INTERNATIONAL MONETARY FUND

GDP Nowcasting Performance of Traditional Econometric Models vs Machine-Learning Algorithms: Simulation and Case Studies

Klakow Akepanidtaworn and Korkrid Akepanidtaworn

WP/25/252

2025
DEC

WORKING PAPER

 WP/25/252

IMF Working Paper
Institute For Capacity Development

GDP Nowcasting Performance of Traditional Econometric Models vs Machine-Learning Algorithms: Simulation and Case Studies
Prepared by Klakow Akepanidtaworn and Korkrid Akepanidtaworn

Authorized for distribution by Natan Epstein
December 2025

ABSTRACT: Are Machine Learning (ML) algorithms superior to traditional econometric models for GDP nowcasting in a time series setting? Based on our evaluation of all models from both classes ever used in nowcasting across simulation and six country cases, traditional econometric models tend to outperform ML algorithms. Among the ML algorithms, linear ML algorithm – Lasso and Elastic Net – perform best in nowcasting, even surpassing traditional econometric models in cases of long GDP data and rich high-frequency indicators. Among the traditional econometric models, the Bridge and Dynamic Factor deliver the strongest empirical results, while Three-Pass Regression Filter performs well in our simulation. Due to the relatively short length of GDP series, complex and non-linear ML algorithms are prone to overfitting, which compromises their out-of-sample performance.

JEL Classification Numbers:	C52, C53, C55, C32
Keywords:	Nowcasting; Machine Learning; Forecast evaluation; Real-time data
Author’s E-Mail Address:	kakepanidtaworn@imf.org, korkrid.akepanidtaworn@microsoft.com

* “We would like to sincerely thank Andy Berg, John McDermott, Paul Cashin, Karim Barhoumi, Michal Andrle, Yunhui Zhao, Jiaxiong Yao, Felix Simione, Hany Abdel-Latif, and Iyke Maduako for their valuable comments, data, and insightful suggestions. Any remaining errors are our own.

WORKING PAPERS

GDP Nowcasting Performance of Traditional Econometric Models vs Machine-Learning Algorithms: Simulation and Case Studies

Prepared by Klakow Akepanidaworn, and Korkid Akepanidtaworn

Contents

Glossary

ML	Machine Learning
GDP	Gross Domestic Product
MIDAS	Mixed Data Sampling regressions
U-MIDAS	Unrestricted Mixed Data Sampling regression
DGP	Data-Generating Process
TRPF	Three-Pass Regression Filter
DFM	Dynamic Factor Model
PC	Principal Component
OLS	Ordinary Least Square
RFE	Recursive Feature Elimination
RMSE	Root Mean Square Error
OOS	Out-Of-Sample
LASSO	Least Absolute Shrinkage and Selection Operator
DT	Decision Tree
RF	Random Forest
SVRR	Support Vector Regression – Radial basis
SVRP	Support Vector Regression – Polynomial basis
SVRL	Support Vector Regression – Linear basis
NNET	Neural Network
LSTM	Long Short-Term Memory Neutral Network

1 Introduction

Nowcasting has become an essential tool in academic, finance, and government, to exploit high-frequency information to monitor economic activities. Stock and Watson (2017) regarded nowcasting as one of the top ten developments in twenty years of time series econometrics research. A wide range of models has been used in nowcasting, which we categorize into two classes: (1) traditional econometric models and (2) machine learning algorithms (ML), according to Table 1. The former were developed first and has been used by many institutions such as the Federal Reserve Bank of New York, the Federal Reserve Bank of Atlanta, and the Bank of England. Recently, ML algorithms have gained prominence in nowcasting applications, as used in Soybilgen and Yazgan (2021), Dauphin et al. (2022), Barhoumi et al. (2022)), and many others.

Table 1: List of Evaluated Models

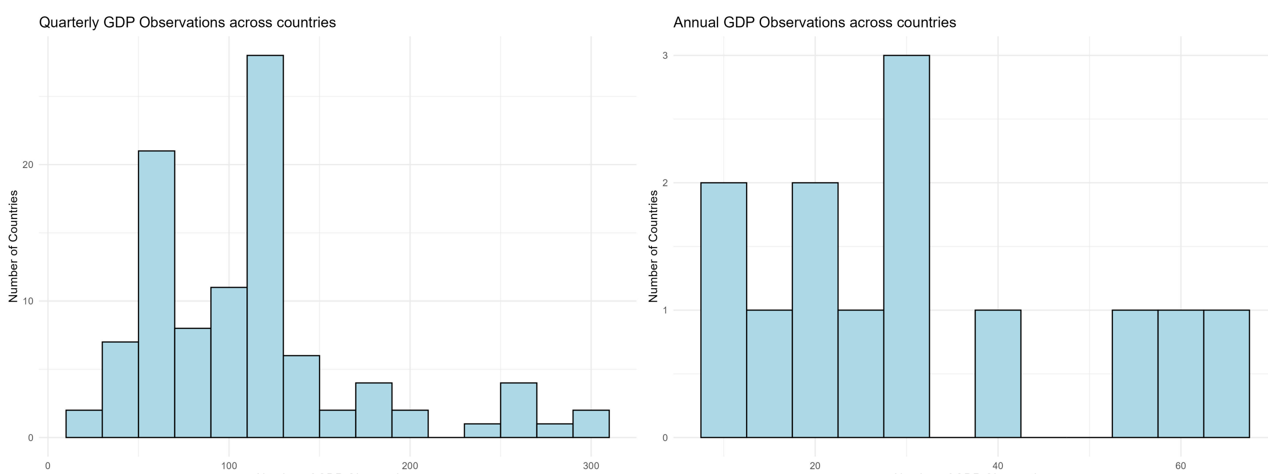
This table presents a list of models from the two classes - traditional econometric models and machine learning algorithms - that are evaluated in this paper.

Traditional Econometric Models	**Machine Learning Algorithms**
1. Bridge	1. LASSO
2. MIDAS	2. Elastic Net
3. U-MIDAS	3. Decision Tree (DT)
4. MIDAS-PC	4. Random Forest (RF)
5. U-MIDAS-PC	5. Light GBM
6. Three Pass Regression Filter (TPRF)	6. XGBoost
7. Dynamic Factor Model (DFM)	7. Support Vector Regression – Radial basis (SVRR)
	8. Support Vector Regression – Linear basis (SVRL)
	9. Support Vector Regression – Polynomial basis (SVRP)
	10. Neutral Network (NNET)
	11. Long Short-Term Memory Neutral Network (LSTM)
	Ensemble methods: LASSO-DT, LASSO-DT-SVRP, LASSO-SVRP-NNET, and LASSO-RF-SVRR

We thoroughly evaluate real-time GDP nowcasting performance of an extensive set of models from these two classes in a single-country time series context. Our evaluation is one of, if not, the most comprehensive in the literature, encompassing (1) multiple country cases, (2) both theoretical (simulation) and empirical settings, and (3) the broadest range of models. Unlike many studies that benchmark against a single model like a simple ARIMA, we rigorously compare performance of each model against almost all models that have been used in nowcasting.

Ex-ante, no theory establishes the superiority of any one model over others for GDP nowcasting. Literature found strong performance across both classes of models. Several studies such as Dauphin et al. (2022) and Forini and Marcellino (2013, 2014), highlighted that nowcasting performance varied based on data, countries, or time periods analyzed. Nonetheless, we question the effectiveness of ML or complex (highly parameterized) models in GDP nowcasting in a single-country application. A key practical challenge is the short span of GDP data for a majority of countries (Figure 1). Over half of the countries in the world have fewer than 100 GDP observations, with maximum around 300 observations, which are are relatively small in the context of machine learning applications. Thus, simple models can minimize the risk of overfitting and produce strong predictive performance.

Figure 1: Number of GDP observations across the world, as of July 6, 2024.
The left (right) chart plots a histogram of the number of quarterly (annual) GDP observations for countries that publish quarterly (annual) GDP.



Our simulations, which cover cases of both linear and nonlinear Data Generating Processes (DGP) reveal that certain traditional econometric model or linear ML algorithms demonstrate strongest out-of-sample performance. In particular, TPRF often ranks first in both linear and non-linear DGPs, followed by linear ML algorithms such as Elastic Net and

Lasso, respectively. Armong the ML algorithms, linear algorithms such as Elastic Net and Lasso tend to outperform more complex ones including random forest, support vector regressions (SVRL, SVRP, SVRR), and neural networks (NNET). Those linear ML algorithms emerge as the best performer, when the number of observations is sufficiently large, i.e. exceeding 100, and the DGP is linear. When the sample size is small, the TPRF performs the best.

Next, we assess the pseudo-real-time nowcasting performance of traditional economic models and ML algorithms, listed in Table 1, for each of the six countries: China, France, India, Mauritius, Türkiye, and Vietnam. Although the number of countries studied is limited, this set of countries cover diverse characteristics in terms of the availability of high-frequency indicators and GDP observations. In most countries, traditional econometric models, particularly Bridge or DFM, are the top performers. Notably, Bridge model delivers strong nowcasting performance in all the six countries. In Türkiye and India, linear ML algorithms – Lasso or Elastic Net – are the best performers. Among the ML algorithms, these two linear ML algorithms outperform complex, non-linear algorithms, suggesting limited nonlinearity or complex interactions in the underlying DGP.

Both our simulation and empirical analyses support the use of traditional econometric models, such as Bridge or DFM, and linear ML algorithms for single-country GDP nowcasting. These models reliably deliver robust nowcasting performance in the six country cases. Complex non-linear model could overfit short GDP series, resulting in poorer nowcasting performance, relative to simple models. As a caveat, our conclusions may depend on the simulation design, high-frequency indicators, and the choice of countries.

2 Literature Review

Nowcasting has emerged as an essential tool for economists across academia, policy institutions, and the private sector. The formal introduction of the term "nowcasting" into economics is attributed to the seminal work of Giannone et al. (2008). Their paper systematically illustrated how economists could incorporate real-time information to monitor the state of the economy, thereby catalyzing a significant increase in research and applications in this field. Today, numerous policy institutions, including central banks and financial institutions, have integrated nowcasting into their toolkits. Various methodologies are employed in nowcasting, which we broadly categorize into two classes: traditional econometric models and ML algorithms.

The early development of nowcasting primarily focused on traditional econometric models. While there had been work to utilize high-frequency information to monitor economic activities such as Miller and Chin (1996) and Rathjens and Robins (1993), Baffigi et al. (2004) formalized nowcasting with the 'Bridge' model, linking high- and low-frequency data via aggregation and ordinary least squares (OLS). Concurrently, Gyesels et al. (2004, 2007) developed a Mixed Data Sampling (MIDAS) regression model, which splits high-frequency data into multiple low-frequency variables while imposing parameter restrictions to mitigate parameter proliferation. Clements and Galvo (2008, 2009) further enhanced MIDAS by incorporating autoregressive (AR) terms. When frequency mismatches between the nowcast target and high-frequency data are minimal, Forini et al. (2011) proposed using OLS to estimate the Unrestricted Mixed Data Sampling (U-MIDAS) regression model, which does not place any no restriction on high-frequency parameters.

Factor models are another valuable tool in nowcasting, as they summarize information from numerous high-frequency indicators and effectively reduce the number of parameters. Stock and Watson (2002) formalized the use of factor models in forecasting applications, employing Principal Components (PC) for estimation. Marcellino and Schumacher (2010) and Forini and Marcelino (2014) integrated factor modeling with MIDAS, in the MIDAS-PC framework. Similarly, factors can be utilized in the U-MIDAS model, referred to as U-MIDAS-PC. Giannone et al. (2008) introduced a dynamic factor model (DFM), which can be estimated using the Kalman Filter algorithm, as in Doz et al. (2011). The DFM is particularly useful in nowcasting due to its capacity to address ragged edge problems and provide intuitive interpretations of news components and their impacts. Three-pass regression filter (TPRF), developed by Kelly and Pruitt (2015), can also be applied to extract factors with maximal correlation with a target variable. Forini and Marcellino (2013) offer a comprehensive survey of econometric methods relevant to nowcasting, primarily focusing on the traditional econometric models discussed above.

Recently, ML algorithms have been increasingly applied in nowcasting, due to their success in real-world predictive applications. James et al. (2013) explained workhorse ML algorithms used in predictive applications. Coulombe et al. (2022) argued non-linear ML is useful for macroeconomic forecasting especially under uncertainties, partly due to its ability to capture non-linearity and its emphasis on minimizing out-of-sample RMSE over in-sample loss. Hauzenberger et al. (2025) developed Bayesian Neural Networks, which were not included in this paper, to deal with the small number of observations, typical of macroeconomic data. Several studies have demonstrated the superior performance of ML algorithms in nowcasting in terms of minimizing RMSE. Examples included trees and random forest for US GDP (Soybilgen and Yazgan (2021)); macroeconomic random forest and linear gradient boosting for nowcasting World Trade (Chinn et al. (2023)); random forest for Dutch GDP (Kant et al. (2022)) and for Türkiye GDP (Bolhuis and Rayner (2020)); Ridge for China GDP (Zhang et al. (2023)); ML algorithms for European countries (Dauphin et al. (2022)) and for Sub-Saharan African countries (Barhoumi et al. (2022)); and ML algorithms with non-traditional data input for Bolivia (Bolivar (2024))[1].

