, compared to directly using GDP growth as the dependent variable. Depending on the month within the quarter the nowcast is generated, they report MSE reductions of up to 11.7% in predicting EA GDP growth rates. Building on these findings, future studies can explore predicting GDP indirectly by aggregating forecasts derived from modeling the individual production or expenditure side components using ML methodologies.

This study contributes to the expanding field of developing machine learning models aimed at enhancing macroeconomic monitoring. The insights gained from the analysis of the components and the algorithms that constitute the successful candidates, provide directions towards building improved ML-based models for nowcasting and forecasting macroeconomic indicators in data-rich environments, taking into account the plethora of indicators that are available at different sampling frequencies.

# References


Abu-Mostafa, Y. S., Magdon-Ismail, M., and Lin, H.-T. (2012). *Learning from data*, volume 4. AMLBook New York, NY, USA:. [35]

Almon, S. (1965). The distributed lag between capital appropriations and expenditures. *Econometrica: Journal of the Econometric Society*, pages 178–196. [7]

Andreou, E., Ghysels, E., and Kourtellos, A. (2010). Regression models with mixed sampling frequencies. *Journal of Econometrics*, 158(2):246–261. [29]

Babii, A., Ghysels, E., and Striaukas, J. (2022). Machine Learning Time Series Regressions with an Application to Nowcasting. *Journal of Business & Economic Statistics*, 40(3):1094–1106. [5, 7, 10, 15, 27, 30, 31, 36, 60]

Bai, J. and Ng, S. (2008). Forecasting economic time series using targeted predictors. *Journal of Econometrics*, 146(2):304–317. [11, 15, 17, 20, 36, 52]

Bai, J. and Ng, S. (2009). Boosting diffusion indices. *Journal of Applied Econometrics*, 24(4):607–629. [4, 9, 10, 15, 22, 23, 36, 48]

Ballarin, G., Dellaportas, P., Grigoryeva, L., Hirt, M., van Huellen, S., and Ortega, J.-P. (2024). Reservoir computing for macroeconomic forecasting with mixed-frequency data. *International Journal of Forecasting*, 40(3):1206–1237. [6]

Bańbura, M., Giannone, D., Modugno, M., and Reichlin, L. (2013). Now-casting and the real-time data flow. In *Handbook of economic forecasting*, volume 2, pages 195–237. Elsevier. [11]


Disaggregating GDP from the production side involves breaking down output by sectors or industries (e.g., agriculture, manufacturing, services etc.) using gross value added (GVA) data, where GDP is calculated as the sum of industry- or sector-specific GVAs plus taxes minus subsidies. From the expenditure side, the total GDP is divided into consumption, investment, government spending, and net exports.

Bańbura, M., Giannone, D., and Reichlin, L. (2010). Large bayesian vector auto regressions. *Journal of applied Econometrics*, 25(1):71–92. [15]

Berman, J. J. (2013). *Principles of big data: Preparing, Sharing, and Analyzing Complex Information*. Morgan Kaufmann. [8]

Beyhum, J. and Striaukas, J. (2023). Sparse plus dense midas regressions and nowcasting during the covid pandemic. *arXiv e-prints*, pages arXiv–2306. [9]

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140. [20]

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32. [15, 23]

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and Regression Trees. *Wadsworth Statistics/Probability Series. Wadsworth Advanced Books and Software.* [23]

Buehlmann, P. (2006). Boosting for high-dimensional linear models. *The Annals of Statistics*, 34(2):559–583. [15]

Bühlmann, P. and Yu, B. (2003). Boosting with the l 2 loss: regression and classification. *Journal of the American Statistical Association*, 98(462):324–339. [21, 22, 23]

Carriero, A., Clark, T. E., and Marcellino, M. (2016). Common Drifting Volatility in Large Bayesian VARs. *Journal of Business & Economic Statistics*, 34(3):375–390. [15, 16, 36]

Carriero, A., Clark, T. E., and Marcellino, M. (2019a). Large bayesian vector autoregressions with stochastic volatility and non-conjugate priors. *Journal of Econometrics*, 212(1):137–154. [5]

Carriero, A., Galvao, A. B., and Kapetanios, G. (2019b). A comprehensive evaluation of macroeconomic forecasting methods. *International Journal of Forecasting*, 35(4):1226–1239. [5, 7]

Chan, J. C. (2020). Large Bayesian VARs: A Flexible Kronecker Error Covariance Structure. *Journal of Business & Economic Statistics*, 38(1):68–79. [36]

Chernozhukov, V., Hansen, C., and Liao, Y. (2017). A lava attack on the recovery of sums of dense and sparse signals. *The Annals of Statistics*, 45(1):39 – 76. [9, 10]

Chinn, M. D., Meunier, B., and Stumpner, S. (2023). Nowcasting world trade with machine learning: a three-step approach. Technical report, National Bureau of Economic Research. [9]

Chow, G. C. and Lin, A.-l. (1971). Best linear unbiased interpolation, distribution, and extrapolation of time series by related series. *The review of Economics and Statistics*, pages 372–375. [29]

