Macro in Motion: Visualizing FRED-MD Predictors for GDP Nowcasting

Hailey A. Reed
University of Connecticut
School of Computing
Storrs, USA
0009-0002-1506-4739

***Abstract*—Gross Domestic Product (GDP) is the most widely used indicator of macroeconomic performance across the world, yet official GDP estimates are released with substantial delay. This creates an information gap for policymakers, financial institutions, and firms that require real-time assessments of economic conditions. In this project, I build an interactive Plotly Dash dashboard to visualize the economic indicators that backbone current state-of-the-art GDP nowcasting models and to examine the behavior of multiple neural-network-based forecasting architectures. Using the FRED-MD panel of 126 monthly indicators, the dashboard allows users to explore raw, stationary, and normalized data, analyze leading versus lagging behavior, and compare model performance across stable and turbulent economic periods. This paper describes the data, preprocessing pipeline, visualization components, modeling methodology, and insights gained from building the dashboard. Results highlight that (i) indicator visualizations help identify candidate predictive features, (ii) shorter context windows produced the best performance, and (iii) different neural architectures excel under different macroeconomic regimes, motivating the ensemble approach discussed in the dashboard presentation.**

I. Introduction

GDP measures the total market value of final goods and services produced in the United States and serves as the central benchmark for evaluating economic health. Policymakers, central banks, and financial institutions rely on GDP to guide interest rate decisions, monitor recession risk, and inform fiscal planning. However, because official GDP estimates are published only quarterly and with about a one-month delay, they do not reflect current economic conditions. This lag motivates the need for "nowcasting," the practice of estimating the present or near-term state of the economy before official data become available.

Traditional statistical approaches to GDP nowcasting, including the Dynamic Factor Model (DFM) used by the Federal Reserve Bank of New York [1], summarize large panels of macroeconomic indicators into a small set of latent factors. While effective for capturing co-movement, these models rely heavily on covariance structure to determine which predictors are informative. Covariance cannot distinguish leading indicators from lagging indicators, a distinction that is critical for short-horizon forecasting.

Fig. 1 illustrates this limitation: although INDPRO and DMANEMP both exhibit high correlation with GDP, INDPRO consistently leads downturns while DMANEMP lags behind. Relying solely on covariance therefore risks selecting predictors that react after economic conditions have already changed.

This motivates the use of machine learning models that can learn temporal dependencies directly from the data, infer which predictors matter at different horizons, and naturally down-weight lagging indicators even when their correlations with GDP are high.

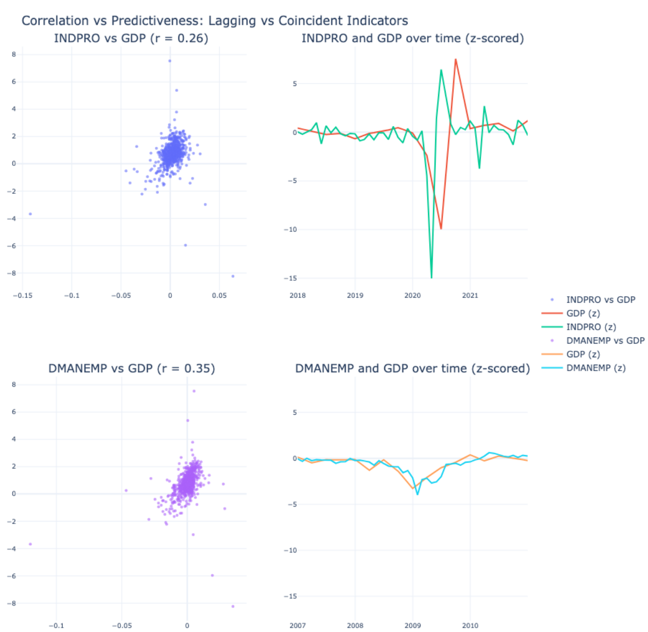

Fig. 1. Illustration of covariance limitations. INDPRO exhibits leading behavior around downturns while DMANEMP lags behind GDP, despite showing strong correlation. Covariance alone is unable to distinguish predictive from reactive indicators.

