Machine Learning-Based Estimation of Monthly GDP*

Yonggeun Jung†

This Version: June, 2025

Abstract

This paper proposes a scalable framework to estimate monthly GDP using machine learning methods. We apply Multi-Layer Perceptron (MLP), Long Short-Term Memory networks (LSTM), Extreme Gradient Boosting (XGBoost), and Elastic Net regression to map monthly indicators to quarterly GDP growth, and reconcile the outputs with actual aggregates. Using data from China, Germany, the UK, and the US, our method delivers robust performance across varied data environments. Benchmark comparisons with prior US studies and UK official statistics validate its accuracy. The approach offers a flexible and data-driven tool for high-frequency macroeconomic monitoring and policy analysis.

JEL Codes: C53, C55, E01

Keywords: Monthly GDP, Machine Learning, Temporal Disaggregation.

*Replication package and data are available at https://github.com/Yonggeun-Jung/monthly_gdp.

†Ph.D. Student in Economics. Department of Economics, University of Florida, PO Box 117140, Gainesville, Florida, USA, 32611. Email: yonggeun.jung@ufl.edu.

1 Introduction

Real gross domestic product (GDP) is a core economic indicator, widely valued for its intuitive and comprehensive representation of economic activity within a given region. While GDP is traditionally published at a quarterly frequency, the increasing demand for timely and granular economic insights has led to growing interest in both nowcasting and high-frequency estimation (Chu & Qureshi, 2023; Ghosh & Ranjan, 2023; Giannone et al., 2008; Jansen et al., 2016; Mitchell et al., 2005).

In particular, the expanding availability of high-frequency data across various domains has intensified efforts to estimate monthly GDP (Brave et al., 2019; Koop et al., 2021, 2023; Mariano & Murasawa, 2003, 2010; Mitchell et al., 2005). Monthly estimates offer distinct advantages: they allow for more granular policy interventions and enable researchers to capture short-run dynamics with greater precision.[1]

To address this gap, Koop et al. (2023) developed a Bayesian Mixed-Frequency Vector Auto-Regressive (MF-VAR) model that integrates quarterly GDP with a range of monthly indicators to estimate monthly GDP in the US. Their approach offers high accuracy in identification and inference, particularly under a well-specified prior. However, its Bayesian structure can pose implementation challenges, especially when applied to countries with limited prior information or less standardized data environments.

In response to these limitations, this study proposes an alternative framework based on machine learning, including deep learning techniques. Our goal is to develop a method for monthly GDP estimation that is both scalable and applicable across a wide range of country contexts, without relying on strong prior assumptions about a country's economic structure. Specifically, we employ four distinct classes of models—a Multi-Layer Perceptron (MLP), a Long Short-Term Memory (LSTM) network, XGBoost, and Elastic Net—to learn the relationship between quarterly GDP growth and monthly economic indicators. MLP is well-suited for capturing nonlinear relationships, while LSTM leverages its ability to retain information from past observations. In contrast to these neural network–based models, XGBoost and Elastic Net offer greater interpretability with respect to the explanatory variables.

After a quarterly model is trained, it is applied to the monthly indicators to generate a preliminary high-frequency signal of GDP growth. This signal is then

[1] Since monthly GDP is often difficult to observe directly, practitioners may rely on proxy indicators. However, such proxies can be biased depending on the structural characteristics of the economy.

adjusted using the Denton (1971) proportional benchmarking method to ensure that the resulting monthly estimates are consistent with the observed quarterly GDP.[2] In doing so, the approach preserves intra-quarter dynamics implied by the machine learning model while maintaining consistency with national accounts.

This paper makes several key contributions. First, it offers a scalable and generalizable framework. The proposed method is successfully applied to four countries with diverse data environments—China, Germany, the UK, and the US—demonstrating its adaptability beyond a single-country context.

Second, it adopts a data-driven modeling approach. By leveraging deep learning models capable of capturing complex nonlinear relationships, the framework allows the data to uncover hidden patterns without relying on rigid structural assumptions.[3]

Third, the framework addresses interpretability. Recognizing the "black box" limitations of neural networks, the study also incorporates XGBoost and Elastic Net—models that provide transparency through variable importance scores and coefficient estimates. This provides interpretive insights into which indicators are most associated with GDP growth, and allows cross-validation of results across different model classes.

Finally, the paper provides empirical validation. The proposed methodology is rigorously benchmarked against monthly estimates from Koop et al. (2023) for the US and the official monthly GDP series from the UK's Office for National Statistics (ONS), reinforcing the credibility of the results.

The remainder of the paper is organized as follows. Section 2 describes the methodological framework, including model structure and data preparation procedures. Section 3 presents estimation results, including out-of-sample test performance and benchmarking against existing monthly GDP series for the US and UK. Section 3 also discusses model interpretability using XGBoost and Elastic Net, highlighting cross-country insights and robustness checks. Section 4 concludes with directions for future research. Additional data and model descriptions, algorithmic details, reconciliation procedures, and supplementary figures are provided in the appendices.

---

[2]However, the application of such methods inherently involves adjusting model outputs using already published quarterly GDP figures, which limits their usefulness for another important task: nowcasting.

[3]See Chow and Lin (1971) and Litterman (1983) for traditional temporal disaggregation methods: Chow and Lin (1971) propose a regression-based approach, while Litterman (1983) uses a state-space variant. These methods typically rely on auxiliary regressors and impose dynamic structures to interpolate or extrapolate monthly values. While powerful, they require strong assumptions about variable relationships and are not directly applicable for reconciling model-generated high-frequency signals with official aggregates.

2 Methodology

2.1 Model

This section describes the model[4] employed to estimate monthly GDP. Let Y_q denote the observed quarterly real GDP growth for quarter q, and let $X_m \in \mathbb{R}^k$ represent the vector of k explanatory variables observed in month m. Typical explanatory variables include industrial production, retail sales, the unemployment rate, consumer sentiment, and financial market indicators.[5]

To relate these high-frequency monthly indicators to the observed quarterly GDP growth, we first aggregate the monthly variables to construct quarterly-level variables $X_q \in \mathbb{R}^k$. For each variable j, the quarterly value $X_q^{(j)}$ is constructed as follows:

(i) For variables measured in levels (e.g., the unemployment rate), we compute the quarterly average:

$$X_q^{(j)} = \frac{1}{3} \sum_{m \in q} X_m^{(j)}. \tag{1}$$

(ii) For variables expressed as growth rates (e.g., first differences of logarithms), we compute the quarterly sum, consistent with the cumulative growth identity:

$$X_q^{(j)} = \sum_{m \in q} X_m^{(j)}. \tag{2}$$

In practice, we first determine which variables require log-differencing based on the Augmented Dickey-Fuller (ADF) test (Said & Dickey, 1984). After constructing the differenced series and aggregating all variables to a quarterly frequency, we split the dataset into training and test sets. To prevent data leakage, we standardize the non-log-differenced level variables only after this split.[6] Specifically, we apply Z-score standardization.[7]

[4]The algorithms used to implement this model are detailed in Appendix C.

[5]For example, Koop et al. (2023) use Average Weekly Hours: Manufacturing, CPI, Industrial Production, Real Personal Consumption Expenditures, the Federal Funds Rate, the 10-Year Treasury Rate, the S&P Stock Price Index, and the Civilian Unemployment Rate.

[6]Standardizing the entire dataset before splitting could allow information from the test set—such as its mean and standard deviation—to leak into the training process.

[7]Each variable X_j is transformed into z_j as follows:

$$z_j = \frac{X_j - \mu_j}{\sigma_j},$$

where μ_j and σ_j denote the sample mean and standard deviation of variable X_j.

We specify a general regression framework to model the relationship between quarterly GDP growth and the aggregated explanatory variables:

$$Y_q = f(X_q; \theta) + e_q, \tag{3}$$

where $f(\cdot)$ denotes the potentially nonlinear function to be estimated, θ represents the set of model parameters, and e_q is the error term. To approximate the unknown function f, we consider four distinct classes of machine learning models that vary in flexibility, structure, and interpretability.

First, we employ a **Multi-Layer Perceptron (MLP)**, a class of feedforward artificial neural networks capable of capturing complex nonlinear relationships (Rumelhart et al., 1986). Second, to account for the time-series nature of the data, we utilize a **Long Short-Term Memory (LSTM)** network, a type of recurrent neural network (RNN) designed to capture long-range dependencies through gated memory cells (Hochreiter & Schmidhuber, 1997). Third, we include **XGBoost** (Extreme Gradient Boosting), a high-performance implementation of the gradient boosting framework that combines an ensemble of decision trees (T. Chen & Guestrin, 2016). In addition to its strong predictive accuracy, XGBoost provides variable importance scores, which help alleviate the interpretability limitations of neural networks and allow for cross-model robustness checks. Finally, as a linear benchmark, we consider the **Elastic Net**, a regularized linear regression model that combines LASSO (λ_1) and Ridge (λ_2) penalties (Zou & Hastie, 2005). Its linear structure enables direct interpretation of each explanatory variable's contribution.[8] [9]

Once the quarterly model $f(\cdot; \hat{\theta})$ is estimated, we generate preliminary monthly estimates by applying it to the original monthly covariate vectors X_m:

$$\tilde{Y}_m = f(X_m; \hat{\theta}). \tag{4}$$

Importantly, since the model is trained on quarterly data, each $\tilde{Y}_m$ reflects a prediction of the *total quarterly* growth rate, conditional on the economic environment observed in month m. The resulting series $\{\tilde{Y}_m\}$ thus serves as a high-frequency, model-based signal that captures intra-quarter variation as inferred from monthly movements in the explanatory variables.

[8] LASSO (Least Absolute Shrinkage and Selection Operator) regularization (λ_1 penalty) performs variable selection by shrinking some coefficients to zero, while Ridge regularization (λ_2 penalty) shrinks coefficients continuously without eliminating any (James et al., 2023).

[9] Detailed model specifications are provided in Appendix B.

Because this series does not, by construction, sum to the observed quarterly total, we apply a proportional benchmarking adjustment, following the approach of Denton (1971).[10] For each quarter q, we compute an adjustment factor k_q, defined as the ratio of the actual quarterly growth rate to the sum of the preliminary monthly estimates within that quarter:

$$k_q = \frac{Y_q}{\sum_{j \in q} \tilde{Y}_j}. \tag{5}$$

We then obtain the adjusted monthly growth rates, $\hat{y}_m$, by scaling each $\tilde{Y}_m$ using the corresponding adjustment factor:[11]

$$\hat{y}_m = k_q \cdot \tilde{Y}_m. \tag{6}$$

This procedure ensures that the adjusted monthly estimates are consistent with the observed quarterly aggregates—i.e., $\sum_{m \in q} \hat{y}_m = Y_q$—while preserving the intra-quarter variation implied by the model. However, since the model output reflects a projection of total quarterly growth conditional on monthly information, rather than a direct estimate of each month's contribution, the resulting high-frequency series should be interpreted as a model-based, internally consistent approximation.

2.2 Data

For the empirical analysis, we collect data for four countries—China, Germany, the UK, and the US. While our target sample spans from 1991 to 2024, the actual coverage varies by country depending on the availability of reliable quarterly GDP figures and monthly indicators. In all cases, we ensure that the quarterly GDP series and all explanatory variables are temporally aligned within a consistent time frame for each country.

[10] See technical details in Appendix D.

[11] In our implementation, we first derive a naive monthly estimate ($\hat{p}_m$) by dividing the preliminary quarterly-scale estimate ($\tilde{Y}_m$) by 3. This scaling factor, however, mathematically cancels out in the final adjustment formula, yielding the same result as the more concise expression presented in Equations (5) and (6). For the detailed algorithm, please see Appendix C.

Table 1. Summary of Datasets (Sources)

Variable	China	Germany	UK	US
CPI	FRED	FRED	FRED	FRED
Retail Trade		FRED		
M1 Money Stock			BOE	FRED
M2 Money Stock				FRED
Total Reserves	FRED	FRED	FRED	
Exchange Rate	FRED		FRED	
Unemployment Rate		FRED	ONS	FRED
Employment Rate			ONS	
Labor Force				FRED
Avg. Weekly Hours				FRED
Industrial Production				FRED
Production Volume		FRED	FRED	
Manufacturing		FRED		
PPI	FRED			
Real Personal Consum.				FRED
Interest Rate (10Y Gov)			FRED	
Interest Rate (Gov Bond)			FRED	
Interest Rate (10Y Trea.)				FRED
Effect. Fund Rate				FRED
Effect. Ex. Rate	FRED	FRED		
S&P/ FTSE / DAX / SSEC	IV	YF	YF	YF
Share Prices			FRED	
Moody's AAA Bond				FRED
(Net) Exports	FRED	FRED	FRED	FRED
Imports	FRED	FRED	FRED	
US Imports from China	FRED			
Price Competitiv.		DB		
Policy Uncertainty	FRED	FRED		
Personal Savings Rate				FRED
Nonfarm Payroll Emp.				FRED
Real GDP (Quarterly)	FRED	FRED	FRED	FRED
Total Variables Used	11	13	14	16
Data Span	01.94-09.23	01.91-12.23	01.91-03.24	01.92-12.24

Note: The entries in the table indicate the primary data source used to fetch each variable. FRED: Federal Reserve Economic Data (https://fred.stlouisfed.org/); YF: Yahoo Finance (https://finance.yahoo.com/); ONS: Office for National Statistics (https://www.ons.gov.uk/); BOE: Bank of England (https://www.bankofengland.co.uk/boeapps/database/); DB: Deutsche Bundesbank (https://www.bundesbank.de/en); IV: Investing.com (https://www.investing.com/). They may serve as a data platform; the original source of each series may differ by variable.