Cimadomo, J., Giannone, D., Lenza, M., Monti, F., and Sokol, A. (2021). Nowcasting with large bayesian vector autoregressions. *Journal of Econometrics*. [8]

Clark, T. E., Leonard, S., Marcellino, M., and Wegmüller, P. (2022). Weekly nowcasting us inflation with enhanced random forests. Technical report, Mimeo. [5]

D'Agostino, A., Giannone, D., and Surico, P. (2007). (un) predictability and macroeconomic stability. *CEPR Discussion Papers*, 6594. [36]

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., and Vapnik, V. (1996). Support vector regression machines. *Advances in neural information processing systems*, 9. [15, 25]

Efron, B. and Hastie, T. (2021). *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*, volume 6. Cambridge University Press. [24]

Elliott, G., Gargano, A., and Timmermann, A. (2013). Complete subset regressions. *Journal of Econometrics*, 177(2):357–373. Dynamic Econometric Modeling and Forecasting. [15,

19]

Elliott, G., Gargano, A., and Timmermann, A. (2015). Complete subset regressions with large-dimensional sets of predictors. *Journal of Economic Dynamics and Control*, 54:86–110. [19]

European Central Bank (2022). What explains recent errors in the inflation projections of eurosystem and ecb staff? *Economic Bulletin*, Issue 3. [2]

Fan, J., Masini, R. P., and Medeiros, M. C. (2023). Bridging factor and sparse models. *The Annals of Statistics*, 51(4):1692–1717. [9]

Foroni, C. and Marcellino, M. (2014). A comparison of mixed frequency approaches for nowcasting euro area macroeconomic aggregates. *International Journal of Forecasting*, 30(3):554–568. [52]

Foroni, C., Marcellino, M., and Schumacher, C. (2015). Unrestricted mixed data sampling (midas): Midas regressions with unrestricted lag polynomials. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 178(1):57–82. [30]

Fortin-Gagnon, O., Leroux, M., Stevanovic, D., and Surprenant, S. (2022). A large canadian database for macroeconomic analysis. *Canadian Journal of Economics/Revue canadienne d'économique*, 55(4):1799–1833. [5]

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232. [15, 21]

Ghysels, E., Santa-Clara, P., and Valkanov, R. (2005). There is a risk-return trade-off after all. *Journal of financial economics*, 76(3):509–548. [29]

Ghysels, E., Santa-Clara, P., and Valkanov, R. (2006). Predicting volatility: Getting the most out of return data sampled at different frequencies. *Journal of Econometrics*, 131(1-2):59–95. [29]

Ghysels, E., Sinko, A., and Valkanov, R. (2007). Midas regressions: Further results and new directions. *Econometric reviews*, 26(1):53–90. [29]

Giannone, D., Lenza, M., and Primiceri, G. E. (2021). Economic predictions with big data: The illusion of sparsity. *Econometrica*, 89(5):2409–2437. [5, 9, 37]

Goulet Coulombe, P., Leroux, M., Stevanovic, D., and Surprenant, S. (2021a). Macroeconomic data transformations matter. *International Journal of Forecasting*, 37(4):1338–1354. [3, 7, 9]

Goulet Coulombe, P., Leroux, M., Stevanovic, D., and Surprenant, S. (2022). How is machine learning useful for macroeconomic forecasting? *Journal of Applied Econometrics*, 37(5):920–964. [3, 5, 7, 9]

Goulet Coulombe, P., Marcellino, M., and Stevanović, D. (2021b). Can Machine Learning Catch the COVID-19 Recession? *National Institute Economic Review*, 256:71–109. [5]

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610. [26]

Groen, J. J. and Kapetanios, G. (2016). Revisiting useful approaches to data-rich macroeconomic forecasting. *Computational Statistics & Data Analysis*, 100:221–239. [4]

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273. [3]

Hansen, P. R. (2005). A test for superior predictive ability. *Journal of Business & Economic Statistics*, 23(4):365–380. [35]

Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The Model Confidence Set. *Econometrica*,

79(2):453–497. [35, 39, 40, 45, 46]

Hepenstrick, C. and Marcellino, M. (2019). Forecasting gross domestic product growth with large unbalanced data sets: the mixed frequency three-pass regression filter. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 182(1):69–99. [5]

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. [15, 26]

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67. [15, 18]

Inoue, A. and Kilian, L. (2008). How useful is bagging in forecasting economic time series? a case study of u.s. consumer price inflation. *Journal of the American Statistical Association*, 103(482):511–522. [4, 15, 20]

Kelly, B. T. and Pruitt, S. (2015). The three-pass regression filter: A new approach to forecasting with many predictors. *Journal of Econometrics*, 186(2):294–316. [5]

Koenig, E. F., Dolmas, S., and Piger, J. (2003). The use and abuse of real-time data in economic forecasting. *Review of Economics and Statistics*, 85(3):618–628. [30]

Kotchoni, R., Leroux, M., and Stevanovic, D. (2019). Macroeconomic forecast accuracy in a data-rich environment. *Journal of Applied Econometrics*, 34(7):1050–1072. [3, 5, 20]