The purpose of this project is to visualize the FRED-MD indicator panel in ways that make these limitations explicit and to evaluate how neural-network architectures perform under realistic forecasting environments. The dashboard developed for this work integrates visualization, preprocessing, and model-evaluation components into a unified framework that mirrors the full workflow of GDP nowcasting. Users can interactively explore the 126 monthly FRED-MD indicators, compare raw versus transformed versions of each series, examine examples of leading and lagging behavior, and evaluate model performance across economic regimes.

In doing so, the dashboard operates both as a diagnostic tool—highlighting where covariance-based methods fail—and as an experimental platform for assessing how different neural architectures capture temporal structure in macroeconomic data. By combining high-frequency indicators, sequence-based modeling, and interactive visualization, this work provides insight into which features and model designs matter most for real-time forecasting and how these insights change under stable versus turbulent economic conditions.

II. Data Description

A. Target Variable: Real GDP

Real GDP (GDPC1) is obtained from the Federal Reserve Economic Data (FRED) and transformed into quarter-over-quarter (QoQ) growth via log-differences, following conventions in the nowcasting literature [2]. The GDP series spans 1960Q1–2024Q2, aligned to monthly frequency to match the predictor panel.

Because GDP releases are delayed and undergo revisions, nowcasting aims to predict the most recent quarter before official publication.

B. Predictor Panel: FRED-MD

The predictor set consists of 126 monthly macroeconomic indicators grouped into eight categories:

- Output & Income
- Labor Market
- Housing
- Consumption & Orders
- Money & Credit
- Interest & Exchange Rates
- Prices
- Stock Market

Each series comes with a recommended transformation (TCODE) to achieve stationarity. These transformations include first differences, log differences, and detrended values, ensuring the predictors have stable mean/variance when passed into statistical or machine-learning models.

The dataset spans 774 months (1960–2024), with 901 missing entries after transformation. In accordance with macroeconomic modeling practice, no global interpolation was applied during preprocessing; instead, NaN handling occurred only within extracted input sequences using forward-fill/backward-fill heuristics.

III. Technologies and Implementation

A. Dashboard Framework

The interactive dashboard is built using the Dash framework, a Python tool that makes it easy to create web-based data applications. Dash connects Python code directly to visual elements in the browser, so users can change predictor categories, data transformations, and evaluation settings without needing to refresh or run separate scripts. The dashboard's visuals are created using Plotly, which provides flexible tools for making line charts, heatmaps, scatterplots, and interactive model comparisons. These features work well for macroeconomic data, which involve many time series and require exploring patterns that change over time.

Dash callbacks are used to make the dashboard interactive. When a user changes an indicator category, sequence length, or information set, the dashboard automatically updates the plots on the screen. This makes it easy to test out different ideas, such as checking which indicators lead or lag GDP, how transformations affect stability, or how model results change under different settings. The layout of the dashboard matches the workflow of this paper, beginning with indicator exploration (Figs. 1, 2, 3, 4), then sequence and information-set visualization (Figs. 5 and 6), and lastly, model evaluation results (Figs. 9 and 10, Tables I and II).

B. Data Preprocessing

Data preprocessing is handled using Pandas and NumPy, which make it easy to work with panel-shaped data (774 monthly observations across 126 predictors). Preparing the FRED-MD dataset involves several steps:

- TCODE-based transformations to enforce stationarity,
- Missing-value handling applied within sequence windows,
- Alignment of monthly indicators to quarterly GDP targets,
- Construction of input sequences of different lengths and information sets.

Since forecasting models rely on the correct order of events, preserving the time index is important. Pandas' DateTimeIndex helps maintain this order and supports the sliding-window approach used to build the input sequences.

C. Model Training

The artificial neural network (ANN) models were built and trained using PyTorch, which makes it easy to define and experiment with different architectures (MLP, CNN, RNN, LSTM, and GRU). PyTorch allows quick testing of many model settings, including:

- four different sequence lengths,
- three information sets,
- several hyperparameters (learning rate, dropout, L1 penalty, hidden size), and
- thirty random seeds for each configuration.