Quarterly real GDP growth is calculated as the log difference of seasonally adjusted real GDP. The monthly explanatory variables cover various dimensions of the economy—including consumption, production, investment, capital markets, labor, and trade—subject to each country's data availability. These variables are collected from central banks, capital market authorities, national statistical offices, and other third-party providers.

While the specific variables—and even the presence of certain categories—vary across countries, we select variables flexibly within and across categories to maximize relevance and comparability. In some cases, certain economic dimensions (e.g., investment or trade) may be missing or sparsely represented, while others may be relatively rich in data. Table 1 summarizes the datasets used for each country, with detailed descriptions provided in Appendix A.

For the US, monthly GDP growth estimates produced by Koop et al. (2023) are available,[12] while the UK's ONS publishes official monthly GDP figures.[13] These two cases allow us to benchmark the performance of our framework. For the remaining countries, we demonstrate that the model performs robustly even when the set of available indicators differs.

3 Estimation Results

3.1 Estimated Monthly GDP by Country

As discussed in Section 2.1, a key starting point of our framework is its ability to accurately predict quarterly GDP growth, $\tilde{Y}_q$, from the transformed quarterly covariates X_q. Table 2 summarizes the test performance of each model across the four countries, using a hold-out test set. Notably, the test period includes the COVID-19 era (2020–2022), enabling an evaluation of how well each model captures large, unexpected economic shocks.[14]

Model performance is assessed using multiple evaluation metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R^2, Sign Accuracy, Symmetric Mean Absolute Percentage Error (sMAPE), and Theil's U1. Lower values indicate better performance for MSE, RMSE, MAE, sMAPE, and Theil's U1, while higher values are preferred for R^2 and Sign Accuracy.

[12] Koop et al. (2023) provide both the replication package and data, which are available on the journal's website: https://www.tandfonline.com/doi/full/10.1080/07350015.2022.2044336.

[13] Available at https://www.ons.gov.uk/economy/grossdomesticproductgdp.

[14] See Appendix F for time series plots by country.

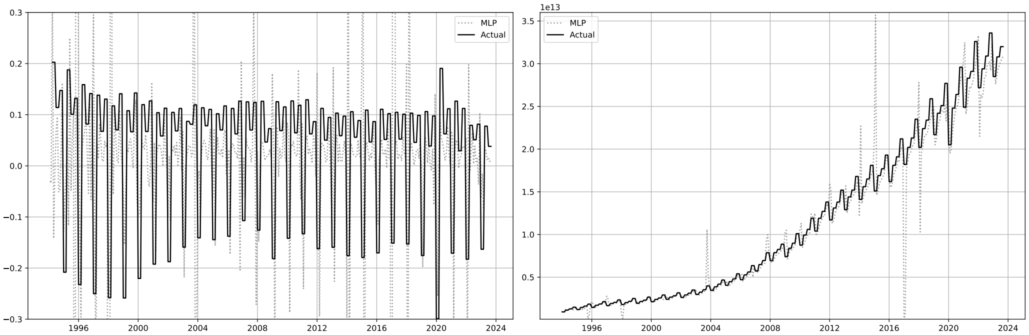
Table 2. Test Performance

Country	Model	MSE	RMSE	MAE	R^2	Sign Acc.	sMAPE	Theil U1
China	MLP	0.006	0.077	0.060	0.628	0.875	78.218	0.335
	LSTM	0.001	0.033	0.026	0.936	0.950	31.807	0.132
	XGB	0.011	0.107	0.084	0.289	0.750	99.990	0.542
	ENET	0.011	0.106	0.091	0.299	0.750	134.608	0.636
Germany	MLP	0.000	0.018	0.011	0.507	0.667	120.109	0.425
	LSTM	0.000	0.021	0.014	0.432	0.435	162.949	0.536
	XGB	0.001	0.024	0.012	0.121	0.741	104.188	0.740
	ENET	0.001	0.025	0.013	0.041	0.667	123.126	0.724
UK	MLP	0.001	0.030	0.017	0.712	0.815	115.059	0.314
	LSTM	0.003	0.057	0.025	0.086	0.696	134.470	0.852
	XGB	0.003	0.053	0.021	0.072	0.741	114.271	0.858
	ENET	0.003	0.054	0.022	0.030	0.556	137.120	0.900
US	MLP	0.000	0.009	0.008	0.826	0.889	79.447	0.203
	LSTM	0.000	0.017	0.011	0.529	0.739	95.018	0.427
	XGB	0.000	0.021	0.009	0.165	0.926	67.349	0.668
	ENET	0.000	0.014	0.008	0.600	0.889	102.448	0.391

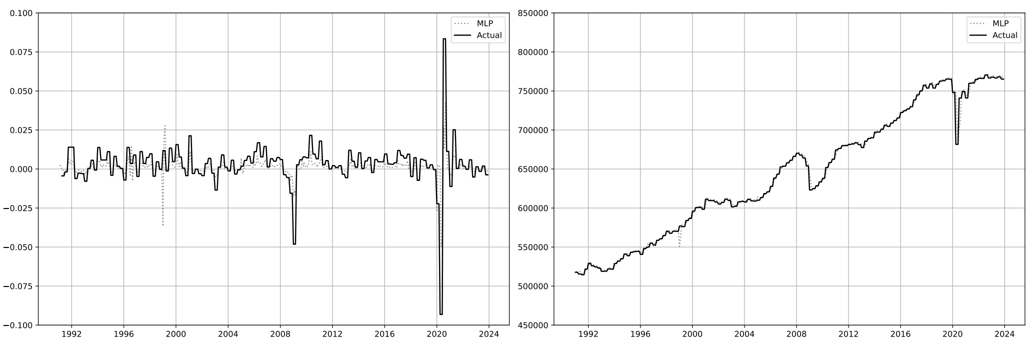
Note: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and symmetric Mean Absolute Percentage Error (sMAPE, %) are all scale-dependent error metrics where lower values indicate better predictive accuracy. The coefficient of determination (R^2) measures the proportion of variance explained by the model and is better when closer to one. Sign Accuracy indicates how often the model correctly predicts the direction of change and is also better when higher. Theil's U1 statistic compares the forecast accuracy of a model to that of a naive (no-change) forecast; values closer to zero indicate better performance.

Among the models, the MLP delivers the most accurate point forecasts overall, generally yielding the lowest MSE and highest R^2, though the LSTM model shows superior performance for China. XGBoost, while exhibiting higher errors, performs relatively well in predicting the direction of change (high sign accuracy).

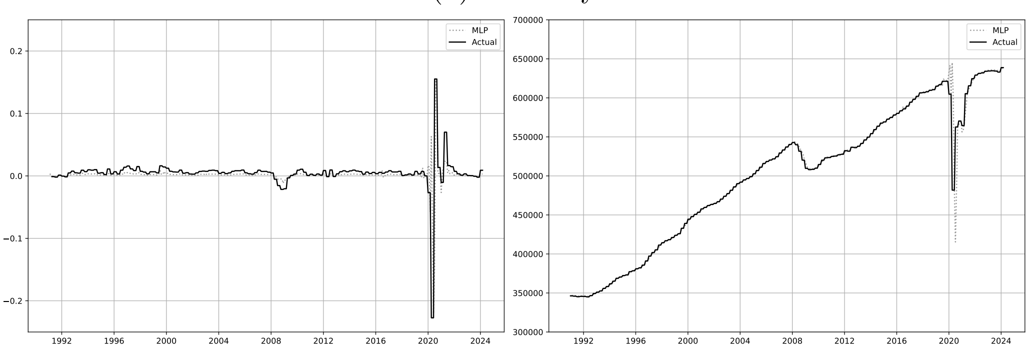
LSTM, despite its theoretical advantage in modeling sequential dependencies, underperforms in this setting, except for China. One strength of LSTM is its ability to learn from long sequences of past observations, potentially capturing persistent dynamics. However, in this application, the model must be trained on quarterly GDP targets using sequences of monthly covariates. As the input sequence length t increases, information loss becomes more likely or significant, which introduces a trade-off between capturing long-run dependencies and preserving data quality. In contrast, MLPs are not affected by this sequence alignment issue and thus appear more suitable for the structure and size of the current dataset.



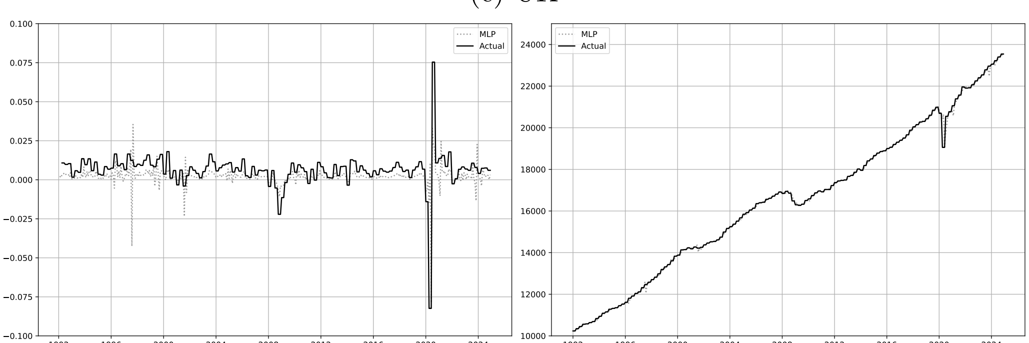
(a) China



(b) Germany



(c) UK



(d) US

Figure 1. Best-Performance Models (MLP): Growth (L) and Level (R)

Across all countries, the MLP model demonstrates relatively stable and robust performance, successfully capturing general trends in monthly GDP—even during periods of economic shocks. In contrast, alternative models such as XGBoost and Elastic Net tend to produce larger outliers during shock periods, particularly in countries where fewer explanatory variables are available.[15] Therefore, while the MLP is preferred for accurate prediction, models like XGBoost or Elastic Net may offer additional insights when interpretability or variable importance is of interest. While they may not outperform MLPs in terms of prediction accuracy, their transparency can help uncover which indicators drive model predictions.

Finally, the Elastic Net model performs the worst in terms of point forecasting accuracy, likely due to its inherent linearity and limited flexibility in capturing nonlinear relationships. These limitations become more pronounced in some countries—especially China—where the lack of rich data further constrains model performance.

Interestingly, while the LSTM model achieves the best out-of-sample performance for China at the quarterly level, its monthly predictions tend to show relatively large deviations from the observed values. This pattern is unlikely to be a result of overfitting, given the model's strong test performance. Rather, it may reflect other factors, such as the challenges of mapping monthly variables to a quarterly variable, limitations in the availability and quality of China's data, or more fundamentally, concerns about the reliability of the official GDP statistics themselves. To further explore this possibility, we re-estimate the MLP model using the OECD's Composite Leading Indicator (CLI) as an alternative dependent variable. See Appendix E for details.

3.2 Benchmark: US and UK Cases

3.2.1 US Case: Koop et al. (2023)

Koop et al. (2023) construct monthly GDP estimates by embedding a Bayesian MF-VAR model that incorporates quarterly expenditure-side (GDP_E) and income-side (GDP_I) GDP, a latent true GDP process, and multiple monthly indicators of economic activity. The model treats GDP_E and GDP_I as noisy observations of true GDP and enforces temporal aggregation constraints to reconcile monthly and quarterly data. Identification is achieved using either instrumental variables (e.g., unemployment) or set-identification via variance bounds.

[15] See Appendix F.