Lütkepohl, H. (1981). A model for non-negative and non-positive distributed lag functions. *Journal of Econometrics*, 16(2):211–219. [29]

Marcellino, M. (1999). Some consequences of temporal aggregation in empirical analysis. *Journal of Business & Economic Statistics*, 17(1):129–136. [29]

McCracken, M. and Ng, S. (2020). FRED-QD: A Quarterly Database for Macroeconomic Research. *National Bureau of Economic Research*. [5, 31]

McCracken, M. W. and Ng, S. (2016). FRED-MD: A Monthly Database for Macroeconomic Research. *Journal of Business & Economic Statistics*, 34(4):574–589. [5, 31]

Medeiros, M. C., Vasconcelos, G. F. R., Álvaro Veiga, and Zilberman, E. (2021). Forecasting inflation in a data-rich environment: The benefits of machine learning methods. *Journal of Business & Economic Statistics*, 39(1):98–119. [3, 5, 9, 20, 21]

Ng, S. (2013). Variable selection in predictive regressions. *Handbook of economic forecasting*, 2:752–789. [9]

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, 5:197–227. [21]

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245. [5, 27]

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14:199–222. [26]

Stock, J. H. and Watson, M. W. (2002). Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, 20(2):147–162. [4, 11, 15, 16, 17, 52]

Stock, J. H. and Watson, M. W. (2005). Implications of dynamic factor models for VAR analysis. *NBER Working Paper 11467*. [31]

Stock, J. H. and Watson, M. W. (2012). Disentangling the Channels of the 2007-2009 Recession. *National Bureau of Economic Research*. [31]

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288. [15, 18]

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer. [25]

White, H. (2000). A reality check for data snooping. *Econometrica*, 68(5):1097–1126. [35]

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67. [27]

Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101(476):1418–1429. [15, 18]

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320. [15, 18]

Zou, H. and Zhang, H. H. (2009). On the adaptive elastic-net with a diverging number of parameters. *Annals of statistics*, 37(4):1733. [19]

# A Legendre Polynomials

To illustrate the flexibility of Legendre polynomials in approximating different lag structures, Figure 5 shows the various sets of weights generated by the four weight functions resulting when the polynomial degree is set to 3. Assuming we are aggregating the first 12 lags of a monthly covariate, each set of weights is composed of 12 values, one for each lagged term. The horizontal axis represents the 12 lags, with the leftmost corresponding to the most recent, while the vertical axis indicates the weight assigned to each lag, under polynomials of different orders. As shown in the figure, employing a dictionary composed of Legendre polynomials allows for a variety of weighting schemes, including linear, downward sloping, and hump-shaped forms, among others. Note that, although the weights for the order-1 polynomial appear to imply an amplifying (rather than a decaying) effect over time of the predictor on the target variable, since the generated sequence of weights is antisymmetric (i.e., each pair of weights on either side of the central point being equal in magnitude but opposite in sign), it enables the linear weighting scheme to capture equally both possibilities of either an increasing or a diminishing influence of the predictor on the target variable over time. Finally, it is worth noting that the first Legendre polynomial, corresponding to degree zero, represents the constant function, which implies a uniform weighting scheme, effectively averaging the lags using flat aggregation, similar to the approach proposed for constructing information set D1. However, a key difference between the two methods is in the observations included in the averaging process. Whereas in D1 we aggregate the three monthly observations within each quarter, in D3 each observation aggregates a full year of lags along with any available leads. For predictors with no leading observations, this method yields quarterly observations that correspond to the annual (rather than the quarterly) average.

# B Temporal Stability and Sparsity in the Target-Predictors Relationship

To get a sense of the relevance of the different predictors across time, and to gain insight into the degree of sparsity between different models and under alternative transformations, this section presents the sparsity pattern plots for two selected algorithms—hard-thresholding and sparse-group LASSO—across the three information sets. The selection of these two algorithms is motivated by the following reasons. First, models incorporating a pre-screening step to define targeted-predictors, consistently rank among the top quartile of best-performing models. Thus, the underlying dynamics driving this outcome merit further attention. Second, the sparse-group LASSO's enhanced variable selection capabilities, that are due to its capacity to recognize that covariates at different lags are temporally related, which is achieved by grouping lags and inducing sparsity both between and within the defined groups, make it an appropriate reference model for comparison. The plots are based on the 220 vintages from the out-of-sample evaluation conducted at the monthly periodicity, and present the variables selected by the models to produce the nowcasts ($n = 0$). Recall that, depending on the specific monthly vintage used, the nowcasts correspond to either $h = 2$ or $h = 1$. Legendre-aggregated covariates from different polynomial orders (in D3) and lags of different