PyTorch's DataLoader and tensor operations help batch and train thousands of input sequences efficiently. This setup enables me to compare the models systematically across both stable (PRE) and turbulent (FULL) economic periods.

D. Integration and Pipeline

The technologies are connected through a unified workflow:

1) Pandas loads and preprocesses the data, returning stationary and normalized variants.
2) NumPy handles vectorized transformations needed for sequence extraction.

3) PyTorch trains the forecasting models using the prepared sequences.
4) Plotly/Dash displays the results through interactive visualization components.

This setup allows the system to work smoothly from start to finish. It can run large forecasting experiments with the neural networks, and it also serves as a user-friendly dashboard that helps users explore the data, understand patterns, and see the strengths and weaknesses of different indicators and models.

IV. Visualization Techniques

A. Line Plots with Category Filters

On the first tab of the dashboard—the Overview tab—there is a brief description of the research motivation and a summary of the dataset. At the bottom of this page is an interactive visualization tool for the FRED-MD predictor panel, where users can select individual indicators or entire categories (e.g., Labor Market) and view them in their raw, stationary, or normalized forms. The interactive line plot in Fig. 2 demonstrates this for the Output and Income category, with a range slider that allows users to zoom in on specific time periods such as recessions or recoveries.

This tool goes beyond simple data inspection by allowing users to closely examine individual indicators and how their behavior relates to GDP over time. By turning indicators on and off, the plot makes it easy to spot leading, coincident, and lagging patterns in the data. The range slider also makes it simple to focus on different time periods and see how indicators respond during events like downturns or recoveries. This was especially helpful during model development, as it allowed me to quickly and effectively test whether high-correlation indicators were actually useful for prediction and to confirm the need for ANN models that can learn these timing relationships.

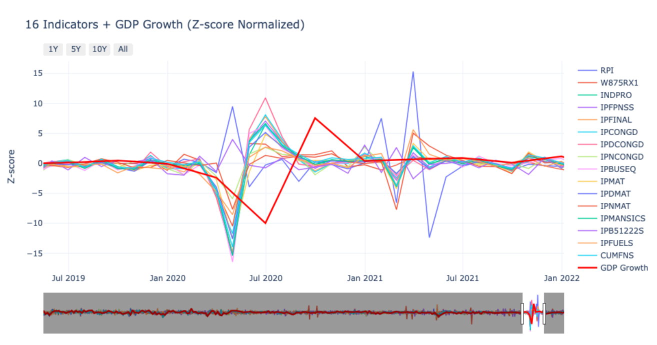

Fig. 2. Interactive line plot showing Output and Income indicators. The range slider at the bottom allows users to select a custom time window. Here, a 3-year period (2019–2022) is displayed.

B. Correlation Plot & Heatmaps

The correlation plot (Fig. 3) and heatmap (Fig. 4) show the covariance between the predictor panel and GDP growth. Methods like these form the basis of traditional feature selection approaches such as PCA. However, as highlighted in this project, high absolute correlation does not imply predictive usefulness. Many indicators with strong correlation to GDP are in fact lagging variables, as shown in Fig. 1, emphasizing the need for models that can learn which features provide true forward-looking, predictive information.

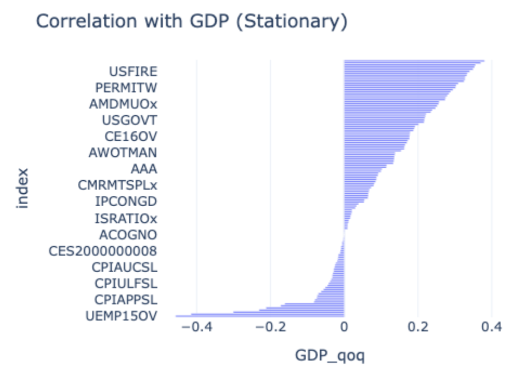

Fig. 3. Correlation plot showing the relationship between stationary indicators and GDP growth. High correlation does not guarantee predictive value, as many indicators are lagging rather than leading.