Forecasting high-frequency indicators is an integral part of nowcasting underpinning all the methods above. For a suitable nowcast, the indicators must cover the entire period being nowcasted. If some high-frequency data have a few months or weeks not yet been released, they can be extended by using, for instance, ARMA models. Some models such as DFM can fill in the unreleased indicators internally, but they are still being forecasted. In our paper, we evaluate performance at quarter-end, assuming that all high-frequency indicators in each model are fully released. Hence, we do not analyze errors arising from forecasting high-frequency indicators.

All the above ML algorithms or some traditional econometric models are not designed to address frequency mismatch issues internally, as they are typically estimated at a single frequency. This problem is typically mitigated outside the model through temporal aggregation. In our ML applications, we report the lower RMSE from these two choices: averaging and blocking[2]. To specifically tackle frequency mismatch, Babii et al. (2022) introduced Lasso-MIDAS and Lasso-U-MIDAS. Babii et al. (2024) subsequently extended Lasso-MIDAS and Lasso-U-MIDAS from a single time series to panel data.

Numerous institutions have adopted nowcasting tools to assess the economy in real-time. The Federal Reserve Bank of New York employs the Dynamic Factor Model for GDP growth nowcasting, as detailed by Bok et al. (2017) and Almuzara et al. (2023). Higgins (2014)

[1] Bolivar (2024) used satellite imageries as non-traditional data in his analysis.

[2] To implement, we compute quarterly average of monthly variables. For blocking, we split monthly variables into three quarterly variables (each month relative to a quarter end).

describes the Bridge model underpinning the GDPNow tool of the Federal Reserve Bank of Atlanta. The Bank of England utilizes a suite of models for nowcasting UK and world GDP, such as MIDAS, DFM, and Mixed Frequency Bayesian VAR, as explained by Per Anesti et al. (2017), Kindberg-Hanlon and Sokol (2018), and Moreira (2025). The European Central Bank (ECB) employs the Bridge model and DFM for nowcasting, as explained in ECB (2008), Angelini et al. (2008), and Banbura and Saiz (2020). Linzenich and Meunier (2024) from the ECB have developed a programming toolkit for running several nowcasting models, including Bridge, DFM, and Bayesian VAR. Emerging market central banks also utilize relatively simple nowcasting methods; for instance, Heng et al. (2024) describe the Bridge, MIDAS, and U-MIDAS models employed by the National Bank of Cambodia. improve this sentence: Dauphin et al. (2022) and Barhoumi et al. (2022), from the IMF, applied various ML algorithms to GDP nowcasting in Europe and Sub-Saharan Africa respectively.

Many studies have evaluated the performance of different nowcasting models. While support exists for all models mentioned, a consensus in the literature is that no single model is universally superior, as nowcasting performance can vary significantly based on data, countries, or time periods, as echoed by Dauphin et al. (2022), Forini and Marcellino (2013, 2014), among others. Forini and Marcelino (2014) found that the simple Bridge model performed well overall in nowcasting Euro Area GDP and its subcomponents. Kuzin et al. (2011) reported that MIDAS-AR performed effectively for short horizons, while MF-VAR excelled in long horizons for Euro Area GDP nowcasting. Lizenich and Meunier (2024) established an automated framework for various traditional econometric nowcasting models, concluding that a four-factor DFM performed best in nowcasting global GDP. Anesti et al. (2017) and Moreira (2025) found that MIDAS was the top performer in nowcasting UK GDP among the models utilized by the Bank of England. Yet, when nowcasting World GDP in the BoE, Kindberg-Hanlon and Sokol (2018) noted MIDAS excelled in normal times, while DFM was more effective during crises. In contrast, Banbura and Saiz (2020) highlighted the findings from the ECB's 2015 review that DFM is more susceptible to structural changes, such as crises, compared to the Bridge model.

The closest studies to our research are Richardson et al. (2021), Zhang et al. (2023), and Kant et al. (2022). Richardson et al. (2021) found superior performance for ML algorithms over DFM in nowcasting New Zealand GDP. Zhang et al. (2023) evaluated a comprehensive list of both traditional econometric and ML algorithms for nowcasting China

GDP, concluding that linear ML algorithms (Ridge, Lasso, and Elastic Net) delivered superior performance. Kant et al. (2022) found Random Forest achieved the best performance in nowcasting Dutch GDP. This literature has found mixed results due to differences in modeling choices, countries, and sample periods. To help reconcile conflicting results, our paper provides rigorous valuation of nowcasting performance, comparing the most exhaustive lists of ML algorithms and traditional econometric models across multiple countries and simulations.

3 Competing Models

We assess predictive nowcasting performance of (1) traditional econometric models and (2) ML algorithms that have been utilized in nowcasting. Each model is applied in a time-series context (a single country) rather than in a panel data setting. Our evaluation presents a higher hurdle, as we compare each model against a comprehensive list of models from the opposing class, than many studies in this literature[3]. Additionally, we evaluate these models through simulations and empirically using data from six countries.

Table 1 lists traditional econometric models and ML algorithms evaluated in this paper. Those traditional econometric models differ in their methods for linking high-frequency data to low-frequency nowcast targets. Since these models do not incorporate a variable selection algorithm, we apply judgment to select variables based on their correlation with GDP growth or through a recursive addition and deletion of indicators.

The ML algorithms considered encompass most techniques commonly applied in nowcasting and forecasting. Except for Lasso and Elastic Net, all other ML algorithms are non-linear. In terms of complexity, Lasso and Elastic Net are the simplest. A key advantage of these ML models is their built-in regularization algorithms, which can effectively narrow down the set of variables from a large pool of high-frequency indicators. This regularization can handle cases where the number of cross-sectional variables exceeds the number of time observations ($N > T$), allowing for automatic variable selection without user intervention.

We report predictive performance of these models via out-of-sample root-mean-square errors (OOS RMSE) relative to a certain benchmark over an evaluation period (test set). In

[3]The literature typically assesses models against a simple benchmark, such as autoregressive (AR) models, or a limited number of competing models.

the simulation, we use AR(1) model as a benchmark, whereas we use the automatic ARIMA model as a benchmark in the empirical section. Let y_i be a GDP growth nowcast target, and $\hat{y}_i^{benchmark}$ be the GDP growth forecast by a benchmark model, for period i. For a model f with inputs X_i, the relative OOS RMSE is computed by:

$$Relative\ OOS\ RMSE = \frac{\overbrace{\sqrt{\sum_{i \in Test} (y_i - f(X_i))}}^{\text{OOS RMSE of a model, f}}}{\underbrace{\sqrt{\sum_{i \in Test} (y_i - \hat{y}_i^{benchmark})}}_{\text{OOS RMSE of a benchmark model}}}.$$

More details of both traditional econometric models and ML algorithms are described below.

3.1 Traditional econometric models

Throughout this and next subsections, we denote nowcast target (GDP) as y_q where q represents quarterly time (low frequency). Each high-frequency indicator, i, is denoted as $x_{i,t}$. When it is converted into the low frequency, we denote as $x_{i,q}$. Error term is represented by ϵ_q.

3.1.1 Bridge

The Bridge model, formalized by Baffigi et al. (2004), converts high-frequency indicators into low-frequency using temporal aggregations such as period averages, sums, or end-period values. Due to frequency consolidation, the Bridge model converts all high-frequency indicators into the same low frequency as the nowcast target (GDP growth), allowing it to be estimated using the ordinary least squares (OLS) method. Unreleased high-frequency indicators must be forecasted to aggregate them into the low frequency. Mathematically, the Bridge equation can be represented as:

$$y_q = \beta_0 + \sum_{l=1}^{m} \gamma_l y_{q-l} + \sum_{i=1}^{n} \beta_i x_{i.q} + \epsilon_q$$

3.1.2 MIDAS

Gyesels et al. (2004, 2007) introduced a MIDAS regression involving data in multiple frequencies. In our setting, the MIDAS approach splits high-frequency indicators into low-frequency (quarterly) relative to the end of a quarter[4]. For example, a monthly indicator may naturally be split into three quarterly variables: one for the last month of the quarter, one for the second-to-last month, and one for the first month of a quarter[5].

As each split variable theoretically adds one coefficient to be estimated, MIDAS imposes a functional form on the coefficients of the split high-frequency indicators, to reduce the number of parameters. We employ the Almon polynomial restrictions of degree 2 (delta i,h) , a common choice in MIDAS applications that accommodates numerous lags of high-frequency indicators with just two parameters[6]. For a month relative to end-quarter, h, the MIDAS model is estimated using non-linear least squares and can be expressed as follows:

$$\begin{aligned} y_q &= \beta_0 + \sum_{l=1}^{m} \gamma_l y_{q-l} + \sum_{i=1}^{n} \sum_{h=1}^{L(i)} \delta_{i,h} x_{i.3q-h} + \epsilon_q \\ \delta_{i.h}(w_{1,i}, w_{2,i}) &= \frac{\exp(w_{1,i}h + w_{2,i}h^2)}{\sum_{h=1}^{L(i)} \exp(w_{1,i}h + w_{2,i}h^2)} \end{aligned}$$

The MIDAS model estimated in our paper includes AR terms, which can be estimated using the methodology of Clements and Galvo (2008, 2009).

3.1.3 U-MIDAS

The Unrestricted MIDAS (U-MIDAS) approach, formally introduced by Forini et al. (2011), can be applied to nowcasting. In the U-MIDAS model, high-frequency indicators are split into multiple low-frequency variables and estimated without imposing restrictions on their coefficients using OLS in the low frequency. This approach is particularly effective when the frequency mismatch is minimal.

The U-MIDAS model can be expressed as follows:

[4] This practice is sometimes referred to as "blocking".

[5] One can have many more lags. Given the quarterly-monthly frequency mismatch, if a monthly indicator is split into 3 lags or more, it will take up values in the previous quarter.

[6] Almon polynomial of degree 2 allows creating coefficient of h^{th} lag by using 2 parameters w_1, w_2.

$$y_q = \beta_0 + \sum_{l=1}^{m} \gamma_l y_{q-l} + \sum_{i=1}^{n} \sum_{h=1}^{L(i)} \beta_{i,h} x_{i.3q-h} + \epsilon_q$$

3.1.4 MIDAS-PC

When dealing with numerous high-frequency indicators, mixed-frequency factor models can be employed to summarize the information among indicators, as demonstrated by Stock and Watson (2002). Marcellino and Schumacher (2010) and Forini and Marcelino (2014) explicitly utilized factors in MIDAS models. These factors can be extracted by using principal components of high-frequency indicators. The model – MIDAS-PC - is estimated similarly to the MIDAS, with principal components being regressors. Using fewer principal components than the original set of high-frequency indicators mitigates parameter proliferation problems.

3.1.5 UMIDAS-PC

This is similar to UMIDAS but replaces the original variables with principal components. in In cases where the number of factors in the model is small, it is unnecessary to impose MIDAS restrictions on coefficient parameters. Consequently, the U-MIDAS-PC model can be estimated without coefficient restrictions using OLS in the low frequency.

3.1.6 Three-pass Regression Filter (TPRF)

Kelly and Pruitt (2015) introduced the Three-Pass Regression Filter (TPRF), a method for forecasting with multiple predictors. TPRF algorithm extracts factors in a way that maximally correlates with the forecast target. An advantage of using TPRF factors in nowcasting is that they optimally assign loadings to indicators based on their correlation with real GDP growth, resulting in a factor that closely tracks GDP in principle. After converting high-frequency indicators into low frequency, the TPRF model is estimated in three steps:

Step 1: Run a time series regression of the nowcast target on each high-frequency indicator, $x_{i.q}$.

$$y_q = \beta_0 + \beta_i x_{i,q} + \epsilon_{i,q}$$

Step 2: For each time period t, rrun a cross-sectional regression of X on the beta coefficients from Step 1. This yields a TPRF factor over time ($TPRF_t$).

$$x_{i,t} = \beta_0 + TPRF_t \beta_i + \epsilon_i$$

Step 3: Using the TPRF factor from Step 2, run the nowcasting regression in low frequency.

$$y_q = \beta_0 + \sum_{l=1}^{m} \gamma_l y_{q-l} + \theta TPRF_q + \epsilon_q$$

There is no look-ahead bias, because both steps 1 and 2 only involve lagged GDP and are effectively estimated with past-quarter data. The nowcast is computed by plugging in the current quarter value of $TPRF_q$, in the third step.

3.1.7 Dynamic Factor Model (DFM)

Giannone et al. (2008) formalized the use of the Dynamic Factor Model for nowcasting. Since its introduction, it has been employed in various studies and by policy institutions, such as the Federal Reserve Bank of New York (see Bok et al. (2017) and Almuzara et al. (2023)). The DFM addresses frequency mismatch and ragged-edge problems, which are two primary challenges in nowcasting[7]. The DFM allows for immediate updates to nowcasts upon the release of any high-frequency indicator, even when they are released asynchronously. The DFM can be estimated via Kalman Filtering, as described by Doz et al. (2011).