Figure 2. Benchmark to Koop et al. (2023): US Estimated Monthly GDP

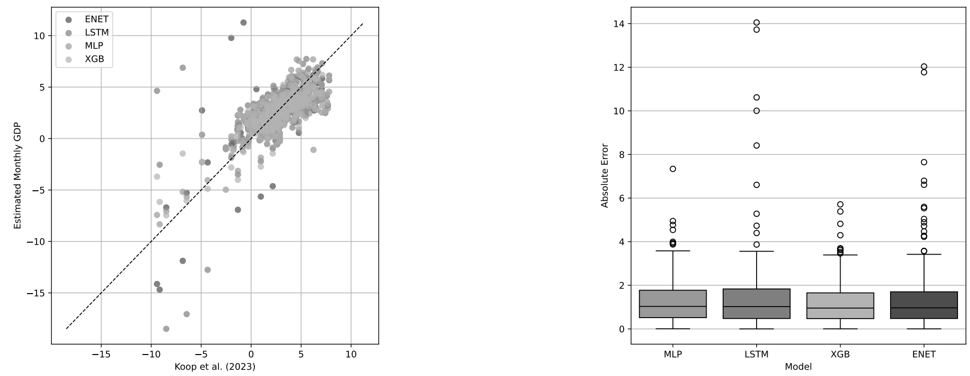
Following Koop et al. (2023)—and building on the methodology of Mariano and Murasawa (2003, 2010) and Mitchell et al. (2005)—we transform our model's final monthly growth rate $\hat{y}_m$ into an annualized rate for comparison using the same five-month weighted average formula:

$$\hat{y}^A_{m,t} = \left(\frac{1}{3}\hat{y}_{m,t} + \frac{2}{3}\hat{y}_{m,t-1} + \hat{y}_{m,t-2} + \frac{2}{3}\hat{y}_{m,t-3} + \frac{1}{3}\hat{y}_{m,t-4}\right) \times 4 \times 100 \tag{7}$$

While Koop et al. (2023) provide estimates from 1960 to 2019, our estimation spans from 1992 to 2024. For comparability, we focus on the overlapping period from 1992 to 2019. As shown in Figure 2, our estimates broadly track the benchmark series over time. Despite some differences in magnitude and timing, especially around turning points, the overall trajectory—including during the 2008 Great Recession—is notably similar.

To quantify the differences, Figure 3 shows a scatter plot (left) and error distribution (right), comparing our estimates to the benchmark. Across all models, approximately 75% of the errors lie within a 2 percentage point range. However, larger deviations are observed for the LSTM and Elastic Net models, which align with their relatively weaker test set performance.

As reported in Table 3, the XGBoost model attains the lowest mean absolute error (MAE) and highest correlation with the benchmark, with the MLP model following closely. All models achieve a sign accuracy of approximately 93%, indicating a high



(a) Scatter plot with benchmark values

(b) Distribution of forecast errors

Figure 3. Benchmark to Koop et al. (2023): Scatter (L) and Error Distribution (R)

level of directional consistency with the benchmark series.

Table 3. US Benchmark Comparison

Model	MAE	RMSE	Corr.	Sign Acc.
MLP	1.269	1.624	0.773	0.933
LSTM	1.401	2.127	0.642	0.924
XGB	1.168	1.504	0.810	0.933
ENET	1.311	1.918	0.708	0.927

Note: MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error) are scale-dependent measures of forecast accuracy where lower values indicate better performance. Correlation measures linear association between predicted and actual values. Sign Accuracy refers to the proportion of times the model correctly predicts the direction of change.

3.2.2 UK Case: Monthly GDP from ONS

The UK's ONS publishes official monthly GDP estimates from January 1997 onward, with the most recent data available as of April 2025. The series is indexed to 100 at October 2022. For comparison, we rescale our predicted monthly GDP series to match this base and focus on the overlapping period. Figure 4 shows the comparison between the official ONS series and our predicted values using the MLP model.[16]

Overall, the MLP model closely tracks the general pattern of the official monthly GDP series. Slight deviations are observed during periods of heightened volatility, such as the COVID-19 shock, where the predicted series tends to display greater

[16]We present only the MLP model in the main text, as including all models in a single figure obscures the dynamics due to extreme outliers. Full model comparisons are available in Appendix F.

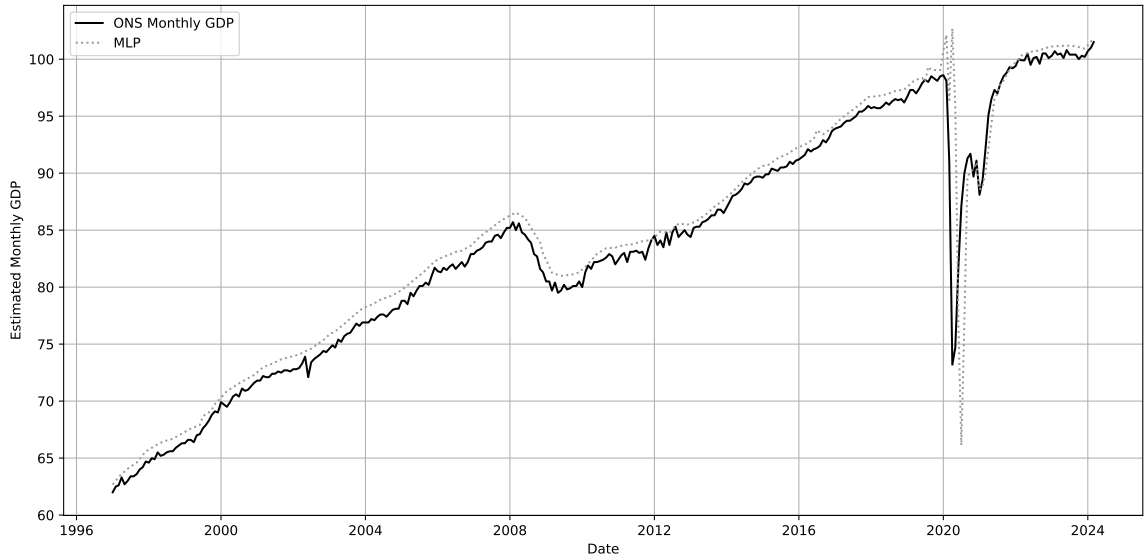
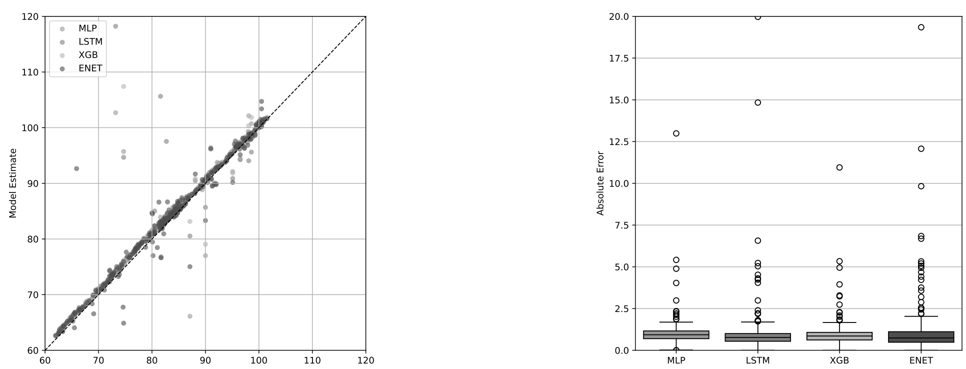


Figure 4. Benchmark to ONS: UK Estimated Monthly GDP (MLP)

amplitude and noise. As reported in Table 4, benchmark performance for the UK is somewhat lower than that for the US but still acceptable. This performance gap may be partly attributed to data availability: the US model utilizes 15 explanatory variables, while the UK model includes only 13.



(a) Scatter plot with benchmark values

(b) Distribution of forecast errors

Figure 5. Benchmark to ONS: Scatter (L) and Error Distribution (R)

Figure 5 presents the corresponding scatter and error distribution plots. The Elastic Net model performs poorly in this case, due in part to extreme outliers observed in certain periods. Since the UK's ONS reports monthly GDP as an index series, we transform the data to enable a meaningful comparison of metrics like sign accuracy and to maintain consistency with the US benchmark. We first compute the monthly growth rate by log-differencing the index, and then annualize it using Equation (7). Under this transformation, the MLP model achieves a sign accuracy of approximately 93%, on par with the US case. Other models perform less well, with the Elastic Net model falling as low as 80%.

Table 4. UK Benchmark Comparison

Model	MAE	RMSE	Corr.	Sign Acc.
MLP	1.213	2.664	0.972	0.929
LSTM	1.176	3.335	0.954	0.907
XGB	1.323	5.733	0.873	0.910
ENET	49.365	779.262	0.056	0.807

Note: MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error) are scale-dependent measures of forecast accuracy where lower values indicate better performance. Correlation measures the linear association between predicted and actual values. Sign Accuracy refers to the proportion of times the model correctly predicts the direction of change. Since the UK benchmark is based on level data, sign accuracy is not directly comparable. To enable meaningful comparison, we transformed the series according to equation (7), as done in the US benchmark.

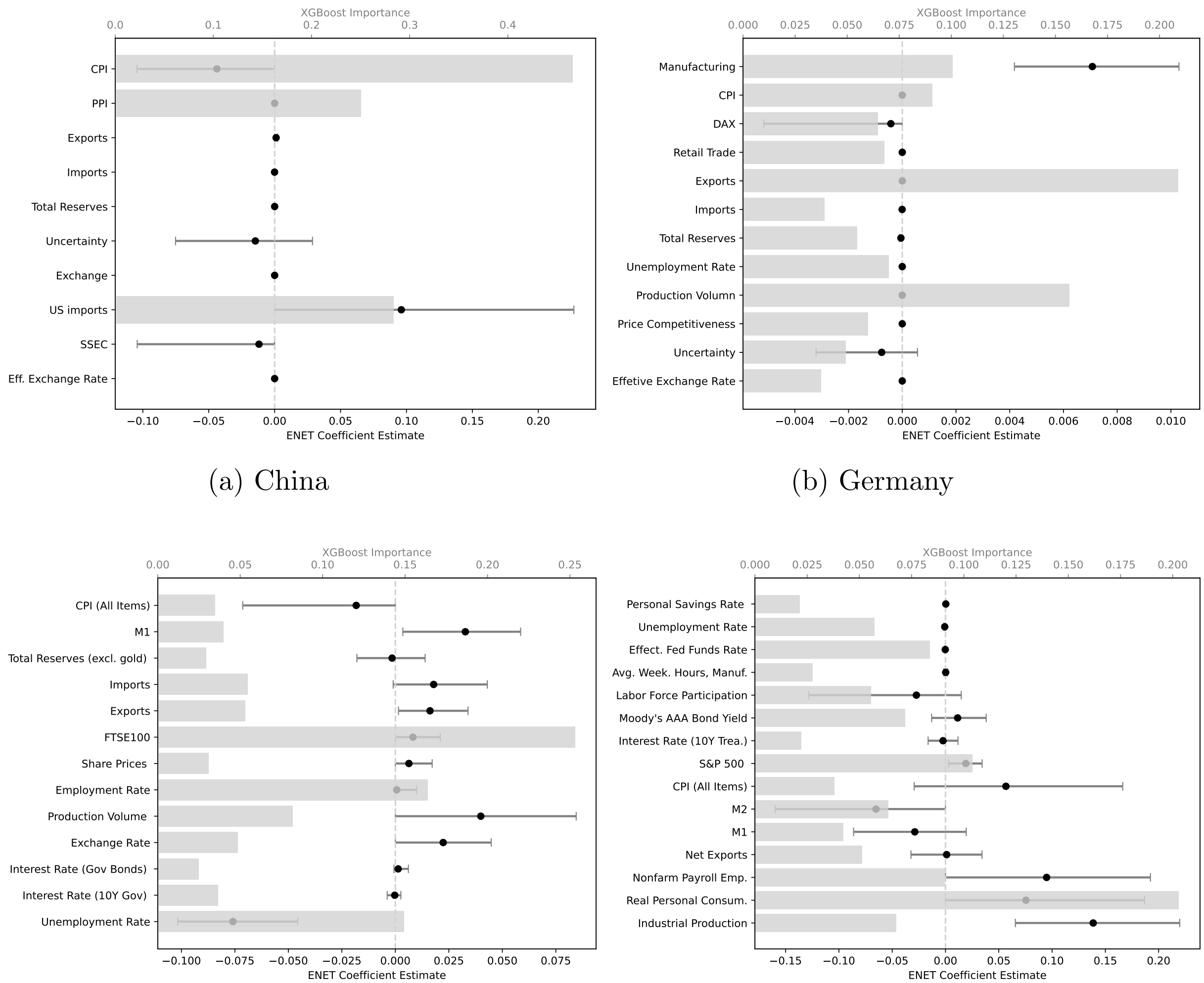
3.3 Interpretation with XGBoost and Elastic Net

As demonstrated in our results, neural network models generally exhibit strong out-of-sample predictive performance. However, their "black box" nature presents a key limitation in academic research, where interpretability is often as important as accuracy.[17]

In contrast, models based on decision trees or linear regressions offer more transparent interpretations by providing variable importance scores or direct coefficient estimates. These models can thus serve a complementary role, especially when the differences in predictive performance across models are not substantial. For instance, XGBoost and Elastic Net allow for further examination of the role each explanatory variable plays in the prediction task.

XGBoost, being tree-based, provides intuitive variable importance rankings that are accessible even to non-specialists. However, it does not yield information on coefficient signs or confidence intervals, limiting its utility. On the other hand, while Elastic Net often underperforms in terms of prediction accuracy, it offers notable advantages in interpretability. Through its combination of LASSO (λ_1) and Ridge

[17] Although recent advances—such as SHAP (SHapley Additive exPlanations)—have made strides in opening these models to greater scrutiny and interpretability. See Lundberg and Lee (2017).