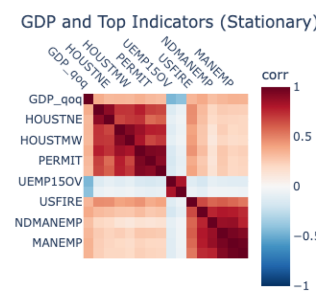

Fig. 4. Correlation heatmap of stationary indicators and GDP growth. Blocks of high correlation reveal shared cycles but do not indicate whether an indicator leads or lags GDP.

These visualizations were chosen because they represent the standard tools used in statistical modeling, particularly methods such as PCA that depend on covariance structure. They are useful for highlighting broad co-movements and identifying groups of indicators that behave similarly, but they also make the limitations of such methods clear. By displaying promising but ultimately lagging relationships, the correlation plot and heatmap help illustrate why covariance alone cannot guide effective feature selection and why temporal models are required to uncover indicators with true predictive, leading behavior.

C. Sequence & Info-set Visualization

The dashboard illustrates how training inputs change across different sequence lengths and information sets. For each sequence length $L \in \{8, 18, 36, 48\}$ months, the system displays the most recent L observations used to predict the target quarter. As shown in Fig. 5, shorter windows place greater weight on recent movements, while longer windows capture multi-year business-cycle patterns and smoother macroeconomic trends.

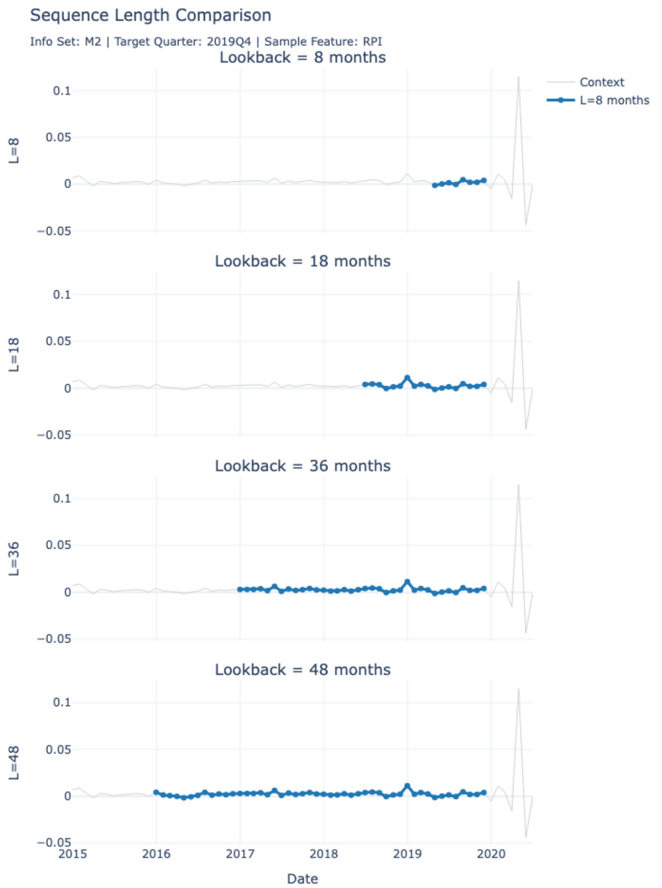

Fig. 5. Comparison of input sequence lengths used for model training. Each subplot corresponds to a different lookback window ($L \in \{8, 18, 36, 48\}$), highlighting how shorter sequences emphasize recent dynamics while longer sequences capture broader macroeconomic cycles. The vertical line marks the start of the target quarter.

The dashboard also visualizes the real-time information available for forecasting under each information set (M1–M3). As shown in Fig. 6, M1 contains only the first month of the target quarter, M2 contains the first two months, and M3 includes all three. These views make clear how the amount of usable data expands throughout the quarter and how training inputs differ across realistic forecasting scenarios that account for publication delays.

Each neural network architecture was trained on all combinations of sequence lengths and information sets, enabling direct comparison of how temporal depth and real-time data availability influence predictive performance.