Let $Z_q = [y_q, x_{1,q}, \ldots, x_{N,q}]$ denote a $(N+1) \times q$ matrix of real GDP growth and transformed high-frequency indicators and $F = [f_1, \ldots, f_r]$ denote a $r \times q$ matrix of unobserved factors. The DFM can be expressed as

$$z_{i,q} = \mu + \Lambda F_q + \epsilon_{i.q}, z_{i,q} \in Z_q$$
$$F_q = AF_{q-1} + Bu_q, u_q \sim N(0, I)$$

We further assume $\epsilon'_{i.q}s$ are cross-sectionally white-noise, and orthogonal to the common shock, u_q. Kalman Filter can be applied to extract the factor, F_q, under this structure.

The DFM GDP nowcast can be obtained by first regressing y_q on the estimated factor, $\hat{F}_q$ with other control variables, and then calculating the nowcast using current value of $\hat{F}_q$ in the estimated regression.

[7]Dynamic factor models (DFMs) solve the 'ragged edge' problem by filling in missing data via Kalman filtering. The Kalman filter helps project both observed and state variables, efficiently controlling for missing data in mixed-frequency datasets.

3.2 Machine Learning Algorithms

ML algorithms have gained considerable traction in real-world economic applications due to their predictive performance. As highlighted in Section 2, multiple studies have demonstrated the effectiveness of ML algorithms in nowcasting GDP, prompting policy and financial institutions to increasingly integrate these methods into their nowcasting toolkits. In this paper, we evaluate the machine learning algorithms described below. We employ five-fold cross-validation approach across various hyperparameter configurations to select parameters[8]. Table 2 summarizes key parameter settings in our empirical implementation of each ML algorithm.

3.2.1 Least Absolute Shrinkage and Selection Operator Regression (Lasso)

Lasso regression is a powerful algorithm that builds in variable selection by shrinking the coefficients of less important variables toward or equal to zero, resulting in a more parsimonious and interpretable model. The Lasso objective function is given by

$$\beta = \text{argmin} \left(\underbrace{\sum_{q=1}^{T}(y_q - \beta_0 - \sum_{j=1}^{k} x_{j,q}\beta_j)^2 +}_{\text{OLS objective}} \qquad \underbrace{+\lambda \sum_{j=1}^{k} |\beta_j|}_{\text{Lasso Penalty, "L1 regularization"}} \right)$$

3.2.2 Elastic Net

Similar to Lasso, Elastic Net is a regularization technique that combines both L1 (Lasso) and L2 (Ridge) penalties with a mixing parameter (α), controlling the weight between the two penalties. Its objective function is given by:

$$\beta = \text{argmin} \left(\underbrace{\sum_{q=1}^{T}(y_q - \beta_0 - \sum_{j=1}^{k} x_{j,q}\beta_j)^2 +}_{\text{OLS objective}} \quad \lambda \times \left[\underbrace{\alpha \sum_{j=1}^{k} |\beta_j| +}_{\text{Lasso Penalty}} \quad \underbrace{(1-\alpha) \sum_{j=1}^{k} \beta_j^2}_{\text{Ridge Penalty}} \right] \right)$$

We tune this model based on 3,150 different combinations of penalty strengths (λ) and mixing values (α) using 5-fold cross-validation as described in Table 2.

[8] In 5-fold cross-validation, a dataset is split into five non-overlapping blocks or folds. The model is trained on four of the folds and tested on the omitted fold. This process is repeated five times, with a different omitted fold used as the test set each time. The models' hyperparameters are chosen by picking values that give the best performance on the average of the results from all five iterations

Table 2: Machine-Learning Parameter Tuning

This table summarizes the parameter values used to tune each machine learning model, along with their descriptions and the total number of parameter combinations.

Models	Parameter	Values	Description	Total Combinations
Lasso	λ	0.0001 to 100	Penalty strength for shrinking coefficients. Higher values denotes greater penalty.	100
Elastic Net	α	$[0, 0.05.0.1, \ldots, 1]$	Controls the mix between Ridge (0) and LASSO (1).	21
	λ	$[0.0001, , \ldots, 100]$	Penalty strength for shrinking coefficients. Higher values denotes greater penalty.	150
Decision Tree (DT)	cp	0.0001 to 0.1	tree pruning control	20
Random Forest	mytr	2, 4, 6, ..., up to number of predictors e.g., if 10 predictors, then 2, 4, 6, 8, 10 = 5)	Number of variables randomly sampled as candidates at each split.	Depends on predictors
LightGBM	n_estimators	50, 100	Number of boosting rounds (trees). More trees can improve accuracy but may overfit.	2
	max_depth	2, 4, 8	Maximum depth of each tree. Controls complexity; deeper trees can capture more patterns but may overfit.	3
	learning_rate	0.01, 0.05, 0.1	Step size for updating weights. Lower values make learning slower but safer from overfitting.	3
XGBoost	n_estimators	50, 100	Number of trees (boosting rounds). More trees can improve accuracy but increase computation.	2
	max_depth	2, 4, 8	Maximum depth of each tree. Controls model complexity; deeper trees can capture more patterns but may overfit.	3
	learning_rate	0.01, 0.05, 0.1	Step size shrinkage used in updates to prevent overfitting. Lower values make learning slower but can improve generalization.	3
SVRL (Linear Kernel)	C	$[10^{-3}, 10^{2}]$(log scale)	Regularization strength; controls trade-off between error and margin width	10
SVRR (Radial Kernel)	σ	0.001, 0.01, 0.05, 0.1, 0.2	Controls the spread of the kernel; affects how far each point influences	5
	C	0.1, 1, 10, 100, 1000	Regularization strength; same as above	5
SVRP (Polynomial Kernel)	C	$[10^{-2}, 10^{2}]$(log scale)	Regularization strength; same as above	5
	Degree	2,4,8	Degree of the polynomial kernel; controls curve flexibility	3
	Scale	0.1,1,10	Scaling factor for the polynomial kernel; affects feature scaling	3
NNET	Hidden Layer	$3, 5, 10$	Number of neurons	3
	Regularization	$0.0001, 0.001, 0.01, 0.1$	Weight decay	4
LSTM	LSTM Layers	1,2,3	Number of layers	3
	Units per Layer	16 to 128 (step 16, tuned for each layer)	Number of neurons	8
	Activation	tanh	Activation function	1
	Learning Rate	0.0001 to 0.01 (log scale)	step size during optimization	depends on optimizer
	Epochs	30	number of complete pass on data	1
	Batch Size	8	number of independent sequences in forward pass	1
	Validation Split	0.1	percent of data set aside for validation	1

3.2.3 Decision Tree

A decision tree, introduced by Breiman et al. (1984), follows a hierarchical framework, consisting of several key components: the root node, internal nodes (decision nodes), branches, and leaf nodes (terminal nodes). Each internal node applies a decision rule to guide the observations into the most appropriate subset, thereby refining the predictive capabilities of the tree. Leaf nodes represent the endpoints of the decision process and encapsulate the learned patterns in the dataset. The key complexity parameter (cp) controls tree pruning: lower values allow more splits, and higher values make the tree simpler. The smaller step size allows for a more detailed search for the best cp value.

3.2.4 Random Forest

Random forest, introduced by Breiman (2001), is a workhorse machine learning algorithm that combines the output of multiple decision trees to reach a single result. The core principle is to optimize individual decision trees within the forest for best predictive performance. We tune the Random Forest using the "mtry" parameter, which controls the number of variable randomly sampled at each split and determines how many predictors are considered for splitting at each tree node.

3.2.5 LightGBM

Introduced by Microsoft, LightGBM is an open-source framework for efficient gradient boosting on large datasets, designed for speed and scalability. Boosting fits each tree to the residuals of the previous tree. By fitting small trees to the residuals repeatedly, the model slowly improves in areas where it previously fitted poorly. Typically, for good performance, each tree should be small with very few terminal nodes. In our implementation, we test 18 different parameter combinations for LightGBM using grid search.

3.2.6 XGBoost

Extreme Gradient Boosting (XGBoost) builds upon the fundamental principles of gradient boosting by iteratively training weak learners—typically decision trees—to optimally correct the residual errors of prior iterations. It embeds regularization, sparse-aware learning, and parallel execution, making it a highly effective model for structured data. While LightGBM is the state-of-the-art library by Microsoft, we want to test XGBoost as well given different implementations and optimizations. Performance can vary depending on the country. In our implementation, we tune three key parameters using grid search, across all 18 combinations.

3.2.7 Support Vector Regression (SVR)

Support Vector Regression (SVR) seeks a hyperplane that fits data in continuous space, mapping inputs to a higher-dimensional feature space while minimizing error. SVR employs kernel functions to manage non-linear relationships, aiming to include points within the decision boundary while ensuring the best-fit line contains the most points. In our paper, we evaluate the SVR model using three different kernel types: Linear (SVRL), Radial Basis Function (SVRR), and Polynomial (SVRP). Each kernel captures distinct patterns, depending on whether the relationship is linear or nonlinear. Since economic data such as GDP can exhibit both linear and nonlinear trends—depending on the country, time period, or external factors—we aim to capture this complexity while avoiding overfitting to a single pattern. The objective of SVR is to identify a function, $f(x)$, that minimizes the following regularized loss function, ensuring that predictions within an ε-margin of the true values are not penalized.

$$L(\omega, b) = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n} L_\epsilon(y_i, f(x_i))$$

3.2.8 Deep Neural Network (NNET)

Neural networks, inspired by the human brain, mimic how biological neurons communicate. Typically, input is received by the first layer, processed by the hidden layers, and the output layer (GDP growth) produces the result. We implement a Feedforward Neutral Network (nnet) in a single hidden layer. In total, we tune the model across 12 combinations of neuron size and decay parameter values.

3.2.9 Long Short-Term Memory (LSTM)

LSTM neural networks, based on Hochreiter (1997), excel at capturing long-term dependencies, making them ideal for sequence prediction tasks. Unlike traditional neural networks, LSTMs incorporate feedback connections, enabling them to process entire data sequences rather than individual data points. We use Keras Tuner (RandomSearch, 5 max trials) to automatically search for the best model configuration.

3.2.10 Ensemble learning

Ensemble learning is a machine learning technique that combines multiple models to achieve better performance than any single model alone. By leveraging the strengths and diversity of each model, ensemble methods can reduce overall error rates while preserving each model's unique advantages, such as low bias for specific data subsets.

In this paper, we experiment with a simple averaging ensemble and experiment with a few selected models. These ensemble models do not have their own parameters themselves as they simply average the predictions from the selected models. We calculate the following ensembles: (1) average of LASSO and Decision Tree predictions (LASSO_DT); (2) average of LASSO, Decision Tree, and SVRP predictions (LASSO_DT_SVRP); (3) average of LASSO, Random Forest, and SVRR predictions (LASSO_RF_SVRBF); (4) average of LASSO, SVRP, and Neural Network predictions (LASSO_SVRP_NN).

4 Simulation

4.1 Setup and Assumptions

This section outlines the factor structure that underpins the simulation. The unobservable factor, $F(m)$, influences both high-frequency data, $X(m)$, and the low-frequency nowcast target, $y(q)$. For simplicity, we address the frequency mismatch between monthly observed data and quarterly nowcast targets, thus setting $q = 3m$. Alongside the factor, $F(m)$, we introduce unobservable factors $G(m)$, which exclusively drive the observed high-frequency data. Both factors, F and G, are characterized by autoregressive structures featuring heterogeneous loading and independent error terms.

$$
\begin{aligned}
F =&(f_1, f_2, \ldots, f_k) \\
G =&(f_1, f_2, \ldots, g_l) \\
F_t =&\lambda_i F_{t-1} + \eta_t^f, \eta_t^f \sim^{iid} N(0,1) \\
G_t =&\lambda_j G_{t-1} + \eta_t^g, \eta_t^g \sim^{iid} N(0,1) \\
\lambda \sim&Uniform(\lambda^{Lo}, \lambda^{Hi})
\end{aligned}
$$

The high-frequency data matrices, $X(m)$, consist of unobservable components $(X^U(m))$ and observed components, $(X^O(m))$. In this setup, only the unobservable components enter the data-generating process (DGP) for the nowcast target. We denote the unobservable components as the first k components $(X_1, \ldots, X_k)$, while the observable components are indexed from $X_{k+1} to X_N$. The observed components are driven by both factors, F and G, whereas the unobservable components are solely driven by F(m) and contribute to the DGP for the nowcast target. This setup is realistic, as observed high-frequency indicators often intermingle with various factors, some of which may be unrelated to GDP. Mathematically, the evolution of each component of the high-frequency data can be expressed as follows:

$$
\begin{aligned}
X_U(i,t) =&\beta F + \epsilon_U, \epsilon_U \sim^{iid} N(0,1), j = 1 \text{ to } k \\
X_O(i,t) =&\beta^U F + \beta^O G + \epsilon_O, \epsilon_O \sim^{iid} N(0,1), j = k+1 \text{ to } N \\
\beta, \beta^U, \beta^O \sim&Uniform(\beta^{Lo}, \beta^{Hi})
\end{aligned}
$$

The low-frequency nowcast target, Y, is determined solely by the unobserved high-frequency data $(X^U(m))$. Although Y is solely generated by X^U, it remains correlated with X^O due to the influences of the factors, F. Given the quarterly-monthly frequency mismatch, each month of the quarter impacts the nowcast target and is denoted by$X_U(I, 3m - 0), X_U(I, 3m - 1), X_U(I, 3m - 2)$[9]. We simulate three distinct data-generating processes for Y, referred to as the true DGP.