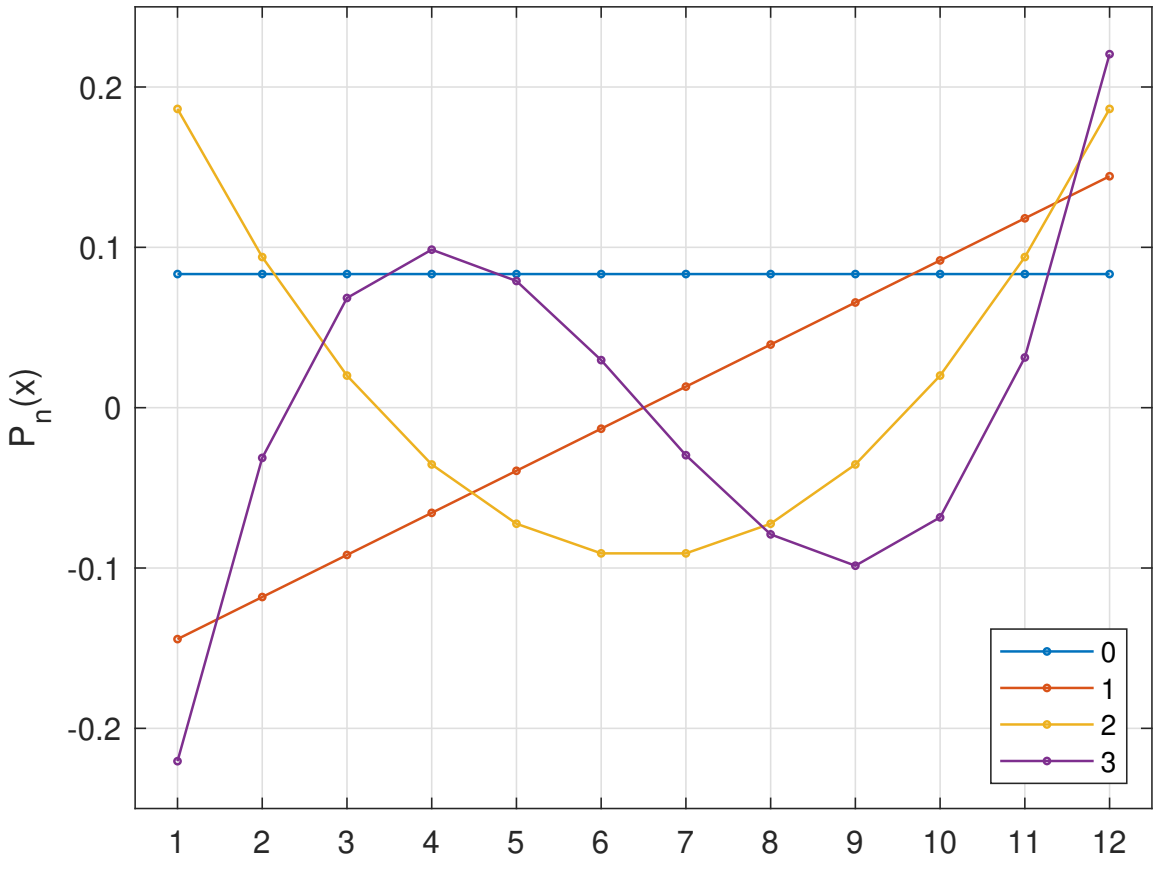


Figure 5: Legendre Polynomials for 12 Lags and Degrees 0 to 3

variables (in D1 and D2) have been aggregated across predictors, so that each entry on the horizontal axis in this section's figures, represents one of the 257 variables that comprise the three information sets, including one entry for each PCA-factor and one for all autoregressive lags. To see how the various economic categories are represented over time, indicators are further grouped together using the categories defined in Section 4.

Figures 6 to 8 present the covariates selected by applying the hard-thresholding algorithm on information sets D1, D2 and D3, while 9 shows the features selected by the sg-LASSO-MIDAS model, which is only trained using the information set based on the Legendre polynomials, as per the recommendation of Babii et al. (2022). Hard-thresholding is used in this study to perform the preselection step in three instances: (1) in the Target CSR model, (2) in the Linear Bagging model, and (3) to select the set of predictors used to construct the factors for the Target ARDI model. However, the specific screening settings vary across each of these models. As such, the sparsity patters plotted here are obtained by setting the absolute t-statistic threshold at 1.96, which roughly corresponds to the critical value at the 5% level of significance, against a two-sided alternative that the variable under investigation is significant.

Note that the sparsity pattern plot for the variable selection conducted on D1 contains 8 entries in the 'Factors' category, whereas plots for D2 and D3 contain twice as many entries, as two sets of factors are created for the mixed-frequency information sets—one from the monthly and one from the quarterly panel. Specifically, the first eight Factor-related columns (in brown) correspond to the monthly factors, while the subsequent eight represent the components extracted from the quarterly panel. Furthermore, plots for the hard-thresholding algorithm do not contain an entry for autoregressive lags, because these are added as control variables and are therefore always included among the (potential) predictors.[27]