(a) China (b) Germany

(c) UK (d) US

Figure 6. Elastic Net Coefficients and XGBoost Importance

(λ_2) penalties,[18] it performs effective variable selection and regularization, allowing researchers to estimate coefficient signs and confidence intervals while mitigating the influence of irrelevant predictors.

Figure 6 presents a comparative forest plot of variable importance and estimated coefficients across countries.[19] Discrepancies between the two approaches are occasionally observed. For example, in the UK case, the employment rate receives low weight in the Elastic Net model, likely due to high correlation with the unemployment rate; yet, XGBoost assigns it notable importance. These differences highlight how each model processes information: Elastic Net, based on a regular-

[18]See Appendix B for further details.
[19]See Appendix F for detailed values.

ized linear structure, tends to suppress one of the correlated variables to mitigate multicollinearity, effectively selecting among them. In contrast, XGBoost can accommodate highly correlated variables simultaneously, as it evaluates splits sequentially and non-parametrically in a tree structure.

This distinction can be particularly informative for applied researchers. When one aims to investigate the joint role of closely related economic indicators—such as employment and unemployment rates—Elastic Net may not capture their combined effect due to collinearity penalties, whereas XGBoost may still reflect their joint contribution to prediction. Therefore, contrasting the outputs of these two models provides additional interpretative value, especially in settings where the relationships among predictors are of substantive interest rather than merely nuisance correlations.

These considerations highlight the importance of using multiple models not only for prediction but also for robust interpretability. Cross-model validation of variable relevance can help ensure that empirical findings are not artifacts of a single modeling choice but are instead supported by diverse analytical approaches.

4 Conclusion

This paper proposes a scalable, data-driven framework for estimating monthly GDP using a suite of machine learning models. By training four models—MLP, LSTM, XGBoost, and Elastic Net—on quarterly GDP data and monthly economic indicators, we generate high-frequency GDP estimates for China, Germany, the UK, and the US. The methodology is validated through benchmarking against official monthly GDP figures for the UK and academic estimates for the US, demonstrating both credibility and robustness across different data environments. Among the models, MLP stands out for its balance of predictive accuracy and computational efficiency.

The main contribution of this research is to provide a flexible alternative to traditional econometric approaches, which often rely on strong structural assumptions. While the machine learning framework may be more computationally intensive, it adapts well to varying levels of data availability, making it broadly applicable across countries. Moreover, by combining interpretable models such as XGBoost and Elastic Net with "black box" neural networks, the framework facilitates model-based validation of variable importance and economic mechanisms. This enhances both transparency and trust in the estimated outcomes, offering policymakers timely and interpretable tools for high-frequency economic monitoring.

Despite its promising performance, the framework has several limitations that suggest avenues for future research. A key constraint is its reliance on official quarterly GDP figures for the final reconciliation step, limiting the model's application to historical estimation. To enable real-time nowcasting, a natural extension would be to generate an out-of-sample forecast for the current quarter's GDP, which could then serve as the target for monthly reconciliation. Such an approach would preserve the structure of the model while expanding its utility to current-quarter tracking.[20]

Another potential improvement lies in refining the variable selection process. This study uses the full set of available indicators for each country, but a two-stage strategy could improve both model performance and computational efficiency. For example, Elastic Net could be used in the first stage for variable selection, leveraging its regularization to eliminate noisy or redundant predictors. A nonlinear model such as MLP could then be trained on this reduced set, improving both signal-to-noise ratio and interpretability.

Finally, the framework's applicability could be extended, particularly to regions with sparse data, by incorporating novel data sources. A growing body of research has demonstrated a strong correlation between nighttime light intensity (K. Chen et al., 2025; Pinkovskiy & Sala-i-Martin, 2016), captured via satellite imagery, and economic activity. Integrating such data as an explanatory variable could provide a powerful proxy for economic output, especially in countries or sub-national regions where traditional economic indicators are unavailable or published with significant delays.[21] This would not only enhance the model's performance but also broaden its reach, offering a valuable tool for understanding economic dynamics in data-poor environments. These proposed extensions represent valuable directions for future inquiry, building upon the flexible foundation established in this paper.

[20]For nowcasting purposes, two key conditions must be met under this framework: the availability of sufficiently high-frequency data and the timely updating of such data.

[21]However, a key limitation is that most nighttime light data have only been available since the mid-2010s, resulting in a relatively short time span for training the model at this moment.

References

Brave, S. A., Butters, R. A., Kelley, D., et al. (2019). A New "Big Data" Index of US Economic Activity. *Economic Perspectives*, *43*(1), 1–30.

Chen, K., Wang, Q., Xu, J., & Yao, J. (2025). The International Spillover of Monetary Policy Shock: New Evidence from Nighttime Light.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Chow, G. C., & Lin, A. (1971). Best Linear Unbiased Interpolation, Distribution, and Extrapolation of Time Series by Related Series. *The Review of Economics and Statistics*, 372–375.

Chu, B., & Qureshi, S. (2023). Comparing Out-of-Sample Performance of Machine Learning Methods to Forecast US GDP Growth. *Computational Economics*, *62*(4), 1567–1609.

Denton, F. T. (1971). Adjustment of Monthly or Quarterly Series to Annual Totals: An Approach Based on Quadratic Minimization. *Journal of the American Statistical Association*, *66*(333), 99–102.

Ghosh, S., & Ranjan, A. (2023). A Machine Learning Approach to GDP Nowcasting: An Emerging Market Experience. *Bulletin of Monetary Economics and Banking*, *26*, 33–54.

Giannone, D., Reichlin, L., & Small, D. (2008). Nowcasting: The Real-Time Informational Content of Macroeconomic Data. *Journal of Monetary Economics*, *55*(4), 665–676.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural Computation*, *9*(8), 1735–1780.

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning with Applications in Python*. Springer.

Jansen, W. J., Jin, X., & de Winter, J. M. (2016). Forecasting and Nowcasting Real GDP: Comparing Statistical Models and Subjective Forecasts. *International Journal of Forecasting*, *32*(2), 411–436.

Koop, G., McIntyre, S., Mitchell, J., & Poon, A. (2021). Nowcasting 'True' Monthly US GDP During the Pandemic. *National Institute Economic Review*, *256*, 44–70.

Koop, G., McIntyre, S., Mitchell, J., & Poon, A. (2023). Reconciled Estimates of Monthly GDP in the United States. *Journal of Business & Economic Statistics*, *41*(2), 563–577.

Litterman, R. B. (1983). A Random Walk, Markov Model for the Distribution of Time Series. *Journal of Business & Economic Statistics*, *1*(2), 169–173.

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, *30*.

Mariano, R. S., & Murasawa, Y. (2003). A New Coincident Index of Business Cycles Based on Monthly and Quarterly Series. *Journal of Applied Econometrics*, *18*(4), 427–443.

Mariano, R. S., & Murasawa, Y. (2010). A Coincident Index, Common Factors, and Monthly Real GDP. *Oxford Bulletin of Economics and Statistics*, *72*(1), 27–46.

Mitchell, J., Smith, R. J., Weale, M. R., Wright, S., & Salazar, E. L. (2005). An Indicator of Monthly GDP and an Early Estimate of Quarterly GDP Growth. *The Economic Journal*, *115*(501), F108–F129.

Pinkovskiy, M., & Sala-i-Martin, X. (2016). Lights, Camera… Income! Illuminating the National Accounts-household Surveys Debate. *The Quarterly Journal of Economics*, *131*(2), 579–631.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning Representations by Back-propagating Errors. *Nature*, *323*(6088), 533–536.

Said, S. E., & Dickey, D. A. (1984). Testing for Unit Roots in Autoregressive-Moving Average Models of Unknown Order. *Biometrika*, *71*(3), 599–607.

Zou, H., & 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.

Appendices

A Data Descriptions

All data used in this study are publicly available from open-source platforms. Detailed instructions for data acquisition—including Python scripts for API-based access—are provided in the replication package.[22] Most datasets are retrieved via the FRED API, which requires user registration (free of charge), and the Yahoo Finance API, which does not require registration. As of June 2025, all sources remain accessible. However, it is important to note that API providers may not be the original data publishers, and access could be discontinued without prior notice. The processed datasets used in this study are stored as `master_<country>.csv` files in model folder within the replication package. All summary statistics are based on raw data prior to any log-differencing or standardization.

A.1 China Data

Table A1. China Dataset Description

Variable	Code	Unit Root	Transform.
CPI (All Items, Growth)	CPALTT01CNM657N	No	$(X_m - \mu_j)/\sigma_j$
PPI (Furniture, Household)	WPU1261	Yes	$\Delta \log(X_m)$
Total Reserves (excl. gold)	TRESEGCNM052N	Yes	$\Delta \log(X_m)$
Exports	XTEXVA01CNM664S	Yes	$\Delta \log(X_m)$
Imports	XTIMVA01CNM664S	No	$\Delta \log(X_m)$
US Imports	IMPCH	Yes	$\Delta \log(X_m)$
Effective Exchange Rate	RBCNBIS	Yes	$\Delta \log(X_m)$
Exchange Rate (RMB/USD)	EXCHUS	Yes	$\Delta \log(X_m)$
Policy Uncertainty	CHNMAINLANDEPU	Yes	$\Delta \log(X_m)$
SSEC Index	-	Yes	$\Delta \log(X_m)$
Real GDP	CHNGDPNQDSMEI	-	$\Delta \log(X_m)$

Note: Variable codes are retrieved from FRED (https://fred.stlouisfed.org/). SSEC Index is obtained from Investing.com (https://www.investing.com/).

[22] https://github.com/Yonggeun-Jung/monthly_gdp

Table A2. China Data Summary Statistics

Variable	Obs.	Mean	Std. Dev.	Min	Max
CPI (All Items, Growth)	357	0.248	0.796	-1.403	4.048
PPI (Furniture, Household)	357	220.260	43.093	156.200	330.018
Total Reserves (excl. gold)	357	1.81e+06	1.44e+06	26355	4.01e+06
Exports	357	8.04e+11	5.87e+11	5.75e+10	2.15e+12
Imports	357	6.64e+11	4.63e+11	7.21e+10	1.61e+12
US Imports	357	24764.98	14770.62	2183	52081.07
Exchange Rate (RMB/USD)	357	7.381	0.859	6.051	8.725
Effective Exchange Rate	357	82.613	13.719	52.800	106.380
Policy Uncertainty	357	145.334	112.203	10.111	661.828
SSEC Index	357	2256.469	1017.002	333.920	5954.770
Real GDP	119	1.14e+13	9.59e+12	9.38e+11	3.36e+13

Note: Summary statistics are calculated using raw values before any transformation (e.g., differencing or standardization).

A.2 Germany Data

Table A3. Germany Dataset Description

Variable	Code	Unit Root	Transform.
CPI (All Items)	DEUCPIALLMINMEI	Yes	$\Delta \log(X_m)$
Retail Trade	DEUSARTMISMEI	Yes	$\Delta \log(X_m)$
Unemployment Rate	LRHUTTTTDEM156S	Yes	$\Delta \log(X_m)$
Production Volume	DEUPROINDMISMEI	Yes	$\Delta \log(X_m)$
Manufacturing	DEUPRMNTO01GPSAM	No	$(X_m - \mu_j)/\sigma_j$
Exports	XTEXVA01DEM664S	Yes	$\Delta \log(X_m)$
Imports	XTIMVA01DEM664S	Yes	$\Delta \log(X_m)$
Price Competitiveness	-	Yes	$\Delta \log(X_m)$
Effective Exchange Rate	RNDEBIS	Yes	$\Delta \log(X_m)$
Total Reserves (excl. gold)	TRESEGDEM052N	Yes	$\Delta \log(X_m)$
DAX Index	GDAXI	Yes	$\Delta \log(X_m)$
Policy Uncertainty	EUEPUINDXM	Yes	$\Delta \log(X_m)$
Real GDP	CLVMNACSCAB1GQDE	-	$\Delta \log(X_m)$

Note: Variable codes are retrieved from FRED (https://fred.stlouisfed.org/) and Yahoo Finance (https://finance.yahoo.com/). Price competitiveness index is obtained from the Deutsche Bundesbank (https://www.bundesbank.de/en).

Table A4. Germany Data Summary Statistics

Variable	Obs.	Mean	Std. Dev.	Min	Max
CPI (All Items)	396	90.15	13.72	63.998	124.19
Retail Trade	396	100.12	7.16	90.71	122.61
Unemployment Rate	396	6.66	2.48	2.90	11.20
Production Volume	396	87.85	11.68	65.25	107.38
Manufacturing	396	0.07	2.15	-22.02	11.94
Exports	396	7.21e+10	3.13e+10	2.58e+10	1.39e+11
Imports	396	6.08e+10	2.66e+10	2.33e+10	1.34e+11
Price Competitiveness	396	98.72	3.90	90.90	111.90
Effective Exchange Rate	396	102.63	4.33	93.89	116.97
Total Reserves (excl. gold)	396	64913.49	15365.64	39972.28	120255.30
DAX Index	396	7025.15	4126.87	1373.61	16681.73
Policy Uncertainty	396	153.02	77.56	47.69	433.28
Real GDP	132	644258.50	80194.45	514515.30	770584.30

Note: Summary statistics are calculated using raw values before any transformation (e.g., differencing or standardization).