DGP 1: Linear Data-Generating Process.

[9]Each month of X is divided into three quarterly observations relative to a quarter's end. For instance, $X_U(I, 3m - 1)$ refers to an observation one month prior to a quarter's end. Although the natural frequency mismatch is three, monthly observations can be partitioned into as many quarterly observations as desired.

$$
\begin{aligned}
y_t = 1.2 &+ 0.5x_1^{(0)} + 0.5x_1^{(-1)} + 0.5x_1^{(-2)} \\
&+ 0.3x_2^{(0)} + 0.6x_2^{(-1)} + 0.4x_2^{(-2)} \\
&- 0.25x_3^{(0)} - 0.2x_3^{(-1)} - 0.3x_3^{(-2)}
\end{aligned}
$$

DGP 2: Non-Linear Quadratic Data-Generating Process

$$
\begin{aligned}
y_t = 1.2 &+ 0.5x_1^{(0)} + 0.5x_1^{(-1)} + 0.5x_1^{(-2)} \\
&+ 0.3x_2^{(0)} + 0.6x_2^{(-1)} + 0.4x_2^{(-2)} \\
&- 0.25x_3^{(0)} - 0.2x_3^{(-1)} - 0.3x_3^{(-2)} \\
&+ 0.05X_{2,(0)}^2 + 0.02X_{2,(-1)}^2 + 0.03X_{2,(-2)}^2
\end{aligned}
$$

DGP 3: Non-Linear Data-Generating Process with Quadratic and Interaction Terms

$$
\begin{aligned}
y_t = 1.2 &+ 0.5x_1^{(0)} + 0.5x_1^{(-1)} + 0.5x_1^{(-2)} \\
&+ 0.3x_2^{(0)} + 0.6x_2^{(-1)} + 0.4x_2^{(-2)} \\
&- 0.25x_3^{(0)} - 0.2x_3^{(-1)} - 0.3x_3^{(-2)} \\
&+ 0.05X_{2,(0)}^2 + 0.02X_{2,(-1)}^2 + 0.03X_{2,(-2)}^2 \\
&+ 0.15X_{3,(0)}X_{4,(0)} + 0.15X_{3,(-1)}X_{4,(-1)} + 0.15X_{3,(-2)}X_{4,(-2)} \\
&- 0.2X_{4,(0)}X_{5,(0)} - 0.3X_{4,(-1)} * X_{5,(-1)} - 0.1X_{4,(-2)}X_{5,(-2)}
\end{aligned}
$$

In the simulation, we set $K = 5$ and $N = 105$, indicating that five unobservable components generate GDP. The components $X_1, X_2,$andX_3 are included in all DGPs. The loadings on X_1 across each month in a quarter are homogeneous, while those for $X_2$and X_3 are heterogeneous. While DGP 1 represents a purely linear model, DGPs 2 and 3 are non-linear; specifically, DGP 2 features quadratic terms, while DGP 3 includes non-linear interaction terms between X_4 and X_5.

This framework highlights key issues relevant to the practical application of GDP nowcasting. First, the true data-generating process for GDP is unknown, as captured by the unobservable nature of X_1 to X_5. Second, the true data-generating process could be non-linear as in DGPs 2 and 3, which may not be well approximated by linear models. Third, high-frequency indicators are relevant, but imperfect predictors of GDP, as they share the common factors, F, with GDP but also load on unrelated factors, G.

4.2 Monte Carlo Experiments

We conduct Monte Carlo experiments to evaluate nowcasting performance. A total of 500 data draws were simulated, according to the data-generating processes in Section 4.1. We set the number of factors, F and G, to 5 and 3, respectively, yielding a reasonable factor structure with observed correlations between simulated$X$ and nowcast targets ranging from 0.2 to 0.6[10]. The nowcast target is generated solely by unobserved X_1 to X_5, which are influenced only by factors, F. Additionally, we set the number of observed high-frequency indicators to 120, reflecting a scenario with a broad array of potential predictors. The high-frequency indicators are derived from both factors F and G. Other parameters are established as follows:

$$\begin{aligned} \lambda &\sim^{iid} Uniform(0.4, 0.6) \\ \beta &\sim^{iid} Uniform(0, 0.9) \\ \eta_f, \eta_g, \epsilon &\sim^{iid} N(0, 1) \end{aligned}$$

This parameterization implies the factors are independent and that the data exhibits mild persistence. Observed data are driven by these factors and include idiosyncratic error terms. This range of beta implies that some observed indicators may have significant loading on the factor driving GDP, (F).

We also evaluate nowcasting performance as a function of time observations. A limited number of time observations is, in practice, a significant challenge for nowcasting, especially for developing countries. To mimic the availability of actual GDP data, we vary quarterly time dimension as follows: $T_q = 40, 50, 60, 70, 80, 100, 120, 150, 200$. In the monthly frequency, the corresponding T_m's are $120, 150, 180, 210, 240, 300, 360, 450, 600$ respectively. For each

[10]Should the correlations between observed indicators and the nowcast target be excessively low, the nowcasting models will not perform effectively, regardless of methodology.

time dimension, our simulation generates 500 iterations of η_f and η_g, β, and λ to create synthetic data.

In each iteration, we randomly split 70% of data in time dimension into a training set, with the remaining 30% reserved for a test set[11]. For instance, when $T = 80$, the training and test sets comprise 56 and 34 observations of X_U, and Y, respectively. Each of these sets is drawn differently for each of the 500 iterations. The training set is utilized for model estimation, while models are subsequently applied to the test set to generate nowcasts[12]. The metric for evaluating model performance is the relative OOS RMSE calculated on the test set.

Candidate models in Section 3 are evaluated based on their relative RMSEs. A key challenge for traditional econometric models is their inability to perform variable selection among the 120 high-frequency indicators. Traditional econometric models cannot be estimated if the number of regressors exceeds the number of time periods. To narrow down the list of regressors, we algorithmically apply a recursive process of adding and dropping indicators using a Recursive Feature Elimination (RFE) technique with a constraint on the maximum number of variables based on time observations[13]. When T is small, the constraint on the maximum number of variables set in RFE limits the number of selected variables, mimicking low-data environment. The same selected indicators are then used in other traditional econometric models and neural networks so that the difference in model performance can be attributed to models, rather than data input.

Many ML algorithms considered in this section build in feature selection algorithms that effectively narrow down variables. Our ML simulation benefits from those techniques as 361 low-frequency variables (split from 120 HF indicators) far exceed the available time observations (maximum of 150)[14]. Estimation is conducted using 5-fold cross-validation within the training set. The cross-validation method selects parameters that minimize RMSE within each fold of the training set.

[11]This 70/30 choice ensures a sufficient number of observations in a test set, even when T is very small.

[12]To optimize runtime, we do not perform rolling updates to a model to calculate error in the test set in the theoretical simulation because the DGP in the simulation does not change over time.

[13]This selection process is facilitated by the Recursive Feature Elimination (rfe) package in R. The number of indicators selected ranges from 2 to the number of quarterly time periods divided by 5, mitigating parameter proliferation issues while ensuring adequate variable selection.

[14]In total, 120 monthly indicators are transformed via split-sampling (blocking) into 360 quarterly variables, with the addition of one extra variable for the first lag of Y.

4.3 Results

In this section, we highlight the four key insights from the simulation.

First, across all DGPs, the TPRF frequently ranks first, followed by linear ML algorithms such as Elastic Net and Lasso. However, linear ML algorithms emerge as the winner in some instances. In Table 3, the Elastic Net outperforms all models when the number of time observations exceeds 100. In such conditions – (1) a sufficient number of observations and (2) linear DGP - linear ML algorithms closely mimic the true DGP and effectively select relevant predictors.

Second, the superior performance of TPRF in DGPs 2 and 3 (as shown in Tables 4 and 5) is surprising given the nonlinearity in these DGPs. We ex-ante anticipate that non-linear ML algorithms would outperform traditional econometric models due to their ability to capture non-linearity. Yet, the TPRF model delivers the best nowcast. We conjecture that TPRF may be able to proxy to non-linearity to a certain degree, through the factor loadings on high-frequency indicators. In addition, the functional forms of non-linearity modeled in ML algorithms are not the same as the quadratic and interaction terms in DGPs 2 and 3. Therefore, TPRF may approximate non-linearity better than ML algorithms.

Third, some traditional econometric models and ML algorithms perform well under a short sample. In particular, the TPRF excels, when the number of observations is limited (T=40) in Table 3. Other traditional econometric models, such as MIDAS-PC and U-MIDAS-PC, also perform well. Linear ML algorithms—Elastic Net and Lasso—demonstrate strong performance under a short sample, as demonstrated in Tables 3 and 4 (DGP 1 and 2). One common feature among these models is that they minimizes the risk of overfitting through regularization or factor modeling.

Lastly, when the DGP is highly nonlinear featuring interactions, both classes of models may underperform. In Table 5, while TPRF still remains the top performer, its outperformance over other models is not economically and statistically significant. Among the ML algorithms, the non-linear SVR with Radial basis achieves the best results by a small margin over linear ML algorithms.

Our simulation results are in favor of traditional econometric models or linear ML algorithms. Given short GDP series, the risk of overfitting when using complex, non-linear models is high. Using factors such as those in TPRF, or regularization in linear ML algorithms can yield superior out-of-sample nowcasting performance. As a caveat, our conclusions may depend on simulation design and parameterization. In the next section, we will test whether empirical results from six countries confirm the simulation's findings.

5 Empirical Applications

We evaluate the pseudo-real-time out-of-sample GDP nowcasting performance among traditional econometric models vs ML algorithms, as outlined in Section 3, using empirical data from six counties: China, France, India, Mauritius, Türkiye, and Vietnam. The performance criterion is the predictive accuracy (relative OOS RMSE) of each class of models. These six country cases cover various country characteristics, including data availability, sample length, and economic complexity.

5.1 Data Cleaning

Our main source of data for the nowcast target (official real GDP growth) and high-frequency (monthly) indicators is Haver Analytics, which provides a comprehensive array of traditional macroeconomic data commonly utilized in macroeconomic surveillance. In practice, one can supplement Haver data with other data sources that may contain additional useful HF indicators for nowcasting. We gather available domestic monthly indicators that correlate with real GDP growth for each country. Several following steps are performed to prepare the data for analysis.

First, we drop monthly indicators with relatively short time series, as they shorten our estimation samples. This implies chosen indicators for the analysis are timely updated and have long history. Second, we drop monthly indicators with multiple consecutive missing values. For indicators with very few non-consecutive missing values, we employ quadratic interpolation to fill in the gaps. Third, to avoid multicollinearity, we pick a single monthly indicator from a group of indicators exhibiting high correlations (absolute values exceeding 0.95). Lastly, we drop time periods where GDP or any of the monthly indicators are missing. These steps ensure a dataset that is both complete and balanced for our analysis.

Next, we eliminate seasonality by applying X13 seasonal adjustment to variables exhibiting seasonal patterns. For non-seasonal variables or those seasonally adjusted by Haver Analytics, we retain their original values. To ensure stationarity, we next apply log-difference

Table 3: Relative Out-of-sample RMSE, DGP1: Linear

This table presents the average relative out-of-sample RMSEs of various models compared to the AR(1) benchmark, based on Monte Carlo simulations of the linear DGP across different sample sizes (T). Standard errors are computed from the distribution of 500 Monte Carlo draws. A lower RMSE indicates superior nowcasting performance. Bold values highlight the best-performing model for each time sample size.