A visual inspection of the four figures indicates considerable variability in the persistence of individual predictors' relevance across vintages, regardless of the preselection algorithm or the particular information set over which preselection is applied. This temporal instability in the sparsity patterns highlights the evolving nature of target-predictor relationships and is suggestive of the importance of incorporating alternative types of thresholding preselection strategies in data-rich settings, particularly when employing forecasting methods that are not inherently sparsity-inducing. As such, introducing a preliminary step that identifies target-relevant predictors as part of the forecasting framework can enhance predictive accuracy by increasing robustness to structural changes and enabling models to adapt to evolving economic conditions. Examining the sparsity patterns obtained by applying hard-thresholding preselection on the three information sets (Figures 6-8), the quarterly aggregated information set D1 appears to be the sparsest, followed by D3, in which each predictor is temporally aggregated using a variety of weighting schemes obtained by Legendre polynomials of different orders. A few interesting observations arise from the direct comparison of the patterns depicted in Figures 6 and 8. Since all three information sets are comprised of the same 257 predictors, the only difference between the covariates involved in the preselection exercises presented in the two figures lies in the underlying weighting schemes used to aggregate lagged values of each predictor (along with the number of lags involved in the aggregation) There-

[27] As indicated in the previous sections, autoregressive lags enter as control variables both in the first-stage regressions of the screening step, and in the underlying base learners of the CSR and Bagging ensembles.

fore, the differences in the temporal stability of the indicators observed across the two figures highlight the fact that different a priori choices for the weights used to aggregate the lags, can substantially affect the underlying target-predictor correlations. Interestingly, certain groups of indicators, most notably those related to 'Labor', 'Prices', and 'Interest rates', exhibit markedly sparser patterns when looking at the selected predictors from D1 compared to D3. Since the majority of series in these three categories are sampled at a higher frequency, this observation might be indicative of the fact that alternative dynamic structures (e.g., decaying, hump-shaped, or oscillatory) captured by one or multiple of the various lag-weighting functions included in the Legendre dictionary, may provide a more suitable approximation of the target-predictor relationship, as opposed to the flat distributed-lag structure imposed by the uniform weighting scheme used in D1 to temporally aggregate monthly predictors (Figure 6). Another noteworthy observation across the first three figures, that all depict sparsity patterns under hard-thresholding, and is most noticeable in information set D1, is that the series in the 'Interest rates' category become considerably sparser beginning in late 2014. The observed insignificance in the bulk of interest rate and spread series after 2014 coincides chronologically with the end of the zero-interest rate policy followed by the Federal Reserve and other central banks beginning in 2010 in response to the 2007-2009 Global Financial Crisis. As the U.S. economy began to exit the zero lower bound in 2015, long-term correlations with GDP growth appear to have weakened, giving rise to the sparsity patterns observed in the figure. Nevertheless, similarly to the argument made earlier, comparing the patterns obtained by applying hard-thresholding on D1 and D3, the relatively denser post-2014 patterns observed for interest rates in Figure 8 suggest that it is possible that correlations could still be present, but that a shift in the underlying dynamic relationships could have occurred, which could be better captured by one or multiple alternative lag structures approximated through the different weighting functions included in the Legendre dictionary.[28] Finally, overall, the hard-thresholding algorithm, even when applied on information set D1, yields considerably less sparse selections than those implied by the feature-selection mechanism of the sg-LASSO, which becomes apparent by observing Figure 9. The latter not only selects fewer variables at each vintage, but also the number of series across all vintages that are never selected, is significantly larger. This difference can be attributed to the fact that the variable selection mechanism of the sg-LASSO takes into account the correlations between predictors which is something lacking from hard-thresholding techniques, and is increasingly important given the highly correlated nature of macroeconomic data. Another important

[28] As mentioned in the beginning of this Appendix, in Figures 8 and 9 when the underlying variable-(pre)selection algorithm selects any of the alternative Legendre-aggregated versions corresponding to the different polynomial orders (in D3), the graphs mark one entry on the horizonal axis for that particular covariate. Therefore, the current configuration of the graphs does not allow us to see the particular weighting schemes that were relevant for each predictor and any given vintage. An interesting extension of the analysis presented in this Appendix would be to produce a disaggregated version of Figure 8 showing sparsity pattern for information set D3 based on hard-thresholding, but instead of aggregating the selection of the alternative Legendre-aggregated versions of each covariate, presenting 4 disaggregated graphs, one for each weight function resulting when the polynomial degree in the Legendre dictionary is set to 3. Such an analysis would produce insights on the relevance of alternative temporal-aggregation schemes for different categories of predictors and would allow us to investigate whether the shape of the temporal transmission mechanism between the target and different predictors has shifted across time. I reserve this more detailed investigation for future work.

consideration when evaluating target-predictor relationships via hard-thresholding preselection is that, while the graphs illustrate the relevance of each potential predictor for the target variable, they are not informative of the underlying relative importance of these predictors, as they only depict the frequency with which the algorithm select each variable, without capturing the magnitude of the associated coefficients.