A.3 UK Data

Table A5. UK Dataset Description

Variable	Code	Unit Root	Transform.
CPI (All Items)	GBRCPIALLMINMEI	Yes	$\Delta\log(X_m)$
Production Volume	GBRPROINDMISMEI	Yes	$\Delta\log(X_m)$
Unemployment Rate	–	Yes	$\Delta\log(X_m)$
Employment Rate	–	Yes	$\Delta\log(X_m)$
Exports	XTEXVA01GBM664S	Yes	$\Delta\log(X_m)$
Imports	XTIMVA01GBM664S	Yes	$\Delta\log(X_m)$
Exchange Rate	EXUSUK	Yes	$\Delta\log(X_m)$
Interest Rate (Gov Bonds)	INTGSBGBM193N	Yes	$\Delta\log(X_m)$
Interest Rate (10Y Gov)	IRLTLT01GBM156N	Yes	$\Delta\log(X_m)$
Total Reserves (excl. gold)	TRESEGGBM052N	Yes	$\Delta\log(X_m)$
M1 Money Stock	–	Yes	$\Delta\log(X_m)$
Share Prices	SPASTT01GBM661N	Yes	$\Delta\log(X_m)$
FTSE 100 Index	FTSE	Yes	$\Delta\log(X_m)$
Real GDP	NGDPRSAXDCGBQ	—	$\Delta\log(GDP_q)$

Note: Codes, expect for FTSE 100 Index, are from FRED (https://fred.stlouisfed.org/), FTSE 100 Index is retrieved from Yahoo Finance (https://finance.yahoo.com/). Total Reserves and M1 series are from BOE (https://www.bankofengland.co.uk/boeapps/database/). Unemployment and employment rates are from ONS (https://www.ons.gov.uk/). Since BOE and ONS do not offer data in API format, we downloaded full-period datasets manually from their websites and processed them by hand.

Table A6. UK Data Summary Statistics

Variable	Obs.	Mean	Std. Dev.	Min	Max
CPI (All Items)	399	87.01	18.22	57.00	131.60
Production Volume	399	96.73	9.62	75.20	113.32
Unemployment Rate	399	6.24	1.88	3.60	10.70
Employment Rate	399	72.36	2.05	68.30	76.50
Exports	399	1.96e+10	6.42e+09	8.32e+09	3.32e+10
Imports	399	2.67e+10	1.14e+10	9.35e+09	5.87e+10
Exchange Rate	399	1.56	0.20	1.13	2.07
Share Prices	399	83.34	22.75	31.94	120.02
Interest Rate (Gov Bonds)	399	4.34	2.55	0.21	10.59
Interest Rate (10Y Gov)	399	4.34	2.55	0.21	10.59
Total Reserves (excl. gold)	399	82244.13	46307.86	28912.71	176024.00
M1 Money Stock	399	992469.80	680818.10	174324.00	2575027.00
Share Prices	399	83.34	22.75	31.94	120.02
FTSE 100 Index	399	5495.65	1500.32	2106.64	7915.01
Real GDP	133	500639.00	88311.65	344987.00	638746.00

Note: Summary statistics are calculated using raw values before any transformation (e.g., differencing or standardization).

A.4 US Data

Table A7. US Dataset Description

Variable	Code	Unit Root	Transform.
Avg. Week. Hours, Manuf.	AWHMAN	No	$(X_m - \mu_j)/\sigma_j$
CPI (All Items)	CPIAUCSL	Yes	$\Delta \log(X_m)$
Industrial Production	INDPRO	Yes	$\Delta \log(X_m)$
Real Personal Consum.	DPCERA3M086SBEA	Yes	$\Delta \log(X_m)$
Effect. Fed Funds Rate	FEDFUNDS	No	$(X_m - \mu_j)/\sigma_j$
Interest Rate (10Y Trea.)	GS10	Yes	$\Delta \log(X_m)$
Unemployment Rate	UNRATE	No	$(X_m - \mu_j)/\sigma_j$
M2 Money Stock	M2SL	Yes	$\Delta \log(X_m)$
M1 Money Stock	M1SL	Yes	$\Delta \log(X_m)$
Labor Force Participation	CIVPART	Yes	$\Delta \log(X_m)$
Personal Savings Rate	PSAVERT	No	$(X_m - \mu_j)/\sigma_j$
Nonfarm Payroll Emp.	PAYEMS	Yes	$\Delta \log(X_m)$
Moody's AAA Bond Yield	AAA	Yes	$\Delta \log(X_m)$
Net Exports	BOPTEXP	Yes	$\Delta \log(X_m)$
S&P 500 Index	GSPC	Yes	$\Delta \log(X_m)$
Real GDP	GDPC1	—	$\Delta \log(GDP_q)$

Note: All data, except for the S&P 500 Index, are from FRED (https://fred.stlouisfed.org/). The S&P 500 Index is obtained from Yahoo Finance (https://finance.yahoo.com/).

Table A8. US Data Summary Statistics

Variable	Obs.	Mean	Std. Dev.	Min	Max
Avg. Week. Hours, Manuf.	396	41.19	0.61	38.40	42.40
CPI (All Items)	396	211.53	46.71	138.30	317.60
Industrial Production	396	91.84	11.41	61.48	104.10
Real Personal Consum.	396	83.12	19.72	49.07	123.08
Effect. Fed Funds Rate	396	2.60	2.18	0.05	6.54
Interest Rate (10Y Trea.)	396	4.01	1.75	0.62	7.96
Unemployment Rate	396	5.68	1.79	3.40	14.80
M2 Money Stock	396	9526.84	5663.39	3381.20	21749.60
M1 Money Stock	396	4237.49	5996.20	910.40	20727.10
Labor Force Participation	396	64.85	1.89	60.10	67.30
Personal Savings Rate	396	5.94	2.92	1.40	32.00
Nonfarm Payroll Emp.	396	134619.49	12153.07	108313.00	158942.00
Moody's AAA Bond Yield	396	5.40	1.60	2.14	8.68
Net Exports	396	142927.29	64637.35	50044.00	273516.00
S&P 500 Index	396	1776.10	1244.48	407.36	6017.37
Real GDP	132	16525.10	3582.27	10236.43	23542.35

Note: Summary statistics are calculated using raw values before any transformation (e.g., differencing or standardization).

B Model Descriptions

B.1 MLP: Multi-Layer Perceptron

In the Multi-Layer Perceptron (MLP) networks we estimate the regression model

$$Y_q = f(X_q; \theta) + e_q, \tag{B1}$$

where $f(X_q; \theta)$ is the function learned by an MLP which is a class of feedforward artificial neural networks. An MLP models the dependent (target) variable as a nested series of nonlinear transformations of the explanatory variables (features). For instance, a simple MLP with one hidden layer can be represented as:

$$f(X_q; \theta) = W_2\, \sigma(W_1 X_q + b_1) + b_2, \tag{B2}$$

where W_1, W_2 are weight matrices, b_1, b_2 are bias vectors, σ is a nonlinear activation function, and $\theta = \{W_1, b_1, W_2, b_2\}$ represents the set of all model parameters. The model parameters θ are estimated by minimizing the mean squared error (MSE) between the actual quarterly growth Y_q and the model's predictions $f(X_q; \theta)$. This minimization is performed using the Adam optimizer.[23] The architecture and training process of the MLP are governed by several hyperparameters, which we optimize through a Bayesian optimization search to find the best-performing model specification.[24]

B.2 LSTM: Long Short-Term Memory

The Long Short-Term Memory (LSTM) network is a type of Recurrent Neural Network (RNN) specifically designed to model time-series and sequential data (Hochreiter & Schmidhuber, 1997). Unlike feedforward networks such as the MLP, LSTM contains feedback connections, allowing information to persist. Its key innovation is the gating mechanism (comprising forget, input, and output gates), which regulates the flow of information through the network's cell state. This structure enables the model to effectively capture long-range dependencies and mitigate the vanishing

[23] Adam (Adaptive Moment Estimation) is a stochastic optimizer that adapts learning rates using estimates of first and second moments of gradients (Kingma, 2014).

[24] In practice, we utilized the `Keras-Tuner` package for this purpose. Bayesian optimization is a sequential strategy that finds the optimal hyperparameter combination by building a probabilistic model of the function mapping hyperparameters to model performance, making the search more efficient than random or grid search. See https://keras.io/keras_tuner/.

gradient problem common in simple RNNs.

In our implementation, we structure the quarterly data into sequences of a fixed length (t).[25] The LSTM model is then trained to predict the GDP growth of the subsequent quarter, Y_q, based on the sequence of explanatory variables from the preceding four quarters, $\{X_{q-1}, X_{q-2}, \ldots, X_{q-t}\}$. Similar to the MLP, the model parameters are estimated by minimizing the MSE using the Adam optimizer, and the optimal network architecture is determined through Bayesian hyperparameter optimization.

B.3 XGBoost: Extreme Gradient Boosting

We also employ XGBoost (Extreme Gradient Boosting), a high-performance, tree-based ensemble learning method based on the gradient boosting framework (Chen & Guestrin, 2016). The core idea of gradient boosting is to build models sequentially, with each new model correcting the errors (residuals) of its predecessor. XGBoost builds on this by using an optimized and regularized implementation of decision trees as base learners.

The model is trained to predict quarterly GDP growth, Y_q, from the aggregated quarterly features, X_q. We treat it as a standard regression problem and optimize its key hyperparameters—such as the number of trees (`n_estimators`), tree depth (`max_depth`), and `learning_rate`—using a cross-validated search to find the model with the best predictive performance. In addition to its predictive power, XGBoost also provides feature importance scores, offering a nonlinear perspective on the key drivers of GDP growth.

B.4 Elastic Net

This section describes the Elastic Net model that is employed mainly to explain the role of each explanatory variable in linear projection setting. We estimate the regression model

$$Y_q = f(X_q; \beta) + e_q, \tag{B3}$$

where $f(X_q; \beta)$ is a linear function of the aggregated regressors and e_q is an error term. The parameter vector $\beta \in \mathbb{R}^k$ denotes the marginal effect of each explanatory variable on quarterly GDP growth. Specifically, we minimize the following penalized

[25] We set 4 as the default

loss function:

$$\mathcal{L}(\beta) = \sum_{q} \left(Y_q - X_q^\top \beta\right)^2 + \lambda_1 \sum_{j=1}^{k} |\beta_j| + \lambda_2 \sum_{j=1}^{k} \beta_j^2, \tag{B4}$$

where the first term penalizes the discrepancy between the observed quarterly GDP growth (Y_q) and the predictions derived from the quarter-level regressors ($X_q^\top \beta$). The second and third terms introduce regularization penalties to control model complexity: specifically, (i) $\sum_{j=1}^{k} |\beta_j|$ is the sum of the absolute values of the elements, which encourages sparsity in the coefficients. This enables automatic selection of a subset of relevant variables, setting others to zero; (ii) $\sum_{j=1}^{k} \beta_j^2$ is the sum of squared coefficients, which shrinks large coefficients and mitigates overfitting when explanatory variables are highly correlated. This combination of penalties defines the Elastic Net regularization method, which balances the strengths of both LASSO (Least Absolute Shrinkage and Selection Operator) and Ridge regularization (James et al., 2023).[26] The regularization parameters λ_1 and λ_2 are treated as hyperparameters, which must be tuned by the researcher using a validation set. Their values determine the model's bias-variance tradeoff and affect the selection and shrinkage behavior of explanatory variables.

Furthermore, a nonparametric bootstrap approach is also employed to improve the model's predictive accuracy (Hansen, 2022; James et al., 2023). Specifically, this involves generating a number of bootstrap samples ($B = 50,000$) by resampling with replacement from the original dataset. The model is then re-estimated on each of these samples. This process not only helps in enhancing the stability of the predictions, often by averaging outcomes from the B models, but also allows for the empirical estimation of confidence intervals for the predicted monthly GDP figures.

C Algorithms

This section illustrates algorithms described in Section 2.1. We implement the algorithms by Python. The Python scripts are available at the replication package.[27]

[26] LASSO regularization (λ_1 penalty) performs variable selection by shrinking some coefficients to exactly zero, while Ridge regularization (λ_2 penalty) shrinks coefficients continuously without eliminating any (James et al., 2023).