	Standard Nowcasting Models							**ML Models**								
T	Bridge	UMIDAS	MIDAS	UMIDAS_PC	MIDAS_PC	TPRF	DFM	Lasso	Enet	SVRR	SVRL	SVRP	RF	DT	NNET
40	1.09	1.17	1.04	0.93	0.75	**0.62**	1.22	0.81	0.76	0.90	0.93	1.22	0.92	1.13	1.01
	(0.17)	(0.22)	(0.17)	(0.21)	(0.20)	**(0.11)**	(0.19)	(0.10)	(0.09)	(0.08)	(0.09)	(0.19)	(0.09)	(0.16)	(0.08)
50	1.02	1.17	1.02	0.82	0.72	**0.58**	1.11	0.72	0.67	0.87	0.90	1.09	0.90	1.11	1.01
	(0.15)	(0.20)	(0.16)	(0.21)	(0.20)	**(0.11)**	(0.18)	(0.08)	(0.07)	(0.07)	(0.08)	(0.10)	(0.08)	(0.14)	(0.07)
60	0.97	1.14	0.99	0.76	0.69	**0.55**	1.06	0.66	0.61	0.85	0.89	1.14	0.89	1.11	1.02
	(0.12)	(0.17)	(0.13)	(0.21)	(0.20)	**(0.09)**	(0.17)	(0.07)	(0.07)	(0.07)	(0.07)	(0.09)	(0.07)	(0.12)	(0.07)
70	0.92	1.07	0.93	0.73	0.68	**0.53**	0.99	0.60	0.57	0.82	0.87	1.08	0.87	1.10	1.01
	(0.10)	(0.13)	(0.11)	(0.19)	(0.18)	**(0.08)**	(0.18)	(0.05)	(0.05)	(0.07)	(0.06)	(0.07)	(0.06)	(0.09)	(0.07)
80	0.86	1.04	0.89	0.69	0.65	**0.50**	0.93	0.55	0.52	0.79	0.86	1.04	0.86	1.08	1.01
	(0.32)	(0.38)	(0.27)	(0.43)	(0.30)	**(0.17)**	(0.38)	(0.20)	(0.19)	(0.12)	(0.15)	(3.03)	(0.15)	(0.26)	(0.11)
100	0.77	0.98	0.83	0.67	0.65	**0.47**	0.89	0.50	0.48	0.74	0.83	1.00	0.83	1.06	1.01
	(0.25)	(0.35)	(0.23)	(0.32)	(0.27)	**(0.15)**	(0.29)	(0.17)	(0.16)	(0.11)	(0.14)	(0.52)	(0.14)	(0.24)	(0.10)
120	0.72	0.90	0.76	0.66	0.65	0.46	0.83	0.46	**0.44**	0.70	0.83	0.97	0.83	1.05	1.01
	(0.25)	(0.33)	(0.24)	(0.28)	(0.24)	(0.14)	(0.26)	(0.16)	**(0.14)**	(0.10)	(0.12)	(1.51)	(0.12)	(0.23)	(0.10)
150	0.63	0.83	0.69	0.65	0.64	0.44	0.80	0.42	**0.41**	0.64	0.80	0.93	0.80	1.02	1.01
	(0.20)	(0.25)	(0.19)	(0.26)	(0.22)	(0.13)	(0.24)	(0.13)	**(0.12)**	(0.09)	(0.10)	(0.75)	(0.10)	(0.19)	(0.08)
200	0.57	0.71	0.62	0.66	0.64	0.42	0.79	0.38	**0.37**	0.56	0.78	0.89	0.78	0.99	1.01
	(0.20)	(0.25)	(0.19)	(0.23)	(0.21)	(0.12)	(0.19)	(0.10)	**(0.10)**	(0.09)	(0.09)	(0.45)	(0.10)	(0.18)	(0.08)

Table 4: Relative Out-of-sample RMSE, DGP2: Quadratic

This table presents the average relative out-of-sample RMSEs of various models compared to the AR(1) benchmark, based on Monte Carlo simulations of the quadratic DGP across different sample sizes (T). Standard errors are computed from the distribution of 500 Monte Carlo draws. A lower RMSE indicates superior nowcasting performance. Bold values highlight the best-performing model for each time sample size.

	Standard Nowcasting Models							**ML Models**							
T	Bridge	UMIDAS	MIDAS	UMIDAS_PC	MIDAS_PC	TPRF	DFM	Lasso	Enet	SVRR	SVRL	SVRP	RF	DT	NNET
40	1.16	1.24	1.13	1.24	1.01	**0.82**	1.29	0.99	0.96	0.94	0.97	1.12	0.96	1.16	1.23
	(0.16)	(0.25)	(0.17)	(0.18)	(0.17)	**(0.13)**	(0.19)	(0.14)	(0.14)	(0.09)	(0.09)	(0.32)	(0.10)	(0.17)	(0.15)
50	1.13	1.22	1.11	1.11	0.95	**0.81**	1.23	0.96	0.93	0.93	0.96	1.15	0.95	1.15	1.21
	(0.15)	(0.19)	(0.15)	(0.16)	(0.15)	**(0.12)**	(0.17)	(0.12)	(0.12)	(0.07)	(0.08)	(0.14)	(0.08)	(0.15)	(0.12)
60	1.07	1.15	1.05	1.01	0.91	**0.79**	1.17	0.92	0.90	0.91	0.95	1.14	0.95	1.15	1.23
	(0.13)	(0.19)	(0.13)	(0.14)	(0.13)	**(0.11)**	(0.14)	(0.11)	(0.11)	(0.07)	(0.07)	(0.10)	(0.07)	(0.12)	(0.11)
70	1.02	1.15	1.02	0.96	0.89	**0.77**	1.10	0.89	0.88	0.89	0.93	1.05	0.93	1.13	1.24
	(0.12)	(0.16)	(0.13)	(0.12)	(0.12)	**(0.10)**	(0.12)	(0.10)	(0.10)	(0.07)	(0.06)	(0.09)	(0.06)	(0.10)	(0.09)
80	1.01	1.14	1.02	0.94	0.88	**0.75**	1.08	0.88	0.86	0.88	0.93	1.09	0.93	1.13	1.24
	(0.35)	(0.37)	(0.30)	(0.49)	(0.37)	**(0.22)**	(0.39)	(0.24)	(0.22)	(0.12)	(0.15)	(0.56)	(0.15)	(0.28)	(0.24)
100	0.96	1.13	0.99	0.90	0.85	**0.75**	1.03	0.85	0.83	0.85	0.90	1.01	0.90	1.10	1.24
	(0.32)	(0.40)	(0.27)	(0.36)	(0.28)	**(0.20)**	(0.34)	(0.20)	(0.20)	(0.10)	(0.13)	(0.94)	(0.13)	(0.25)	(0.22)
120	0.94	1.08	0.97	0.88	0.84	**0.73**	1.01	0.83	0.81	0.83	0.90	0.97	0.90	1.10	1.25
	(0.25)	(0.26)	(0.20)	(0.29)	(0.24)	**(0.17)**	(0.26)	(0.17)	(0.17)	(0.09)	(0.12)	(1.18)	(0.12)	(0.23)	(0.20)
150	0.88	1.06	0.93	0.85	0.83	**0.73**	0.96	0.81	0.79	0.81	0.88	0.94	0.88	1.06	1.25
	(0.20)	(0.27)	(0.18)	(0.23)	(0.19)	**(0.16)**	(0.22)	(0.15)	(0.15)	(0.09)	(0.10)	(0.39)	(0.10)	(0.20)	(0.18)
200	0.85	1.00	0.90	0.84	0.82	**0.71**	0.94	0.78	0.76	0.77	0.86	0.90	0.86	1.04	1.25
	(0.20)	(0.25)	(0.17)	(0.22)	(0.19)	**(0.14)**	(0.21)	(0.15)	(0.14)	(0.08)	(0.10)	(1.26)	(0.10)	(0.20)	(0.16)

Table 5: Relative Out-of-sample RMSE, DGP3: Non-Linear Data-Generating Process with Quadratic and Interaction Terms

This table presents the average relative out-of-sample RMSEs of various models compared to the AR(1) benchmark, based on Monte Carlo simulations of the quadratic and interactive DGP across different sample sizes (T). Standard errors are computed from the distribution of 500 Monte Carlo draws. A lower RMSE indicates superior nowcasting performance. Bold values highlight the best-performing model for each time sample size.

	Standard Nowcasting Models							**ML Models**								
T	Bridge	UMIDAS	MIDAS	UMIDAS_PC	MIDAS_PC	TPRF	DFM	Lasso	Enet	SVR	SVRL	SVRP	RF	DT	NNET
40	1.18	1.23	1.12	1.40	1.12	**0.96**	1.32	1.01	1.00	0.97	0.99	1.17	0.99	1.17	1.13
	(0.14)	(0.20)	(0.13)	(0.16)	(0.14)	**(0.13)**	(0.18)	(0.10)	(0.11)	(0.07)	(0.07)	(0.41)	(0.07)	(0.15)	(0.11)
50	1.15	1.21	1.11	1.25	1.08	**0.95**	1.26	1.01	1.00	0.96	0.97	1.12	0.97	1.18	1.12
	(0.13)	(0.19)	(0.13)	(0.13)	(0.12)	**(0.11)**	(0.15)	(0.09)	(0.10)	(0.06)	(0.06)	(0.23)	(0.07)	(0.15)	(0.10)
60	1.11	1.18	1.08	1.16	1.05	**0.94**	1.21	1.01	1.01	0.97	0.98	1.11	0.98	1.19	1.15
	(0.11)	(0.16)	(0.11)	(0.11)	(0.11)	**(0.10)**	(0.12)	(0.08)	(0.08)	(0.06)	(0.06)	(0.11)	(0.06)	(0.12)	(0.09)
70	1.09	1.14	1.06	1.09	1.01	**0.92**	1.16	0.99	0.98	0.95	0.97	1.16	0.97	1.17	1.13
	(0.08)	(0.13)	(0.09)	(0.10)	(0.09)	**(0.09)**	(0.10)	(0.07)	(0.07)	(0.05)	(0.05)	(0.08)	(0.05)	(0.09)	(0.08)
80	1.07	1.14	1.06	1.07	1.00	**0.92**	1.15	1.00	0.98	0.95	0.97	1.07	0.97	1.16	1.14
	(0.31)	(0.33)	(0.25)	(0.47)	(0.31)	**(0.21)**	(0.41)	(0.15)	(0.16)	(0.11)	(0.12)	(1.01)	(0.12)	(0.26)	(0.21)
100	1.04	1.11	1.04	1.02	0.97	**0.91**	1.09	0.97	0.97	0.94	0.96	1.05	0.96	1.15	1.14
	(0.26)	(0.32)	(0.22)	(0.36)	(0.28)	**(0.21)**	(0.33)	(0.15)	(0.15)	(0.08)	(0.11)	(0.65)	(0.11)	(0.25)	(0.17)
120	1.03	1.11	1.03	0.98	0.94	**0.88**	1.06	0.96	0.95	0.91	0.94	1.00	0.94	1.11	1.14
	(0.25)	(0.29)	(0.20)	(0.28)	(0.24)	**(0.18)**	(0.26)	(0.14)	(0.15)	(0.08)	(0.10)	(0.53)	(0.10)	(0.22)	(0.15)
150	1.00	1.09	1.02	0.96	0.93	**0.87**	1.02	0.94	0.93	0.90	0.94	0.96	0.94	1.10	1.14
	(0.21)	(0.24)	(0.16)	(0.22)	(0.20)	**(0.16)**	(0.22)	(0.14)	(0.13)	(0.08)	(0.09)	(1.36)	(0.09)	(0.20)	(0.13)
200	0.98	1.07	1.00	0.96	0.93	**0.88**	1.01	0.93	0.92	0.89	0.93	0.93	0.93	1.06	1.14
	(0.17)	(0.22)	(0.15)	(0.20)	(0.18)	**(0.14)**	(0.21)	(0.14)	(0.11)	(0.07)	(0.08)	(0.41)	(0.09)	(0.19)	(0.13)

or first-difference transformations to non-stationary variables, while stationary variables are used as-is in the analysis.

The data-cleaning process yields the final samples presented in Table 6. These six countries cover a broad range of characteristics: our sample contains a large number of high-frequency indicators for France, China, and Türkiye but fewer number from Vietnam, India, and Mauritius. GDP data of France and India are relatively long, those of China and Türkiye are moderately long, and those of Vietnam and Mauritius are short. Appendix 1 lists all the indicators utilized in the analysis for each country after the data-cleaning process is performed[15].

Table 6: Analysis Sample

This table presents the availability of the GDP data, the number of high-frequency indicators, and the sample periods during which both GDP and all the high-frequency indicators are available, for each country in our empirical analysis.

Countries	**Real GDP growth availability**	**Number of HF indicators**	**Complete sample**
China	1992Q1-2024Q2	90	2005Q3-2024Q2
France	1975Q1-2024Q2	77	2000Q2-2024Q1
India	1996Q2-2024Q1	15	2001Q2-2024Q1
Mauritius	2013Q1-2024Q1	13	2014Q2-2024Q1
Türkiye	1987Q1-2024Q1	55	2005Q2-2024Q1
Vietnam	2008Q1-2024Q1	26	2011Q3-2024Q1

We have input the same lagged dependent variables and all the high-frequency indicators across all the ML algorithms, ensuring that any performance differences are due to the models themselves, not the data. The selected machine learning algorithms can handle many regressors, even when the number of regressors (N) exceeds the time dimension (T).

The ML algorithms require converting the monthly indicators into the same frequency as real GDP growth (quarterly) to address the frequency mismatch. We consider two choices: first, convert monthly indicators to quarterly frequency through temporal aggregation (averaging, summing, or end-of-period); second, split each monthly indicator into three quarterly variables corresponding to each month within a quarter. While the second method increases dimensionality by a factor of three, the selected ML algorithms can handle this via built-in

[15] The sectoral coverage in Haver may be unbalanced, which may affect nowcast accuracy. For example, Haver's high-frequency data for Türkiye contain mostly goods indicators, with very few service-related indicators. Yet, the available service-related indicators seem highly relevant, such as tourism arrival, or consumer confidence.

regularization algorithms. Results for each of the ML algorithms are reported based on the lower error among the two frequency-conversion choices.