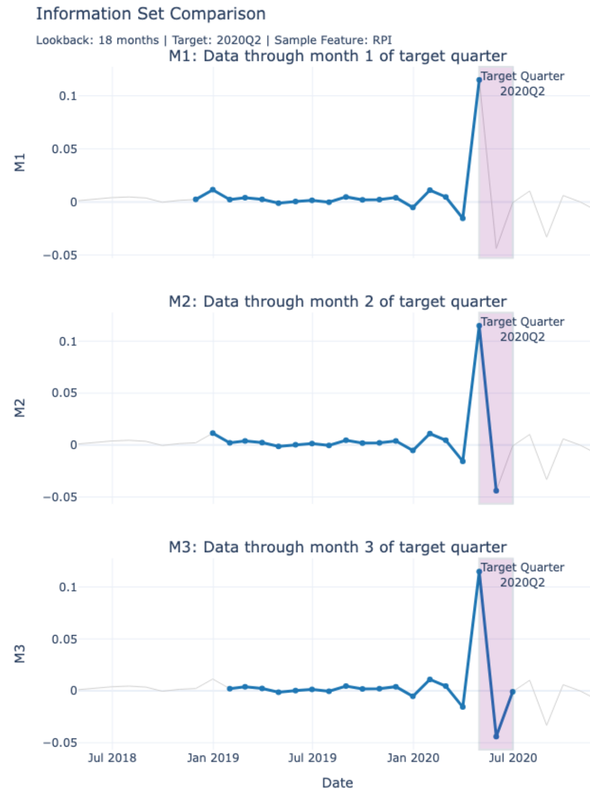

Fig. 6. Information-set visualization for real-time forecasting. Each subplot (M1, M2, M3) shows the portion of data available for predicting a given target quarter, reflecting standard real-time data-release constraints in GDP nowcasting.

D. *Model Performance Visuals*

The model performance plots compare actual GDP growth with predictions from the naive benchmark, the DFM benchmark, and the five ANN architectures (MLP, CNN, RNN, LSTM, and GRU). Results are shown across two evaluation windows: (i) PRE (2012–2019), a period of relative economic stability, and (ii) FULL (2012–2024), which includes the COVID-19 recession and represents a more turbulent regime.

These visuals, together with the summary tables, allow users to see how each architecture performs relative to the benchmarks under different economic environments and training configurations. They highlight which models benefit from longer or shorter sequences, which handle limited information more effectively, and which remain stable during periods of macroeconomic stress.

Line plots were chosen for the performance section because they provide the most intuitive and informative way to compare predicted and actual GDP values over time. Time-series behavior is best understood when users can observe turning points, delays, and trend alignment directly. Alternative visualizations, such as bar charts or scatterplots, are less effective for capturing these temporal patterns.

The RMSE evaluation tables complement the line plots by providing a clear quantitative comparison across models, sequence lengths, and information sets. While line plots offer qualitative insight into model behavior, the tables summarize overall accuracy in a compact, interpretable format. Using both techniques gives users a balanced view of model performance, combining visual intuition with precise numerical evaluation.

V. Methodology

Our modeling strategy draws on the architecture-comparison framework introduced by Stricker et al. [2], who demonstrate that different neural networks learn complementary temporal patterns relevant for GDP nowcasting. Following this motivation, we evaluate five ANN architectures (MLP, CNN, RNN, LSTM, GRU) and train each on fixed-length input sequences extracted from the FRED-MD panel. We test all 5 models across configurations of 4 different sequence lengths and 3 different info-sets.

A. Sequence Construction

Following the approach in Stricker et al. [2], the predictor panel was converted into fixed-length input sequences to train the neural network architectures on temporal dynamics. For each target quarter q, a sliding window of the previous L months of indicators was extracted across all 126 transformed series, as illustrated in Fig. 5. The corresponding real-time information available for each quarter (M1–M3) is shown in Fig. 6, which determines how much of the sequence can be used under realistic forecasting constraints.

B. Model Training

For each combination of lookback length L and information set M, models were trained using an expanding-window approach consistent with macroeconomic forecasting standards. Hyperparameters—including learning rate, dropout rate, L1 penalty, and hidden-layer width—were tuned via grid search over the PRE (2012–2019) validation window. Each configuration was trained across 30 seeds to ensure stable performance estimates and reduce variance in neural network outcomes.