[27] https://github.com/Yonggeun-Jung/monthly_gdp

C.1 Data Preprocessing

Algorithm 1: Monthly Transformation of Explanatory Variables

Data: Master file master_country.csv with columns: DATE, $X_1, \ldots, X_k$, Y; pre-defined log_diff_cols

Result: Transformed monthly explanatory variables X_prime_m_df

Load master_country.csv into DataFrame df.;
Convert DATE column to datetime format.;
Create quarter column by extracting quarter from DATE.;
Initialize X_prime_m_df with columns DATE and quarter.;
for *each explanatory variable* $X_j \in \{X_1, \ldots, X_k\}$ **do**
    Conduct ADF test on X_j to check stationarity and define log_diff_cols.;
    if $X_j \in$ *log_diff_cols* **then**
        $X'_{m,j} \leftarrow \log(X_j) - \log(X_{j,\text{prev}})$;
    else
        $X'_{m,j} \leftarrow X_j$;
    end
    Add $X'_{m,j}$ to X_prime_m_df.;
end

In Algorithm 1, we preprocess the dataset. The data collected for each country are stored in files named master_<country_name>.csv, each containing columns for DATE, $X_1, \ldots, X_k$, and Y. The primary task is to determine how to transform the X variables. For each X_j, we conduct an ADF test using tsa.stattools.adfuller function from statsmodels package to detect a unit root. If X_j exhibits a unit root and is not included in the exception list of variables that should be treated as growth rates, we apply a log-difference transformation. Otherwise, we retain X_j in its original form.

Algorithm 2: Quarterly Aggregation and Final Processing

Data: Transformed monthly `X_prime_m_df`; `log_diff_cols`

Result: Quarterly dataset `X_q_processed`, `Y_q_processed`

```
quarters ← unique sorted quarters from X_prime_m_df;
for each quarter q in quarters do
    X_current_q ← rows in quarter q from X_prime_m_df;
    for each variable j in explanatory variables do
        if j ∈ log_diff_cols then
            X_{q,j} ← Σ_{m∈q} X'_{m,j};
        else
            X_{q,j} ← (1/3) Σ_{m∈q} X'_{m,j};
        end
        X_q_row[j] ← X_{q,j};
    end
end
Extract Y_q as the first month of each quarter in df.;
Y_q_processed ← Y_q ← log(Y_q) − log(Y_{q−1}), dropping first value.;
X_q_processed ← concatenate log-diff and non-log columns.;
```

In Algorithm 2, we aggregate the monthly transformed variables into quarterly observations to match the frequency of the dependent variable Y. For each quarter q, we sum the monthly values of log-differenced variables and average the monthly values of non-log-differenced variables.[28] The Y_q is computed as the log difference.

C.2 Main Body of Models

C.2.1 MLP: Multi-Layer Perceptron

In Algorithm 3, we identify the optimal set of hyperparameters by iteratively training multiple MLP network specifications. Among the candidate models, we retain the one that achieves the lowest validation MSE as the final model. This procedure is implemented using the `keras-tuner` library provided by `TensorFlow`.

[28] In case of the non-log-differenced variables, we standardize it only after splinting data into training/test sets to avoid data leakage.

Algorithm 3: Neural Network Training and Hyperparameter Optimization

Data: `X_q_processed`; `Y_q_processed`; `non_log_cols`

Result: The best-performing trained model, $\hat{f}(\cdot;\hat{\theta})$

`/* Split Dataset and Standardization */`

Split `X_q_processed` and `Y_q_processed` into 80% training sets:;

$(X_q^{\text{train}}, Y_q^{\text{train}})$ and 20% test sets $(X_q^{\text{test}}, Y_q^{\text{test}})$;

if *non_log_cols is not empty* **then**

  scaler ← StandardScaler();

  Fit *scaler* on the `non_log_cols` of X_q^{train} and transform them;

  Transform the `non_log_cols` of X_q^{test} using the fitted *scaler*;

end

`/* Hyperparameter Tuning */`

Define a hyperparameter search space H for MLP architecture;

Initialize a Bayesian Optimization tuner to search space H, with the objective of minimizing the validation mean squared error (`val_mse`);

`/* Model Search and Selection */`

Execute the hyperparameter search on the training set, using the test set for validation;

`// The tuner iteratively builds, compiles, and trains MLP models. Each model is trained using the Adam optimizer and employs Early Stopping to prevent overfitting.`

$\hat{f}(\cdot;\hat{\theta})$ ← Retrieve the best model with the lowest `val_mse`;

While the search space can be expanded for better performance, We impose certain constraints to account for computational limitations. These constraints primarily involve setting upper bounds on the number of hidden layers, neurons, and training epochs. The default configuration uses up to 5 hidden layers and 512 neurons per layer, with the number of epochs[29] set to 2,000. While the exact values may vary by country, these defaults provide a reasonable balance between model flexibility and computational efficiency. The candidate activation functions include Rectified Linear Unit (ReLU), hyperbolic tangent (Tanh), Exponential Linear Unit (ELU), Scaled Exponential Linear Unit (SELU), and Swish, with the final layer using a linear function to reflect the continuous nature of the GDP growth.

[29] An epoch refers to one complete pass through the entire training dataset during model training (James et al., 2023).

C.2.2 LSTM: Long Short-Term Memory

Algorithm 4: LSTM Training and Hyperparameter Optimization

Data: `X_q_processed`; `Y_q_processed`; `non_log_cols`; `TIME_STEPS`

Result: The best-performing trained LSTM model, $\hat{f}_{\text{LSTM}}(\cdot;\hat{\theta})$

`/* Split Dataset and Standardization */`

Split `X_q_processed` and `Y_q_processed` into training and test sets;

Standardize `non_log_cols` using a scaler fitted only on the training data;

`/* Create Time Series Dataset */`

for *each dataset (training and test)* **do**

  Transform the 2D data (N, k) into a 3D sequential dataset $(N - T, T, k)$ using a sliding window of size `TIME_STEPS`;

  Store as $X_{\text{LSTM}}^{\text{train}}, Y_{\text{LSTM}}^{\text{train}}, X_{\text{LSTM}}^{\text{test}}, Y_{\text{LSTM}}^{\text{test}}$;

end

`/* Hyperparameter Tuning */`

Define a search space H for LSTM architecture (e.g., layers, units), dropout rates, and learning rate;

Initialize a Bayesian Optimization tuner to search space H, with the objective of minimizing `val_mse`;

`/* Model Search and Selection */`

Execute the hyperparameter search on $X_{\text{LSTM}}^{\text{train}}$, using $X_{\text{LSTM}}^{\text{test}}$ for validation;

`// The tuner iteratively builds and trains LSTM models using the Adam optimizer and Early Stopping.`

$\hat{f}_{\text{LSTM}}(\cdot;\hat{\theta}) \leftarrow$ Retrieve the best model with the lowest `val_mse`;

The hyperparameter tuning for the LSTM is also conducted using `TensorFlow`'s `keras-tuner` library, making the underlying code structure very similar to that of the MLP. However, since the LSTM takes into account historical data, we must define the length of the input sequence, denoted as t, and reshape the dataset into a 3D structure accordingly. A larger value of t allows the model to learn from a longer history, but it also results in the loss of more observations—especially important in our case, as training is performed on quarterly data. Taking this trade-off into account, we set the default value of t to 4.

C.2.3 XGBoost: Extreme Gradient Boosting

Algorithm 5: XGBoost Training and Hyperparameter Optimization

---

Data: `X_q_processed`; `Y_q_processed`; `non_log_cols`

Result: The best-performing trained XGBoost model, $\hat{f}_{\mathrm{XGB}}(\cdot;\hat{\theta})$

`/* Split Dataset and Standardization */`

1 Split `X_q_processed` and `Y_q_processed` into training and test sets;

2 Standardize `non_log_cols` using a scaler fitted only on the training data;

`/* Hyperparameter Tuning */`

3 Define a hyperparameter grid H for XGBoost (e.g., `n_estimators`, `max_depth`, `learning_rate`, `subsample`);

4 Initialize a `GridSearchCV` or `RandomizedSearchCV` instance with an `XGBRegressor` estimator, the search space H, and K-fold cross-validation;

`/* Model Search and Selection */`

5 Execute the hyperparameter search by fitting the search instance on the training set $(X_q^{\mathrm{train}}, Y_q^{\mathrm{train}})$;

6 $\hat{f}_{\mathrm{XGB}}(\cdot;\hat{\theta}) \leftarrow$ Retrieve the best estimator (model) from the search results that yielded the best cross-validation score;

XGBoost is implemented using the `xgboost` library in Python. As with other models, its key hyperparameters—such as tree depth (which controls the ability to capture complex relationships), step size (to prevent overfitting), and the minimum loss reduction required for further partitioning—must be carefully tuned by the researcher.

To reduce computation time, one can use `RandomizedSearchCV`, which tests a random subset of hyperparameter combinations and selects the best among them. Alternatively, `GridSearchCV` allows the user to explicitly define a wide search space, offering more precise tuning at the cost of significantly greater computation time.

A hybrid approach—conducting a randomized search first, then performing a grid search around the best-performing region—can strike a balance between efficiency and accuracy.

In this study, we prioritize predictive accuracy and therefore apply `GridSearchCV` from the outset. However, `RandomizedSearchCV` or a combined approach may be more appropriate in cases where computational efficiency is critical.

C.2.4 Elastic Net

Algorithm 6: Hyperparameter Tuning and Initial Estimation

Data: Quarterly dataset `X_q_processed`, `Y_q_processed`

Result: Optimal hyperparameters $(\lambda_1^*, \lambda_2^*)$ and coefficient estimates $\hat{\beta}_{\text{orig}}$

`/* Split into Train and Test Sets */`

Define training and test set: $(X_{\text{train}}, Y_{\text{train}})$, $(X_{\text{test}}, Y_{\text{test}})$;

Standardize `non_log_cols` using `StandardScaler` in each set;

`/* Define Penalized Loss Function */`

Elastic Net loss: $\mathcal{L}(\beta) = \sum_q (Y_q - X_q^\top \beta)^2 + \lambda_1 \sum_j |\beta_j| + \lambda_2 \sum_j \beta_j^2$;

`/* Cross Validation on Training Set */`

Define candidate values for (α, ρ).;

Map to $(\lambda_1 = \alpha\rho, \lambda_2 = \alpha(1-\rho))$.;

Split $(X_{\text{train}}, Y_{\text{train}})$ into K sequential folds.;

for *each pair* (α, ρ) *in the grid* **do**

  Initialize `total_cv_error` $\leftarrow 0$;

  for *each fold* $k = 1$ *to* K **do**

    Use folds 1 to k as training set; fold $k+1$ as validation set.;

    Fit ElasticNet(α, ρ) and obtain $\hat{\beta}^{(k)}$;

    Predict on validation set and compute MSE e_k;

    `total_cv_error` $\leftarrow$ `total_cv_error` $+ e_k$;

  end

  if *`total_cv_error`* $<$ *`min_cv_error`* **then**

    Update best (α^*, ρ^*);

  end

end

`/* Initial Model Estimation on the Training Set */`

Fit ElasticNet(α^*, ρ^*) on $(X_{\text{train}}, Y_{\text{train}})$ to obtain $\hat{\beta}_{\text{initial}}$;

Return $(\lambda_1^* = \alpha^*\rho^*,\ \lambda_2^* = \alpha^*(1-\rho^*))$ and $\hat{\beta}_{\text{initial}}$;

Algorithm 7: Bootstrap Estimation

Data: Original training data (`X_q_train`, `Y_q_train`), optimized hyperparameters $(\lambda_1^*, \lambda_2^*)$, transformed monthly explanatory variables `X_prime_m_df`

Result: Monthly GDP estimates $\hat{y}_m$ and optional confidence intervals

`/* 3.2 Bootstrap Procedure and Final Coefficient Estimation */`

$B \leftarrow 50000$; `/* Number of bootstrap */`

Initialize list `beta_hat_bootstrap_list`;

for $b = 1$ *to* B **do**

    $(X_{q,b}^*, Y_{q,b}^*) \leftarrow$ Resample with replacement from `X_q_train`, `Y_q_train`;

    $\hat{\beta}_b^* \leftarrow \mathrm{ElasticNet}(X_{q,b}^*, Y_{q,b}^*, \lambda_1^*, \lambda_2^*)$;

    Add $\hat{\beta}_b^*$ to `beta_hat_bootstrap_list`;

end

$\hat{\beta} \leftarrow \frac{1}{B} \sum_{b=1}^{B} \hat{\beta}_b^*$; `/* Use average */`

`/* 4. Monthly GDP Growth Estimation */`

`X_prime_m_for_prediction` $\leftarrow$ `X_prime_m_df`;

$\hat{y}_m \leftarrow (X_m')^\top \hat{\beta}$;

for *each month* m **do**

    Initialize list `y_m_bootstrap_predictions`;

    for *each* $\hat{\beta}_b^*$ *in* ***beta_hat_bootstrap_list*** **do**

        Add $(X_m')^\top \hat{\beta}_b^*$ to `y_m_bootstrap_predictions`;

    end

    Compute 2.5th and 97.5th percentiles $\rightarrow$ CI for $\hat{y}_m$;

end

C.3 Monthly GDP Estimation

Algorithm 8 produces a preliminary $\tilde{Y}_m = \hat{f}(X_m; \hat{\theta})$. Each $\tilde{Y}_m$ is an estimate on a quarterly scale and their aggregation is not arithmetically consistent with the observed quarterly totals. For each quarter, an adjustment factor (k_q) is calculated by taking the ratio of the actual quarterly growth to the sum of the preliminary monthly estimates within that quarter. This factor is then applied to scale each preliminary estimate, yielding the final adjusted monthly growth rate, $\hat{y}_m$.