For the traditional econometric models, we consider all the high-frequency indicators as potential inputs and apply judgment to select variables in each model. Our model selection is guided by several considerations: first, choosing variables that demonstrate a high correlation with real GDP growth; second, referencing variables selected by the LASSO model as a starting point; and third, recursively adding and deleting variables to identify the subset yielding the lowest RMSE. Particularly, for the factor models (MIDAS-PC, UMIDAS-PC, DFM, 3PRF), we drop variables with low absolute correlation to real GDP growth. Most traditional econometric methods inherently determine the choice of frequency conversion, except for DFM and TPRF, where we opt to convert monthly indicators to quarterly using temporal aggregation.

5.2 Performance Evaluation

To evaluate predictive accuracy, we conduct a pseudo-real-time out-of-sample evaluation at the end of each quarter using an expanding rolling window[16]. This evaluation is illustrated in Figure 2.

We evaluate the models using an expanding rolling window from 2018Q1 to 2024Q1. This period encompasses both normal conditions and the COVID-19 episode, during which GDP nowcasting proved challenging. Each model is re-estimated every quarter upon the release of new GDP data[17]. Consequently, model coefficients vary quarterly and are exclusively used to generate nowcasts for each month within that quarter, mirroring actual nowcasting practices.

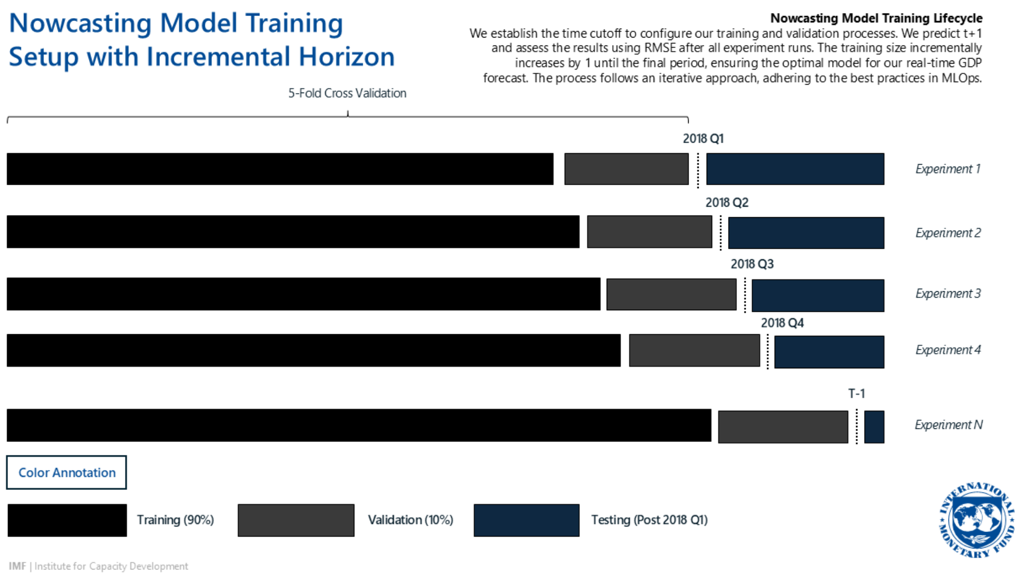
For simplification, we conduct pseudo real-time analysis by assuming high-frequency indicators are available at the end of each quarter prior to the release of real GDP for that quarter, rather than using actual data release dates. Additionally, we don't consider actual GDP revision, as the evaluation is psudo-real-time. There is no look-ahead bias, as

[16]In the expanding rolling window, GDP nowcasts could be updated more frequently—every month—when monthly indicators are released.

[17]This out-of-sample realistic evaluation differs from standard ML model evaluations for model selection, which typically involve splitting data into training and test sets based on a specified percentage (e.g., 80/20). This approach provides a more realistic evaluation of real-time nowcasting errors

Figure 2: Model Evaluation Timeline

This figure illustrates the model evaluation timeline using an expanding rolling window approach to compute the out-of-sample RMSE (OOS RMSE). This timeline closely mirrors the practical implementation of nowcasting. The OOS RMSE is calculated over the test period (2018Q1–2024Q2), during which each model is updated quarterly as new data become available.



all model's coefficients are estimated using previous-quarter data. Current quarter data are then inserted into the estimated models in order to derive nowcasts.

The collections of nowcasts are compared with the actual real GDP growth (in log). We report the OOS RMSEs relative to a univariate automatic ARIMA model [18].

5.3 Nowcast Evaluation Results

Figure 3 presents the relative RMSEs of each of the six countries, with raw RMSE reported in Table 7 Panel A. Consistent with our simulation findings, traditional econometric models tend to outperform their ML algorithms in most cases. Although TPRF is not the top performer, the best-performing models for four out of the six countries are Bridge and DFM. Bridge and DFM each ranks first in two countries. When DFM outperforms Bridge, the margin is substantial, whereas Bridge's outperformance over DFM is relatively modest. No-

[18]The Automatic ARIMA model is computed using the Box-Jenkins methodology, implemented via the "autoarima" package in R.

tably, Bridge models consistently perform well, ranking within the top four models in five of the six countries.

Linear ML algorithms are the best performers in two countries—Türkiye and Mauritius—and typically surpass non-linear ML models across all six countries[19]. Among ML methods, Elastic Net and Lasso perform best, except in India, where SVRL slightly outperforms Lasso by a narrow margin. In Türkiye, the outperformance of Elastic Net contrasts with Bolhuis and Rayner (2020), who favored Random Forest in their horserace, likely due to differences in high-frequency indicator inputs and sample length. The superior performance of Elastic Net with larger sample sizes aligns with our simulation results under the linear data-generating process (DGP 1).

Based on these six country cases, simpler models with fewer parameters tend to perform well, with the notable exception of DFM. Within ML algorithms, the linear methods—Lasso and Elastic Net—outperform their more complex counterparts. Even when a complex ML algorithm is preferred, as with SVRL in India, the basis function remains linear, suggesting limited nonlinearity or complex interactions in the underlying DGP.

We further evaluated all models under the shock and volatile COVID-19 episode during 2020Q1-2022Q4. Nowcasting during the COVID-19 crisis, illustrated in Figure 4 and Table 7 Panel B, proves challenging. In China, France, and Türkiye, traditional econometric models or linear ML algorithms outperform the autoARMA benchmark. However, in the other three countries, both classes of models fail to beat the benchmark. Commonalities of the three successful cases during COVID-19 are a large number of high-quality high-frequency indicators and a sufficiently long analysis sample. During this episode, TPRF and MIDAS perform best for China and France respectively, while Elastic Net performs the best for Türkiye. The difficulty in nowcasting during COVID-19 was also documented elsewhere. For instance, the Federal Reserve Bank of New York's temporary suspension of its external publication of the New York Fed Staff Nowcast.

We also evaluated ensemble forecasts among ML algorithms, examining three combinations: (1) LASSO-SVRP-NNET, (2) LASSO-RF-SVRR, and (3) LASSO-DT. While ensem-

[19]When evaluating the models for Turkiye during 2012Q1-2017Q4, where the macroeconomic environment and inflation were stable, the traditional econometric model performed the best. The top three best-performing models are Bridge, Random Forest, and DFM, respectively. In Mauritius, when evaluating models with a broader set of HF indicators available to the IMF Mauritius country team, over 2018Q1-2023Q2, the top three performing models are Lasso, Bridge, and Elastic Net, respectively.

ble methods often improve performance in other applications, here they do not outperform the best traditional econometric models or the linear ML algorithms in this context.

Overall, our findings support the use of simpler models – namely Bridge or Linear ML algorithms - for GDP nowcasting, alongside the more complex DFM model. Due to the limited number of observations, nonlinear ML algorithms risk overfitting the data, resulting in poorer nowcasting performance. Despite their simplicity, simple models reliably deliver superior nowcasting performance across the six country cases.

Figure 3: Out-of-sample Relative RMSEs, 2018Q1 to 2024Q1

This figure plots the out-of-sample (OOS) RMSEs relative to the AutoARIMA benchmark of an expanding rolling-window evaluation of a test set (2018Q1 to 2024Q1). Each model is re-estimated every quarter using data available up to the previous quarter. A ratio below one indicates that the model outperforms the AutoARIMA benchmark, with smaller values denoting better predictive performance.

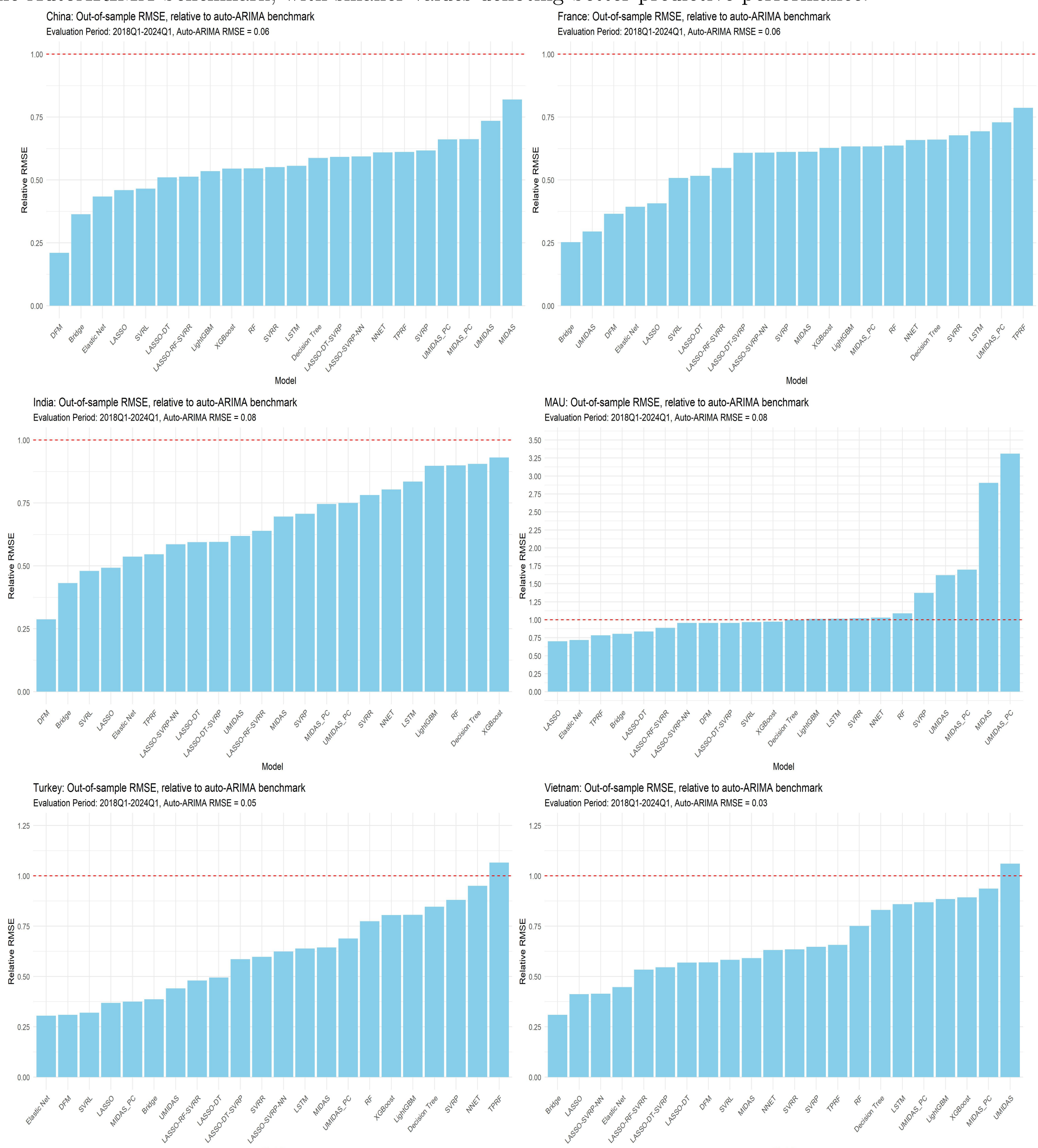
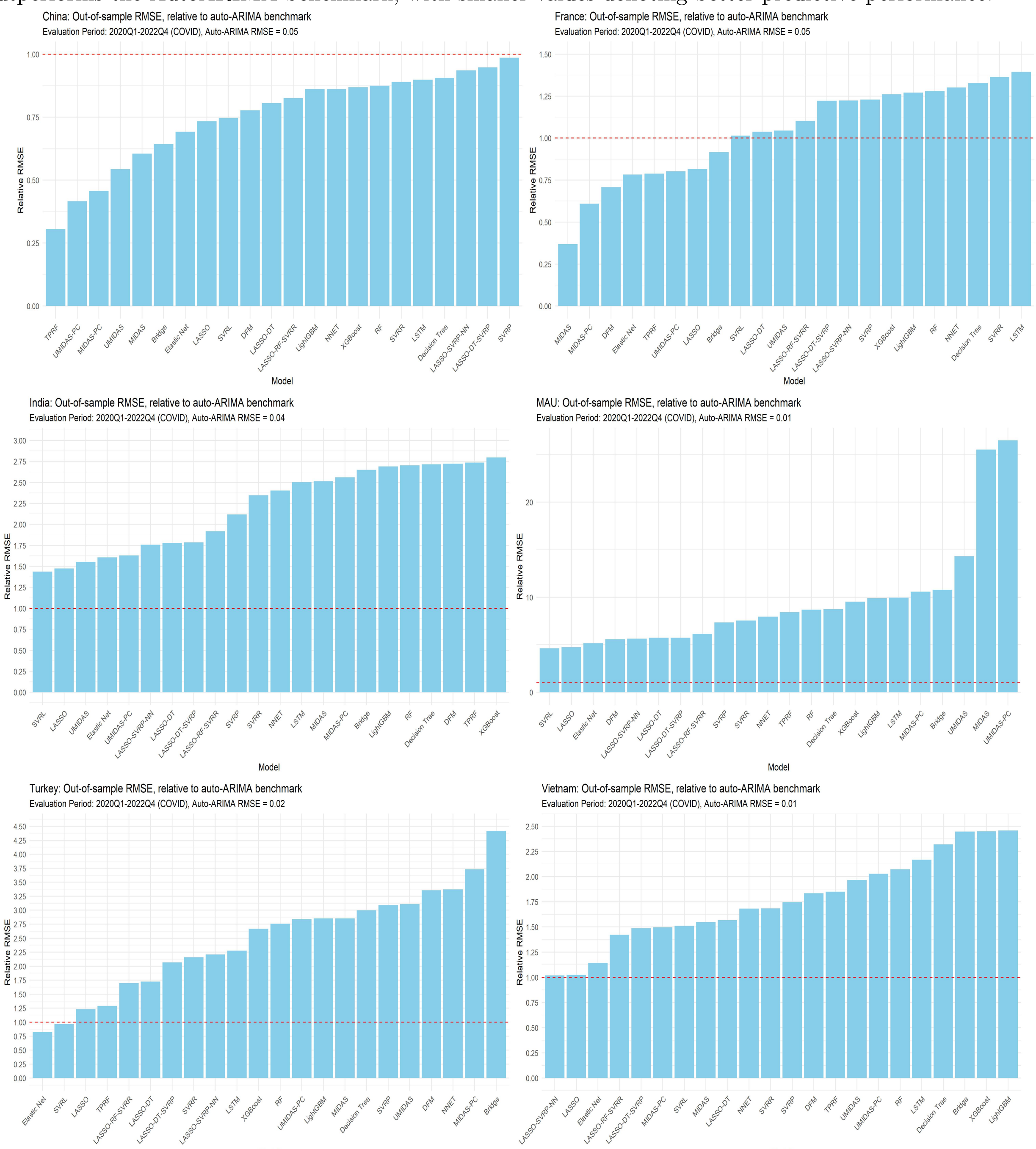