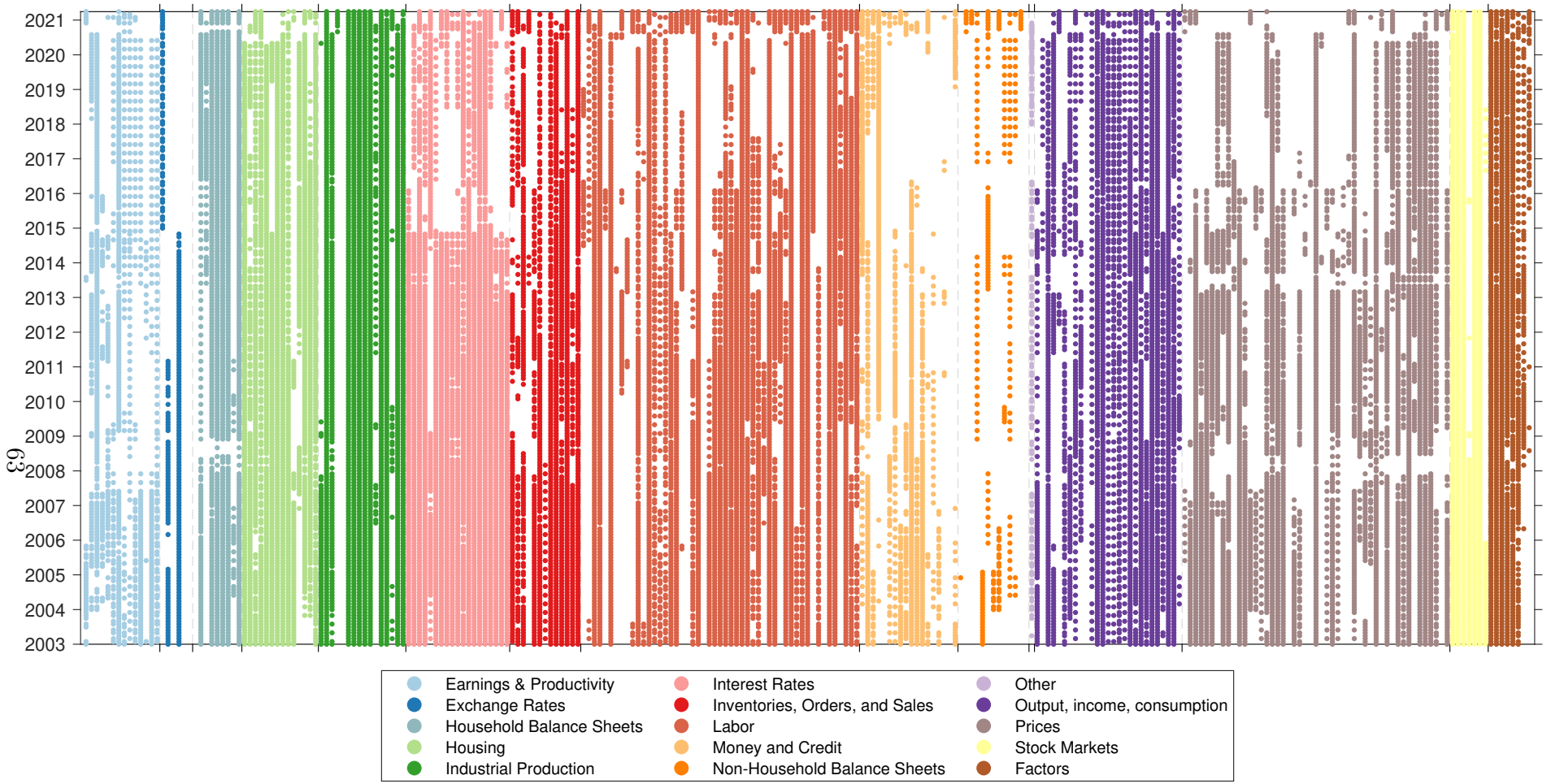


Figure 6: Sparsity patterns for hard-thresholding on D1

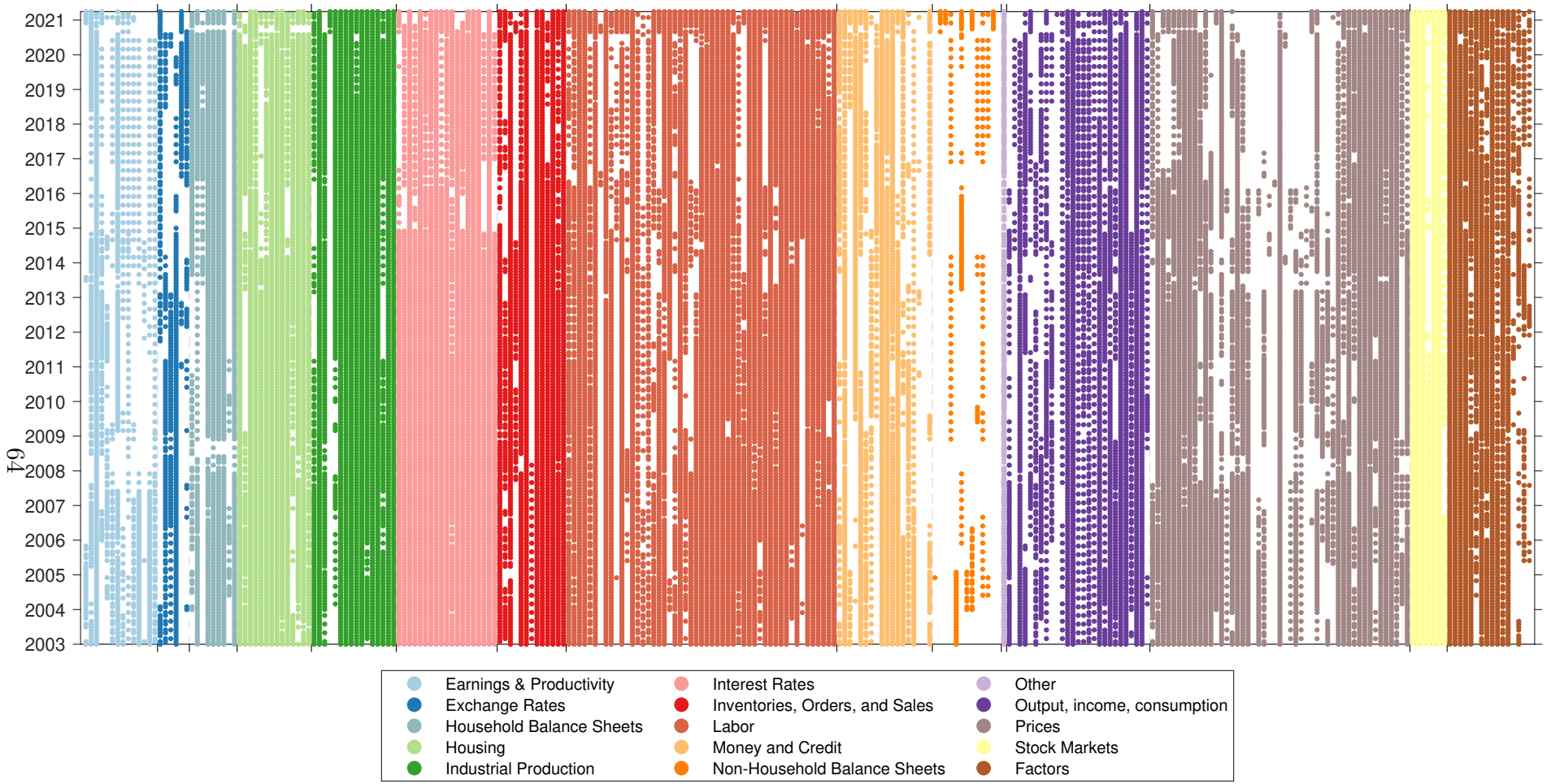