Algorithm 8: Monthly GDP Growth Estimation and Reconciliation

Data: $\hat{f}(\cdot;\hat{\theta})$; `X_prime_m_df`; `Y_q_processed`; `non_log_cols`

Result: The final series of adjusted monthly GDP growth estimates $\hat{y}_m$

`/* Prepare Monthly Input Data */`

`X_prime_m_scaled` $\leftarrow$ Create a copy of the relevant columns from `X_prime_m_df`;

if *`non_log_cols` is not empty* **then**

    Transform the `non_log_cols` of `X_prime_m_scaled`;

end

`/* Generate Preliminary Monthly Estimates */`

$\tilde{Y}_m \leftarrow \hat{f}($`X_prime_m_scaled`$;\hat{\theta})$ for all months m;

$\hat{p}_m \leftarrow \tilde{Y}_m/3$ for all months m ; `// Calculate naive monthly estimates`

`/* Reconcile Monthly Estimates with Quarterly Actuals */`

Initialize an empty series for final monthly estimates, $\hat{y}_m$;

for *each quarter q* **do**

    sum_naive $\leftarrow \sum_{j\in q}\hat{p}_j$; `// Sum of naive estimates in the quarter`

    if *quarter q has a corresponding value in `Y_q_processed`* **then**

        $Y_q \leftarrow$ The actual observed growth for quarter q;

        if *sum_naive* $\neq 0$ **then**

            $k_q \leftarrow Y_q/$sum_naive ; `// Calculate adjustment factor`

            for *each month m in quarter q* **do**

                $\hat{y}_m \leftarrow k_q \cdot \hat{p}_m$; `// Apply adjustment factor`

            end

        else

            `// Distribute actual growth evenly if sum of naive estimates is zero`

            $\hat{y}_m \leftarrow Y_q/$(number of months in q) for each month $m \in q$;

        end

    end

    `// Use naive estimates if no actual quarterly data`

    $\hat{y}_m \leftarrow \hat{p}_m$ for each month $m \in q$;

end

D Reconciliation with Quarterly Aggregates

This section demonstrate the reconciliation method described in Section 2.1. Denton (1971) provides the main idea of it. The preliminary monthly estimates, contained in the vector $\tilde{Y} \in \mathbb{R}^{T\times 1}$ (where T is the total number of months), are reconciled to be consistent with the observed quarterly growth totals, given by the vector $Y \in \mathbb{R}^{Q\times 1}$ (where Q is the total number of quarters, and $T = 3Q$). This consistency is enforced by the temporal aggregation constraint:

$$J\hat{y} = Y, \tag{D1}$$

where $\hat{y} \in \mathbb{R}^{T\times 1}$ is the vector of final adjusted monthly estimates we wish to find. The matrix $J \in \mathbb{R}^{Q\times T}$ is the temporal aggregation matrix that sums the monthly data into quarterly totals. For monthly-to-quarterly aggregation, J takes the form:

$$J = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 \end{pmatrix} = I_Q \otimes \mathbf{1}_3^\top, \tag{D2}$$

where I_Q is the $Q \times Q$ identity matrix, $\mathbf{1}_3^\top$ is a 1×3 row vector of ones, and $\otimes$ denotes the Kronecker product.

Our proportional benchmarking method assumes that the final adjusted estimate for each month, $\hat{y}_m$, is proportional to its preliminary estimate, $\tilde{Y}_m$. The proportionality factor is constant for all months within the same quarter. This relationship can be expressed as:

$$\hat{y}_m = k_q \cdot \tilde{Y}_m, \quad \forall m \in q, \tag{D3}$$

where k_q is the adjustment factor for quarter q. In matrix notation, this is equivalent to:

$$\hat{y} = \mathrm{diag}(\tilde{Y}) J^\top k, \tag{D4}$$

where $\mathrm{diag}(\tilde{Y})$ is a $T \times T$ diagonal matrix with the preliminary estimates $\tilde{Y}_m$ on its diagonal, and $k \in \mathbb{R}^{Q\times 1}$ is the vector of quarterly adjustment factors.

To find the adjustment factors, we substitute this expression for $\hat{y}$ back into the aggregation constraint:

$$J\left(\operatorname{diag}(\tilde{Y})J^{\top}k\right) = Y. \tag{D5}$$

By solving for k, we obtain the vector of quarterly adjustment factors:

$$k = \left(J\operatorname{diag}(\tilde{Y})J^{\top}\right)^{-1}Y. \tag{D6}$$

The matrix $(J\operatorname{diag}(\tilde{Y})J^{\top})$ is a $Q \times Q$ diagonal matrix where each diagonal element is the sum of the preliminary monthly estimates for the corresponding quarter, i.e., $\sum_{j\in q}\tilde{Y}_j$. This makes its inversion straightforward.

Finally, substituting the solution for k back gives the closed-form solution for the final adjusted monthly estimates:

$$\hat{y} = \operatorname{diag}(\tilde{Y})J^{\top}\left(J\operatorname{diag}(\tilde{Y})J^{\top}\right)^{-1}Y. \tag{D7}$$

It ensures that the final monthly estimates $\hat{y}$ satisfy the quarterly constraints while preserving the relative intra-quarter dynamics provided by the preliminary indicator series $\tilde{Y}$.

E Estimation Results of MLP Model with Chinese CLI data

Concerns have been raised regarding the reliability of China's official statistics, including GDP (Holz, 2014; Rawski, 2001). In response, alternative measures of economic activity have been proposed, such as the so-called "Li Keqiang Index," which was inspired by remarks from Premier Li Keqiang and aims to proxy China's actual economic conditions in place of official GDP figures (Song & He, 2015).[30]

In Section 3, China shows the weakest test performance, which we primarily attribute to the limited number of explanatory variables and shorter time coverage. However, if the official GDP data fail to reflect true economic conditions—that is, if there are concerns not only about the quantity but also the quality of data—then the issue lies on a different level.

To investigate this, we re-estimate the MLP model using the same set of explanatory variables but replacing the official GDP series with the Composite Leading Indicator (CLI) for China, published by the OECD.[31] Although the CLI is a monthly index, we take quarterly averages to construct a comparable quarterly target series. As with the original GDP series, we apply log-differencing to the index level, and we keep all other experimental settings—such as timeframes and hyperparameter search settings—unchanged. We then compare the results with the original model using the official GDP.

Table E1. MLP Test Performance (China)

Dependent Var.	MSE	RMSE	MAE	R^2	Sign Acc.	sMAPE	Theil U1
GDP	0.006	0.077	0.060	0.628	0.875	78.218	0.335
CLI	0.000	0.007	0.006	0.794	0.792	118.143	0.224

Note: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and symmetric Mean Absolute Percentage Error (sMAPE, %) are all scale-dependent error metrics where lower values indicate better predictive accuracy. The coefficient of determination (R^2) measures the proportion of variance explained by the model and is better when closer to one. Sign Accuracy indicates how often the model correctly predicts the direction of change and is also better when higher. Theil's U1 statistic compares the forecast accuracy of a model to that of a naive (no-change) forecast; values closer to zero indicate better performance.

[30] Although the Economist magazine is often credited with constructing the Li Keqiang Index, it appears to use the index primarily in reporting rather than publishing the underlying data. Therefore, we rely on alternative sources for this study.

[31] The CLI series is available through the FRED API under the code CHNLORSGPNOSTSAM. https://fred.stlouisfed.org/series/CHNLORSGPNOSTSAM.

Table E2. China Benchmark with CLI

Model	MAE	RMSE	Corr.	Sign Acc.
MLP	0.004	0.018	0.040	0.818

Note: MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error) are scale-dependent measures of forecast accuracy where lower values indicate better performance. Correlation measures the linear association between predicted and actual values. Sign Accuracy refers to the proportion of times the model correctly predicts the direction of change. Since the CLI is based on level data, sign accuracy is not directly comparable. To enable meaningful comparison, we transformed the series according to equation (7), as done in the US benchmark.

As shown in Table E1, when the dependent variable is changed from official GDP to the CLI while keeping all other conditions constant, the test performance of the MLP model improves noticeably. While a direct comparison between the two dependent variables is inherently difficult due to their structural differences, this result suggests that part of the weaker performance observed in the China model in Section 3 may be attributable to the quality of the official GDP data, rather than solely to the quantity or coverage of explanatory variables.[32]

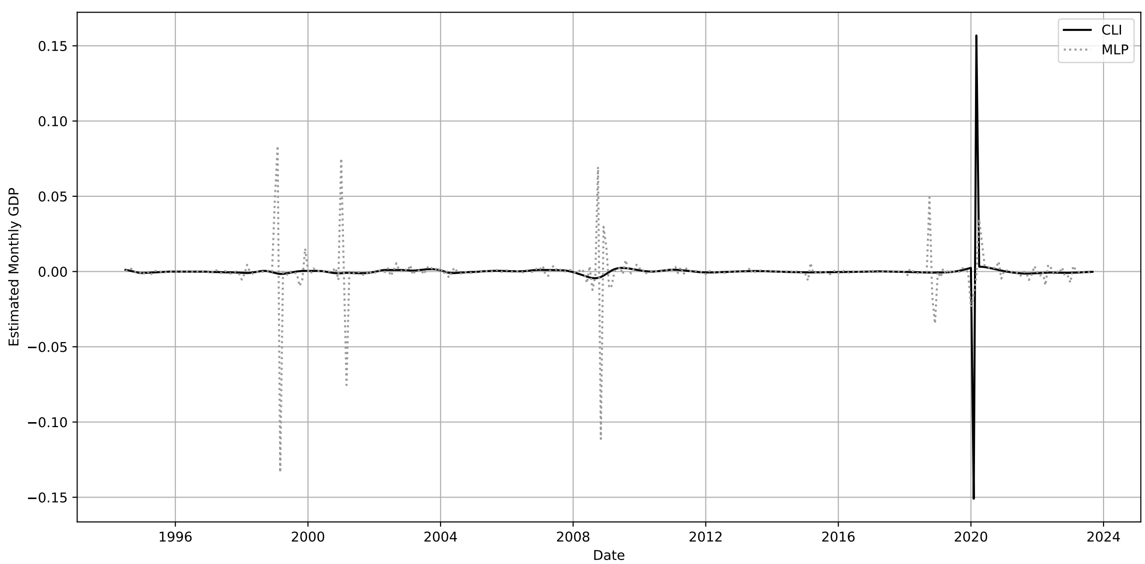


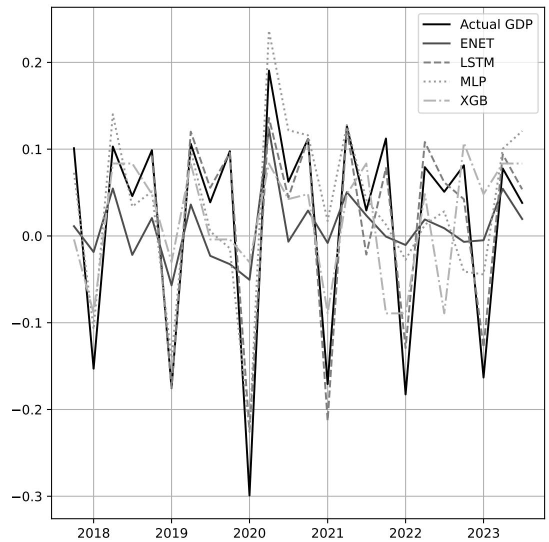
Figure E1. Actual and Predicted CLI growth (MLP)

In addition, the MLP model's predictions of monthly CLI growth follow the overall trends of the actual CLI series reasonably well. Notably, the model captures the

[32]This finding may also be interpreted as additional circumstantial support for concerns regarding the reliability of China's official GDP statistics. However, a detailed discussion of this issue lies beyond the scope of this paper.

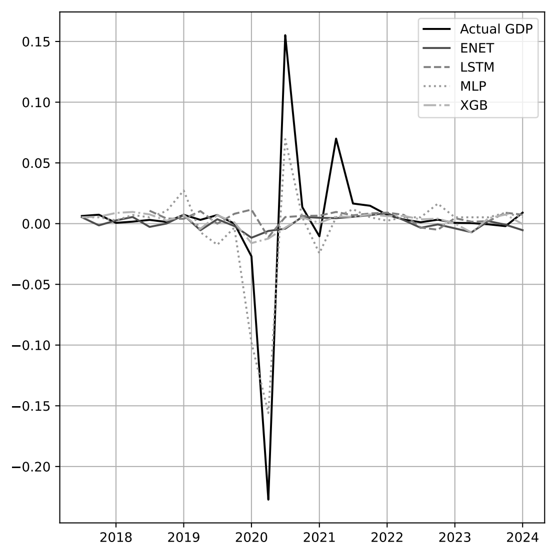
sharp contraction and rebound during the COVID-19 period—despite this period not being included in the training data—suggesting a degree of generalizability to large economic shocks. However, outside major episodes, the model tends to exhibit somewhat exaggerated responses to minor fluctuations compared to the actual CLI, indicating potential oversensitivity in less volatile periods.