Figure 4: Out-of-sample Relative RMSEs during COVID-19

This figure plots the out-of-sample (OOS) RMSEs relative to the AutoARIMA benchmark of an expanding rolling-window evaluation during the COVID-19 period (2020Q1–2022Q4). Each model is re-estimated every quarter using data available up to the previous quarter. A ratio below one indicates that the model outperforms the AutoARIMA benchmark, with smaller values denoting better predictive performance.



6 Conclusion

Among all models used in nowcasting, our simulation and country cases reveal that traditional econometric models often deliver superior nowcasting performance to the ML algorithms. In our simulations, TPRF mostly outperforms others, losing only to Elastic Net model under the linear DGP and long data. In the six country applications, DFM or Bridge outperform ML algorithms in four out of the six countries. Among the ML algorithms, simpler ones – Lasso and Elastic Net – tend to outperform the more complex ones, and even surpass traditional econometric models in two out of the six countries.

These findings raise questions about the effectiveness of complex and non-linear ML algorithms for GDP nowcasting. In practice, the number of GDP observations are relatively short by machine learning standards, so small sample size can hamper the performance of complex models, increasing the risk of overfitting. Besides, the relationship between GDP growth and high-frequency indicators may be adequately captured by linear models. For these reasons, we advocate for the use of simpler models like Bridge, or linear ML algorithms, given their consistent superior performance in our analysis. Simpler models are less prone to overfitting, better suited for applications with a short time horizon. Among the complex models, DFM performs well, ranking first in two countries, with a substantial margin of outperformance when it does.

It is important to interpret our findings with caveats. The prevailing consensus in the literature—that there is no universally superior model —remains relevant. Consequently, model performance could vary for other countries or data sets. For instance, our analysis primarily relies on traditional macroeconomic data commonly used in economic surveillance; using non-traditional data such as satellite data might yield different results. Moreover, our study focuses on single-country time-series application, and does not extend to panel data setup.

Table 7: Raw Out-of-sample RMSE

This table reports the average out-of-sample (OOS) root mean squared errors (RMSEs) of various models over the full test period (2018Q1–2024Q2) in Panel A and the COVID period (2020Q1–2022Q4) in Panel B, based on an expanding-window nowcast evaluation. The upper section of each panel presents the raw OOS RMSEs for the benchmark AutoARIMA and traditional econometric models by country, while the lower section displays the corresponding results for machine learning algorithms and ensemble methods.

Panel A: All periods

	Benchmark	**Traditional Econometric Models**						
Country	AutoARMA	Bridge	UMIDAS	MIDAS	UMIDAS_PC	MIDAS_PC	TPRF	DFM
China	0.06	0.02	0.04	0.05	0.04	0.04	0.03	0.01
France	0.06	0.02	0.02	0.04	0.05	0.04	0.05	0.02
India	0.08	0.03	0.05	0.06	0.06	0.06	0.04	0.02
Mauritius	0.08	0.06	0.13	0.23	0.27	0.14	0.06	0.08
Türkiye	0.05	0.02	0.02	0.03	0.03	0.02	0.05	0.02
Vietnam	0.03	0.01	0.03	0.02	0.03	0.03	0.02	0.02

	Machine Learning Algorithms														
Country	DT	Elastic Net	LASSO	NNET	RF	SVRL	SVRP	SVRR	LightGBM	XGBoost	LSTM	LASSO-DT	LASSO-DT-SVRP	LASSO-RF-SVRR	LASSO-SVRP-NN
China	0.03	0.02	0.03	0.03	0.03	0.03	0.04	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
France	0.04	0.02	0.03	0.04	0.04	0.03	0.04	0.04	0.04	0.04	0.04	0.03	0.04	0.03	0.04
India	0.07	0.04	0.04	0.06	0.07	0.04	0.06	0.06	0.07	0.07	0.07	0.05	0.05	0.05	0.05
Mauritius	0.08	0.06	0.06	0.08	0.09	0.08	0.11	0.08	0.08	0.08	0.08	0.07	0.08	0.07	0.08
Türkiye	0.04	0.01	0.02	0.05	0.04	0.02	0.04	0.03	0.04	0.04	0.03	0.02	0.03	0.02	0.03
Vietnam	0.02	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.02	0.02	0.02	0.02	0.01

Panel B: COVID-19

	Benchmark	**Traditional Econometric Models**						
Country	AutoARMA	Bridge	MIDAS	UMIDAS	MIDAS-PC	UMIDAS-PC	TPRF	DFM
China	0.05	0.03	0.03	0.03	0.02	0.02	0.02	0.04
France	0.05	0.04	0.02	0.05	0.03	0.04	0.04	0.03
India	0.04	0.10	0.10	0.06	0.10	0.06	0.10	0.10
Mauritius	0.01	0.13	0.30	0.17	0.12	0.31	0.10	0.07
Türkiye	0.02	0.08	0.05	0.06	0.07	0.05	0.02	0.06
Vietnam	0.01	0.04	0.02	0.03	0.02	0.03	0.03	0.03

	Machine Learning Algorithms														
Country	DT	Elastic Net	LASSO	NNET	RF	SVRL	SVRP	SVRR	LightGBM	XGBoost	LSTM	LASSO-DT	LASSO-DT-SVRP	LASSO-RF-SVRR	LASSO-SVRP-NN
China	0.05	0.04	0.04	0.04	0.05	0.04	0.05	0.05	0.04	0.04	0.05	0.04	0.05	0.04	0.05
France	0.06	0.04	0.04	0.06	0.06	0.05	0.06	0.06	0.06	0.06	0.06	0.05	0.06	0.05	0.06
India	0.10	0.06	0.06	0.09	0.10	0.05	0.08	0.09	0.10	0.11	0.09	0.07	0.07	0.07	0.07
Mauritius	0.10	0.06	0.06	0.09	0.10	0.05	0.09	0.09	0.12	0.11	0.12	0.07	0.07	0.07	0.07
Türkiye	0.06	0.02	0.02	0.06	0.05	0.02	0.06	0.04	0.05	0.05	0.04	0.03	0.04	0.03	0.04
Vietnam	0.03	0.02	0.02	0.02	0.03	0.02	0.03	0.03	0.04	0.04	0.03	0.02	0.02	0.02	0.02

References

[1] Nesreen K. Ahmed, Amir F. Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29:594–621, 2010.

[2] Omer Faruk Akbal, Seung Mo Choi, Futoshi Narita, and Jiaxiong Yao. Panel nowcasting for countries whose quarterly gdps are unavailable, wp/23/158, august 2023. 8 2023.

[3] Martin Almuzara, Katie Baker, Hannah O'keeffe, and Argia Sbordone. The new york fed staff nowcast 2.0. 6 2023.

[4] Nikoleta Anesti, Simon Hayes, Andre Moreira, and James Tasker. Peering into the present: the bank's approach to gdp nowcasting. 2017.

[5] Elena Angelini, Gonzalo Camba-Méndez, Domenico Giannone, Gerhard Rünstler, and Lucrezia Reichlin. Short-term forecasts of euro area gdp growth. 10 2008.

[6] Milen Arro-Cannarsa and Rolf Scheufele. Nowcasting gdp: what are the gains from machine learning algorithms? 6 2024.

[7] Serkan Arslanalp, Seung Mo Choi, Parisa Kamali, Robin Koepke, Matthew Mcketty, Michele Ruta, Mario Saraiva, and Alessandra Sozzi. Nowcasting global trade from space. 5 2025.

[8] Andrii Babii, Ryan T. Ball, Eric Ghysels, and Jonas Striaukas. Panel data nowcasting: The case of price–earnings ratios. *Journal of Applied Econometrics*, 39:292–307, 3 2024.

[9] Andrii Babii, Eric Ghysels, and Jonas Striaukas. Machine learning time series regressions with an application to nowcasting. *Journal of Business and Economic Statistics*, 40:1094–1106, 2022.

[10] Alberto Baffigi, Roberto Golinelli, and Giuseppe Parigi. Bridge models to forecast the euro area gdp. *International Journal of Forecasting*, 20:447–460, 7 2004.

[11] Karim Barhoumi, Seung Mo Choi, Tara Iyer, Jiakun Li, Franck Ouattara, Andrew Tiffin, and Jiaxiong Yao. Overcoming data sparsity: A machine learning approach to track the real-time impact of covid-19 in sub-saharan africa. 5 2022.

[12] Marta Bańbura, Domenico Giannone, and Lucrezia Reichlin. Nowcasting. 12 2010.

[13] Marta Bańbura and Lorena Saiz. Short-term forecasting of euro area economic activity at the ecb. 6 2020.

[14] Venetia Bell, Lai Wah Co, Sophie Stone, and Gavin Wallis. Nowcasting uk gdp growth. 3 2014.

[15] Brandyn Bok, Daniele Caratelli, Domenico Giannone, Argia Sbordone, and Andrea Tambalotti. Macroeconomic nowcasting and forecasting with big data. 11 2017.

[16] Marijn A Bolhuis and Brett Rayner. Deus ex machina? a framework for macro forecasting with machine learning. 2 2020.

[17] Osmar Bolivar. Gdp nowcasting: A machine learning and remote sensing data-based approach for bolivia. *Latin American Journal of Central Banking*, 5, 9 2024.

[18] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[19] Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. *Classification and Regression Trees*. Taylor and Francis Group, 1984.

[20] Qiong Cao, Yihang Wu, Jia Yang, and Jing Yin. Greenhouse temperature prediction based on time-series features and lightgbm. *Applied Sciences (Switzerland)*, 13, 2 2023.

[21] Danilo Cascaldi-Garcia, Matteo Luciani, and Michele Modugno. Lessons from nowcasting gdp across the world. *International Finance Discussion Paper*, pages 1–37, 12 2023.

[22] Jennifer Castle, David Hendry, and Oleg Kitov. *Forecasting and Nowcasting Macroeconomic Variables: A Methodological Overview*. 2017.

[23] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 13-17-August-2016, pages 785–794. Association for Computing Machinery, 8 2016.

[24] Menzie Chinn, Baptiste Meunier, and Sebastian Stumpner. Nowcasting world trade with machine learning: a three-step approach. 6 2023.

[25] Michael P. Clements and Ana Beatriz Galvão. Macroeconomic forecasting with mixed-frequency data: Forecasting output growth in the united states. *Journal of Business and Economic Statistics*, 26:546–554, 9 2008.