Figure 7: Sparsity patterns for hard-thresholding on D2

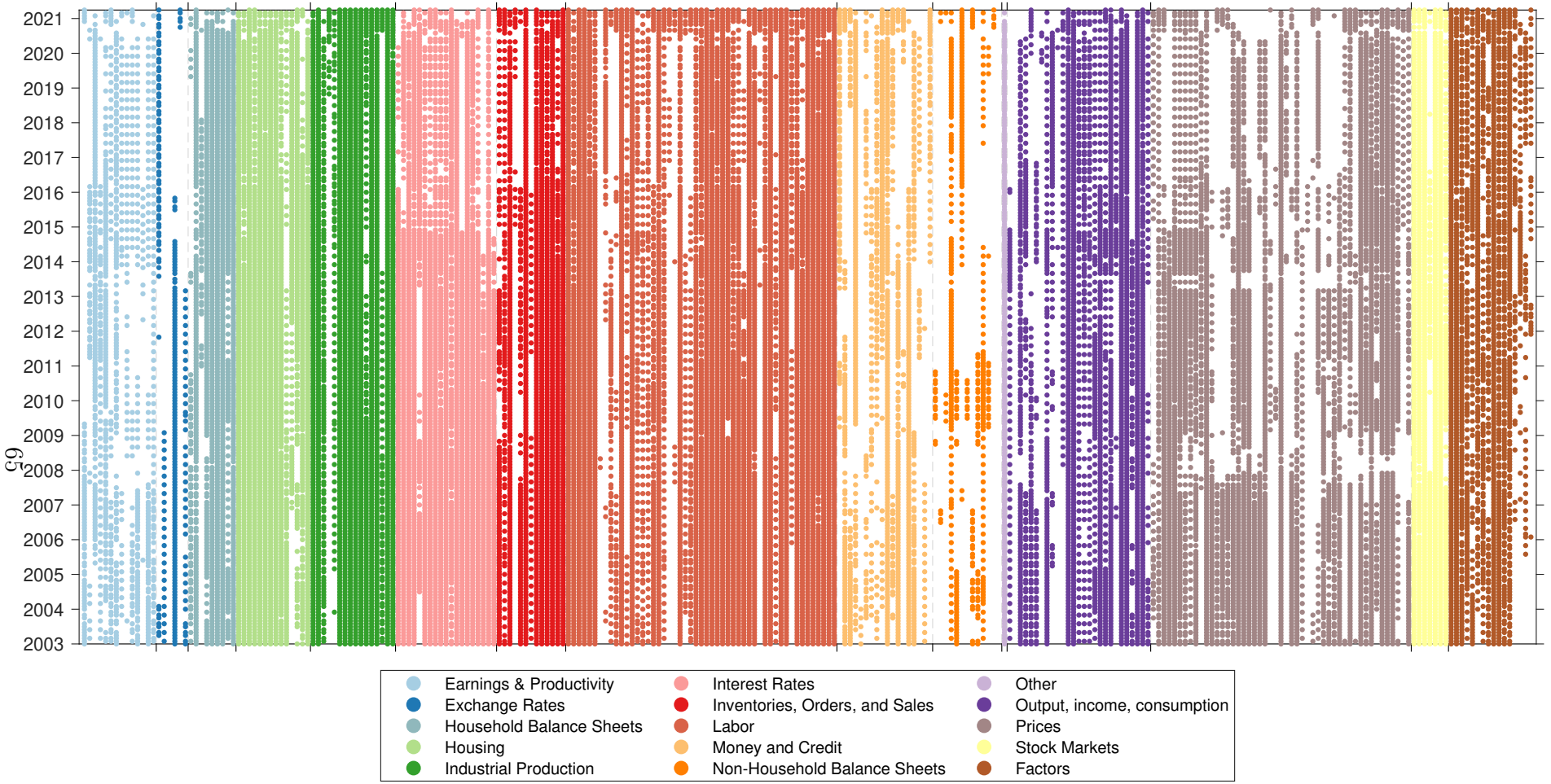


Figure 8: Sparsity pattern