F Supplementary Figures and Tables

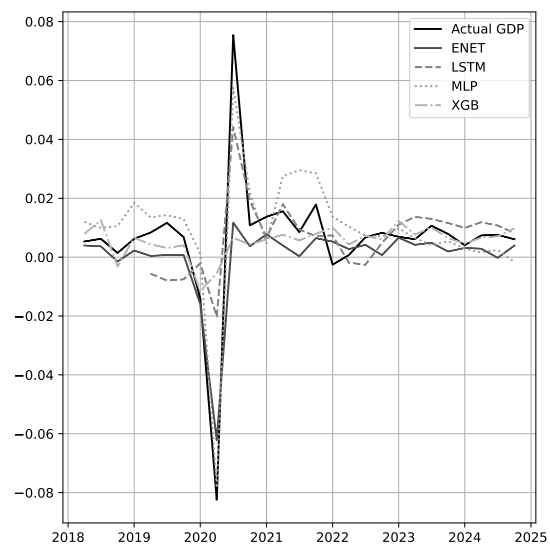


(a) China

(b) Germany

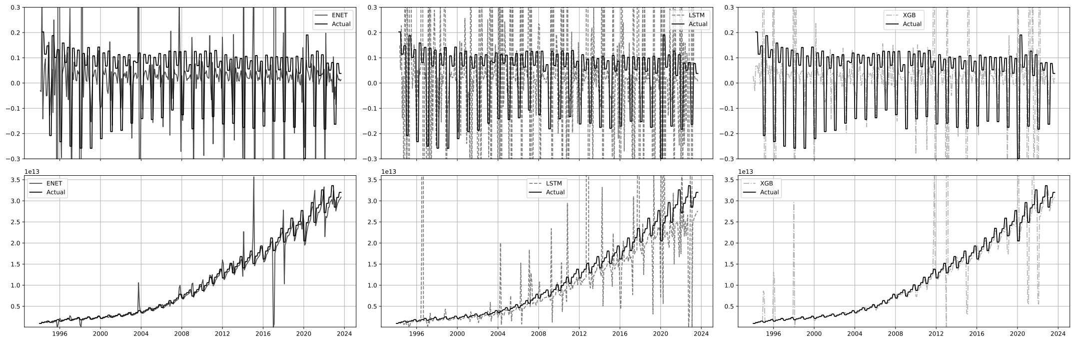


(c) UK

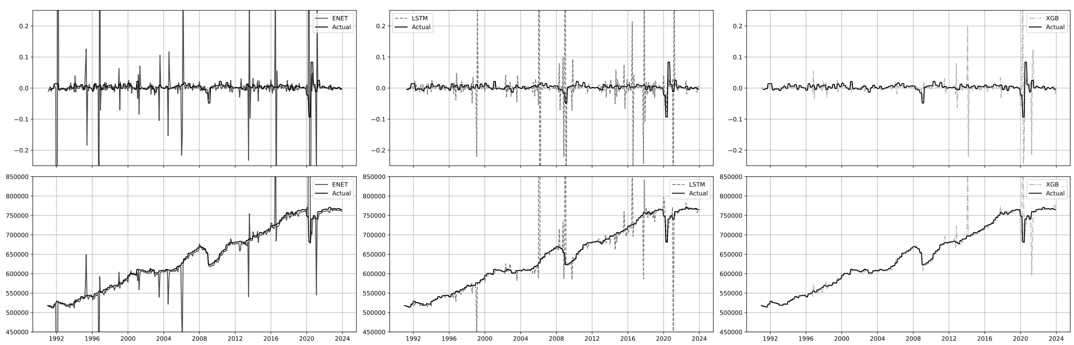


(d) US

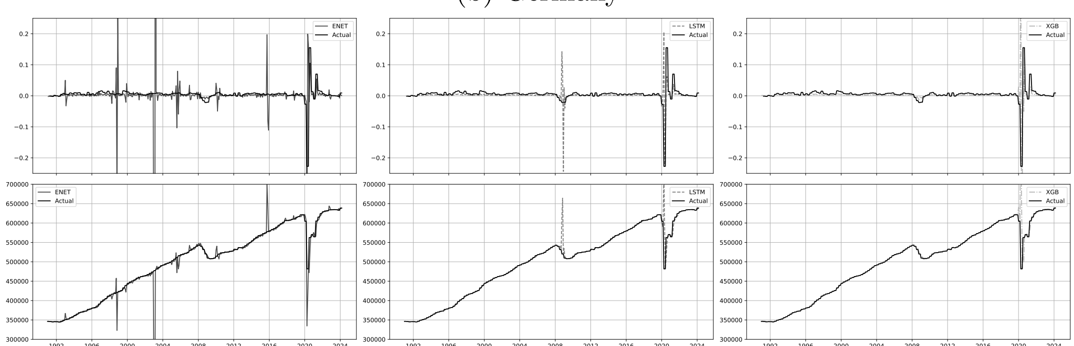
Figure F1. Test Performance of Each Model



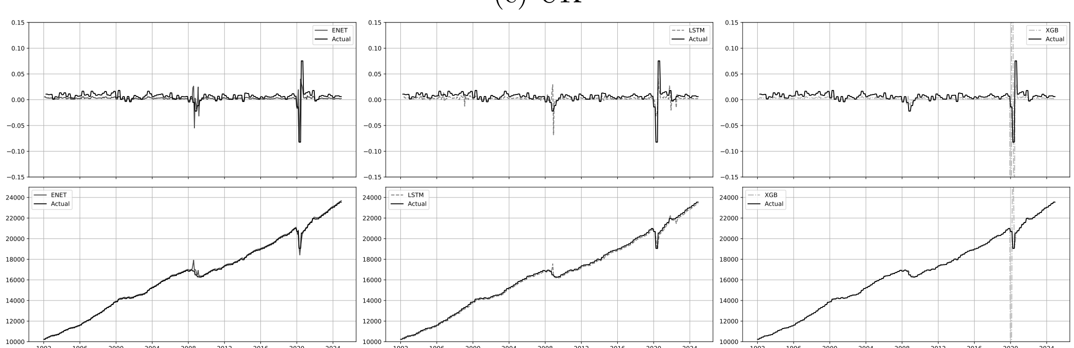
(a) China



(b) Germany



(c) UK



(d) US

Figure F2. Monthly GDP Estimation Results (Ex. MLP): Growth (U) and Level (D)

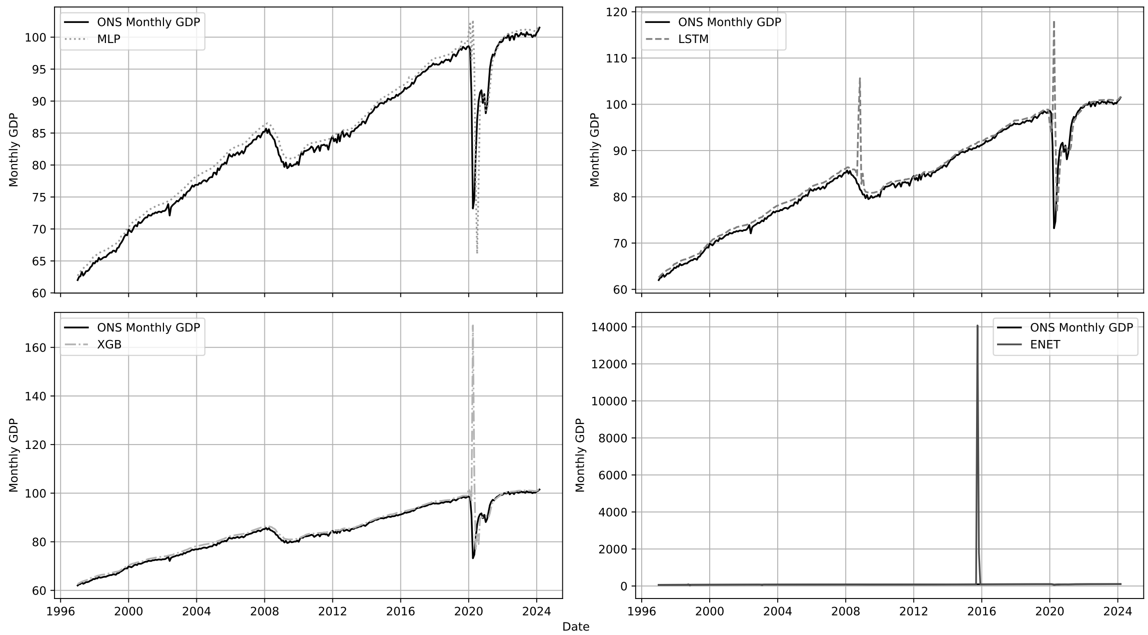


Figure F3. Benchmark to ONS: UK Estimated Monthly GDP (All Models)

Table F1. Elastic Net and XGBoost Results

	China		Germany		UK		US	
Variable	ENET	XGB	ENET	XGB	ENET	XGB	ENET	XGB
CPI	-0.044 (-0.10,0.00)	0.47	0.000 (0.00, 0.00)	0.09	-0.018 (-0.07, 0.00)	0.03	0.057 (-0.03, 0.17)	0.04
Retail Trade			0.000 (0.00, 0.00)	0.07				
M1 Money Stock					0.033** (0.00, 0.06)	0.04	-0.029 (-0.09, 0.02)	0.04
M2 Money Stock							-0.065 (-0.16, 0.00)	0.06
Total Reserves	0.000 (0.00, 0.00)	0.00	-0.000 (0.000,0.000)	0.05	-0.002 (-0.002,0.018)	0.03		
Exchange Rate	0.000 (0.00,0.00)	0.00			0.022* (0.00, 0.05)	0.05		
Unemployment Rate			0.000 (0.00, 0.00)	0.07	-0.076*** (-0.10, -0.05)	0.15	-0.001* (-0.00, -0.00)	0.06
Employment Rate					0.001 (0.00, 0.01)	0.16		
Labor Force							-0.027 (-0.13, 0.02)	0.06
Avg. Weekly Hours							0.000 (-0.00, 0.00)	0.03
Industrial Production							0.139*** (0.07, 0.22)	0.07
Production Volume			0.000 (0.00, 0.00)	0.16	0.040* (0.00, 0.09)	0.08		
Manufacturing			0.007*** (0.00, 0.01)	0.10				
PPI	0.000 (0.00,0.00)	0.25						
Real Personal Consum.							0.075 (0.00, 0.19)	0.20
Interest Rate (10Y Gov)					-0.000 (-0.00, 0.00)	0.04	-0.002 (-0.02, 0.01)	0.02
Interest Rate (Gov Bond)					0.001 (-0.00, 0.01)	0.02		
Effect. Fund Rate							-0.000 (-0.00, 0.00)	0.08
Effect. Ex. Rate	0.000 (0.00, 0.00)	0.00	0.000 (0.00, 0.00)	0.04				
S&P/FTSE/DAX/SSEC	-0.012 (-0.10,0.00)	0.00	-0.000 (-0.01, 0.00)	0.06	0.008 (0.00, 0.02)	0.25	0.019** (0.00, 0.03)	0.10
Share Prices					0.006 (0.00, 0.02)	0.03		
Moody's AAA Bond							0.011 (-0.01, 0.04)	0.07
(Net) Exports	0.001 (0.00,0.00)	0.00	0.000 (0.00, 0.00)	0.21	0.016** (0.00, 0.03)	0.05	0.001 (-0.03, 0.03)	0.05
Imports	-0.000 (0.00,0.00)	0.00	-0.000 (0.00, 0.00)	0.04	0.018 (-0.00, 0.04)	0.05		
US Imports from China	0.096 (0.00, 0.23)	0.28						
Price Competitiv.			0.000 (0.00, 0.00)	0.06				
Policy Uncertainty	-0.015 (-0.08,0.03)	0.00	-0.001 (-0.00, 0.00)					
Personal Savings Rate							0.000 (-0.00, 0.00)	0.02
Nonfarm Payroll Emp.							0.095* (0.00, 0.19)	0.09

Note: * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$. 95% bootstrap confidence intervals in parentheses. Coefficients rounded to 3 decimal places. Bootstrap based on 50,000 replications.

References for Appendices

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Denton, F. T. (1971). Adjustment of Monthly or Quarterly Series to Annual Totals: An Approach Based on Quadratic Minimization. *Journal of the American Statistical Association*, *66*(333), 99–102.

Hansen, B. (2022). *Econometrics*. Princeton University Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural Computation*, *9*(8), 1735–1780.

Holz, C. A. (2014). The Quality of China's GDP Statistics. *China Economic Review*, *30*, 309–338.

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning with Applications in Python*. Springer.

Kingma, D. P. (2014). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Rawski, T. G. (2001). What is Happening to China's GDP Statistics? *China Economic Review*, *12*(4), 347–354.

Song, X., & He, Z. (2015). The Keqiang Index: a New Benchmark for China's Development. *Social Indicators Research*, *123*, 661–676.