[26] Michael P Clements and Ana Beatriz Galvão. Forecasting us output growth using leading indicators: An appraisal using midas models. *Journal of Applied Econometrics*, 24:1187–1206, 12 2009.

[27] Philippe Goulet Coulombe, Maxime Leroux, Dalibor Stevanovic, and Stéphane Surprenant. How is machine learning useful for macroeconomic forecasting? *Journal of Applied Econometrics*, 37:920–964, 8 2022.

[28] Jean-François Dauphin, Kamil Dybczak, Morgan Maneely, Marzie Taheri Sanjani, Nujin Suphaphiphat, Yifei Wang, and Hanqi Zhang. Nowcasting gdp - a scalable approach using dfm, machine learning and novel data, applied to european economies, wp/22/52, march 2022. Technical report, 2022.

[29] Catherine Doz, Domenico Giannone, and Lucrezia Reichlin. A two-step estimator for large approximate dynamic factor models based on kalman filtering. *Journal of Econometrics*, 164:188–205, 9 2011.

[30] ECB. Short-term forecasts of economic activity in the euro area. Technical report, ECB, 2008.

[31] Jairo Flores, Bruno Gonzaga, Walter Ruelas-Huanca, and Juan Tang. Nowcasting peruvian gdp with machine learning methods. 2024.

[32] Claudia Foroni and Massimiliano Marcellino. A survey of econometric methods for mixed-frequency data. 2 2013.

[33] Claudia Foroni and Massimiliano Marcellino. A comparison of mixed frequency approaches for nowcasting euro area macroeconomic aggregates. *International Journal of Forecasting*, 30:554–568, 2014.

[34] Claudia Foroni, Massimiliano Marcellino, Christian Schumacher, and Deutsche Bundesbank. *U-MIDAS: MIDAS regressions with unrestricted lag polynomials*. 2011.

[35] Saurabh Ghosh and Abhishek Ranjan. A machine learning approach to gdp nowcasting: An emerging market experience. *Bulletin of Monetary Economics and Banking*, 26:33–54, 2 2023.

[36] Authors Ghysels, Eric Santa-Clara, and Pedro Valkanov. The midas touch: Mixed data sampling regression models. 6 2004.

[37] Eric Ghysels, Virmantas Kvedaras, and Vaidotas Zemlys. Mixed frequency data sampling regression models: The r package midasr. *Journal of Statistical Software*, 72, 2016.

[38] Eric Ghysels, Virmantas Kvedaras, and Vaidotas Zemlys-Balevičius. *Mixed data sampling (MIDAS) regression models*, volume 42, pages 117–153. Elsevier B.V., 1 2020.

[39] Eric Ghysels, Arthur Sinko, and Rossen Valkanov. Midas regressions: Further results and new directions. *Econometric Reviews*, 26:53–90, 1 2007.

[40] Domenico Giannone, Lucrezia Reichlin, and Saverio Simonelli. Nowcasting euro area economic activity in real time: The role of confidence indicators. *National Institute Economic Review*, 10 2009.

[41] Domenico Giannone, Lucrezia Reichlin, and David Small. Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics*, 55:665–676, 5 2008.

[42] Niko Hauzenberger, Florian Huber, Karin Klieber, and Massimiliano Marcellino. Bayesian neural networks for macroeconomic analysis. *Journal of Econometrics*, 249, 5 2025.

[43] Dyna Heng, Fei Han, Sovanney Chey, Raksmey Uch, Dy Kuchsa, and Pholla Phork. Nowcasting and near-term forecasting cambodia's economy. 7 2024.

[44] Patrick Higgins. Gdpnow: A model for gdp "nowcasting". 7 2014.

[45] Alain Kabundi, Elmarie Nel, and Franz Ruch. Nowcasting real gdp growth in south africa. 2 2016.

[46] Dennis Kant and Andreas Pick. Nowcasting gdp using machine learning methods. Technical report, 2022.

[47] Bryan Kelly and Seth Pruitt. The three-pass regression filter: A new approach to forecasting using many predictors. In *Journal of Econometrics*, volume 186, pages 294–316. Elsevier Ltd, 6 2015.

[48] Gene Kindberg-Hanlon and Andrej Sokol. Gauging the globe: the bank's approach to nowcasting world gdp. Technical report, Bank of England Quarterly Bulletin, 9 2018.

[49] Vladimir Kuzin, Massimiliano Marcellino, and Christian Schumacher. Midas vs. mixed-frequency var: Nowcasting gdp in the euro area. *International Journal of Forecasting*, 27:529–542, 4 2011.

[50] Jan Linzenich and Baptiste Meunier. Nowcasting made easier: a toolbox for economists. 12 2024.

[51] Massimiliano Marcellino and Christian Schumacher. Factor midas for nowcasting and forecasting with ragged-edge data: A model comparison for german gdp. *Oxford Bulletin of Economics and Statistics*, 72:518–550, 8 2010.

[52] Preston J Miller and Daniel M Chin. Using monthly data to improve quarterly forecasts. Technical report, Federal Reserve Bank of Minneapolis, 4 1996.

[53] Andre Moreira. Macro technical paper series nowcasting gdp at the bank of england: a staggered-combination midas approach. Technical report, Bank of England, 6 2025.

[54] Takashi Nakazawa. Constructing gdp nowcasting models using alternative data constructing gdp nowcasting models using alternative data. 7 2022.

[55] Peter Rathjens and Russell P. Robins. Forecasting quarterly data using monthly information. *Journal of Forecasting*, 12:321–330, 1993.

[56] Adam Richardson, Thomas van Florenstein Mulder, and Tuğrul Vehbi. Nowcasting gdp using machine-learning algorithms: A real-time assessment. *International Journal of Forecasting*, 37:941–948, 4 2021.

[57] Barış Soybilgen and Ege Yazgan. Nowcasting us gdp using tree-based ensemble models and dynamic factors. *Computational Economics*, 57:387–417, 1 2021.

[58] James H Stock and Mark W Watson. Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, 12 2002.

[59] James H. Stock and Mark W. Watson. Twenty years of time series econometrics in ten pictures. In *Journal of Economic Perspectives*, volume 31, pages 59–86. American Economic Association, 3 2017.

[60] Juan Tenorio and Wilder Perez. Gdp nowcasting with machine learning and unstructured data. Technical report, 1 2024.

[61] Andrew Tiffin. Seeing in the dark: A machine-learning approach to nowcasting in lebanon. 2016.

[62] Qin Zhang, He Ni, and Hao Xu. Nowcasting chinese gdp in a data-rich environment: Lessons from machine learning algorithms. *Economic Modelling*, 122, 5 2023.

Appendix I

The following indicators serve as input data for the nowcasting models utilized in our analysis. We adopt the following notations: (dl) represents the log difference, while (d) denotes the first difference. All indicators are sourced from Haver Analytics. Although we abbreviate variable names for coding efficiency, each variable name remains intuitive and interpretable.

China

dl_cpi; dl_export_value; dl_export_volume; dl_import_value; dl_import_volume; dl_p_steel; dl_invest_real_estate; dl_fiscal_rev; dl_fiscal_exp; dl_yuan_usd; dl_m2; dl_m1; dl_ip_crude_oil; dl_ip_natural_gas; dl_ip_iron_ore; dl_ip_salt; dl_ip_yarn; dl_ip_cotton; dl_ip_gasoline; dl_ip_kerosene; dl_ip_diesel; dl_ip_coke; dl_output_sulfuric; dl_output_sodiumhy; dl_output_sodiumcar; dl_output_ethylene; dl_output_fertilizer; dl_output_chemical; dl_output_plastic; dl_output_synthetic; dl_ip_plastic; dl_output_cement; dl_output_plate; dl_output_pigiron; dl_output_steel; dl_output_steelprod; dl_output_nonferrous; dl_ip_aluminumo; dl_output_copper; dl_ip_aluminumprod; dl_output_boiler; dl_output_icecar; dl_output_metalcut; dl_output_electric_tool; dl_ip_packaging; dl_ip_cement_equip; dl_ip_smelt; dl_ip_feeding; dl_ip_tractor; dl_ip_airclean; dl_ip_railway; dl_ip_motor; dl_ip_ship; dl_ip_powergen; dl_ip_acmotor; dl_ip_electricity_prod; dl_output_fridge; dl_output_freezer; dl_output_ac; dl_output_wastemac; dl_ip_mobilehandset; dl_output_microcomputer; dl_output_semiconductor;

dl_output_electricmeasure; dl_output_copymac; dl_freight_traffic; dl_freight_turnover; dl_passenger_traffic; dl_passenger_turnover; dl_volume_port; dl_volume_foreigntradegoods; dl_no_expressmail; dl_no_cellphoneuser; dl_epu; d_fiscal_balance; d_neer; d_reer; d_pmi_manu; d_pmi_output; d_pmi_neworder; d_pmi_majorinput; d_pmi_emplotment; d_pmi_deliverytime; d_pmi_backlog; d_pmi_inputp; d_pmi_inputbuy; d_pmi_finishedgoods; d_pmi_newexport; d_pim_import; d_consumer_confidence.

France

dl_us_eur; dl_m1; dl_m2; dl_m3; dl_private_credit; dl_m1_dom; dl_m2_dom; dl_cb_asset; dl_saving; dl_unreg_saving; dl_credit_to_private; dl_loan_to_gov; dl_business_creation; dl_gov_rev; dl_gov_exp; dl_cpi; dl_core_cpi; dl_ppi; dl_imp_price; dl_brent_oil; dl_house_start; dl_house_permit; dl_no_job_seeker; dl_job_vacan; dl_job_vacan_dom; dl_job_vacan_skill; dl_job_vacan_unski; dl_hh_consumption; dl_export; dl_import; dl_num_dealth; dl_construction_creation; d_fis_bal; d_income_tax; d_corp_tax; d_vat; d_trade_bal; d_busness_climate_indicator; d_epu; d_manufacture_survey; d_service_survey; d_construction_survey; d_retail_survey; d_wholesale_survey; d_employment_survey; d_hh_confidence_indicator; d_retail_sale_survey; d_industry_sentiment; d_service_sentiment; d_cap_utilize_survey; r_tbill_1m; r_tbill_3m; r_tnote_1yr; r_bond_10yr; r_saving; r_industrial; r_housing; r_newloan; turnover_indus; turnover_service; retail_sales; turning_point_survey; industry_turn_survey; service_turn_survey; construction_turn_survey; wholesale_turn_survey; surprise_indicator; unemployment_t12_survey; live_standard_p12; current_order_book; d_foreign_order_survey; d_total_order_survey; d_employment_survey_m; emp_forecast_survey; d_production_survey; production_forecast_survey; inventory_survey.

India

dl_ex_sugar; dl_ex_spices; dl_ex_rice; dl_cpi; dl_ecma; dl_agri; dl_consu_petro_coke; dl_consu_gas; dl_consu_motor; dl_consu_diesel; dl_consu_petro; d_pmi_new; d_ip; reverse_repo; bank_rate.

Mauritius

dl_cpi; dl_stock_semdex; dl_stock_demex; dl_stock_semtri; dl_stock_demtri; dl_stoc_sem10; dl_gross_reserve; dl_fx_euro; dl_fx_gbp; dl_fx_usd; dl_fx_rand; repo; d_tourist_arr.

Türkiye

dl_ppi_int; dl_ppi_dur; dl_ppi_ene; dl_ppi_cap; dl_ppi_mine; dl_ppi_manu; dl_ppi_f; dl_tot_veh_prod; dl_tot_auto_prod; dl_gross_electricity; dl_export_oil; dl_export_gem; dl_export_capital; dl_export_interm; dl_export_consu; dl_export_auto; dl_cpi; dl_cpi_f; dl_cpi_h; dl_cpi_g; dl_cpi_e; dl_cpi_s; dl_ppi; dl_p_gold; dl_ipi; dl_ipi_mine; dl_ipi_manu; dl_ipi_gas; dl_ipi_food; dl_ipi_gar; dl_ipi_electronic; dl_ipi_machine; dl_ipi_car; dl_motor_sale; dl_commercial_vehicle_sale; dl_tourism_arr; dl_cli; dl_consu_confi; dl_export_fish; dl_export_mine; dl_export_manu; dl_neer; dl_reer_cpi_jpm; dl_reep_ppi_jpm; dl_reer_cpi_cb; dl_reer_cpi_developed; dl_reer_cpi_developing; dl_try_us_avg; dl_stock_index_avg; dl_int_reserve; dl_cb_reserve_asset; dl_fx_reserve_commerical; dl_m3; i_consu; i_deposit.

Vietnam

dl_retail_sales; dl_im; dl_cpi; dl_powder_milk; dl_stock_index; dl_int_visitor; dl_reer; dl_liquid_gas; dl_been; dl_seafood; dl_cig; dl_paint; dl_tv; dl_cement; dl_motorbike; dl_garment; dl_coal; dl_polyester; dl_electricity; dl_water; dl_cotton; dl_retail_service; d_pmi_manu_cap; d_pmi_manu; d_pmi_manu_new; d_pmi_manu_new_ex.

PUBLICATIONS

GDP Nowcasting Performance of Traditional Econometric Models vs Machine-Learning Algorithms: Simulation and Case Studies

Working Paper No. WP/2025/252