graph for hard-thresholding on D3

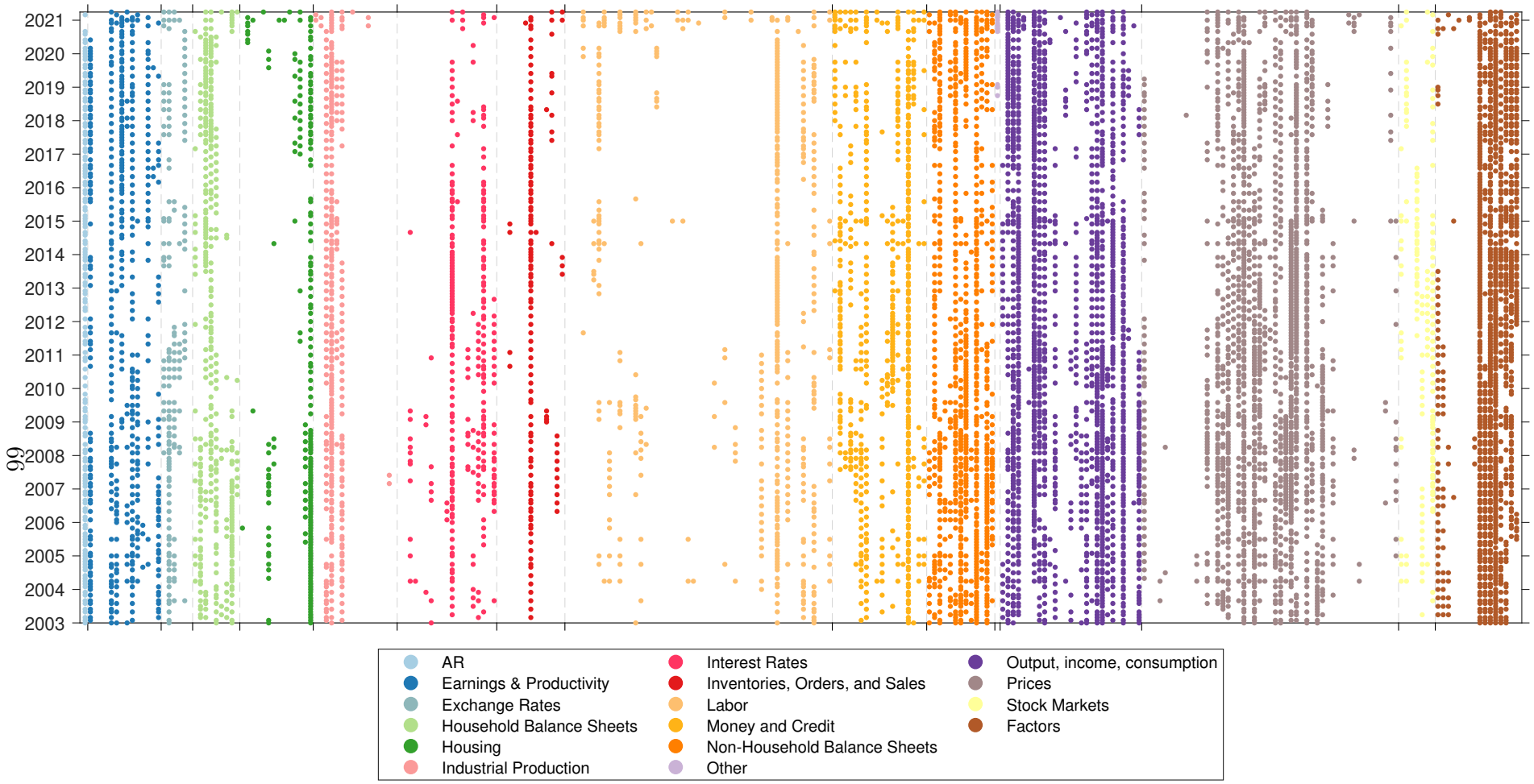


Figure 9: Sparsity patterns for Sg-LASSO-MIDAS on D3