VI. Performance Metrics and Benchmarks

Model performance is evaluated using relative Root Mean Squared Error (RMSE) with respect to two baseline forecasting methods: a naive benchmark and a Dynamic Factor Model (DFM) benchmark.

A. Naive Benchmark

The naive model predicts GDP growth for quarter q using the most recently observed GDP value, i.e.,

$$\hat{y}_q^{\text{naive}} = y_{q-1}.$$

This corresponds to a lag-1 persistence forecast and serves as a lower bound on model sophistication. Models with relative RMSE below 1 outperform the naive benchmark.

B. Dynamic Factor Model Benchmark

The DFM represents a widely used statistical approach for macroeconomic nowcasting. It summarizes comovement among large predictor panels using latent factors estimated within a state-space framework. The benchmark model used in this study follows the implementation outlined in the Statsmodels documentation [3]. Relative RMSE below 1 indicates that the neural architecture outperforms the DFM baseline.

Figs. 7 and 8 show the naive and DFM benchmark predictions over the PRE and FULL evaluation windows, respectively. These plots illustrate how the DFM captures broad business-cycle movements more accurately than the naive lag-1 forecast, particularly around turning points.

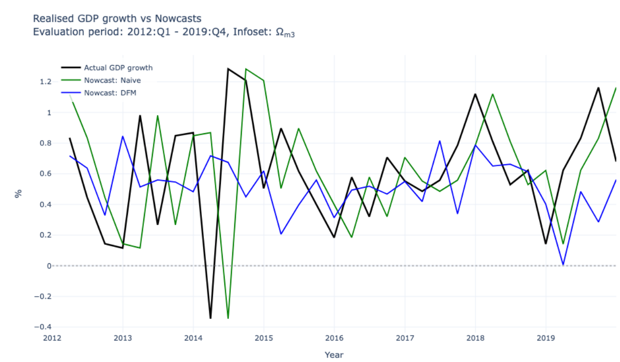

Fig. 7. Benchmark models for PRE eval window.

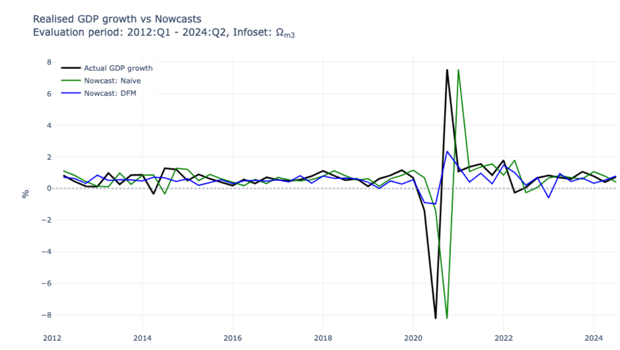

Fig. 8. Benchmark models for FULL eval window.

C. Relative RMSE

For each model, information set, and evaluation window, performance is summarized using

$$\Omega = \frac{\text{RMSE}_{\text{model}}}{\text{RMSE}_{\text{benchmark}}},$$

where the benchmark is either the naive or DFM model. This metric enables comparison across heterogeneous forecasting approaches and highlights conditions under which neural architectures offer performance gains.

VII. Results

This section presents model performance under two evaluation windows: a stable economic period (PRE: 2012–2019) and a turbulent period including the COVID-19 recession (FULL: 2012–2024). These two windows allow us to assess the sensitivity of each neural architecture to structural breaks, volatility spikes, and unusual macroeconomic behavior. All model performance is evaluated relative to the naive and DFM benchmarks described in Section VI.

Fig. 9 and Table I summarize the PRE-window results. During this relatively stable regime, several architectures perform competitively, with CNN models frequently achieving the lowest relative RMSE across the M1–M3 information sets. This suggests that CNNs benefit from their ability to extract localized temporal patterns, which remain consistent during periods of moderate macroeconomic variation.

Fig. 10 and Table II show performance over the FULL window, which includes the sharp contraction and recovery associated with the COVID-19 shock. In this regime, no single architecture consistently outperforms the benchmarks across all information sets. Performance deteriorates substantially for all models, reflecting the difficulty of learning from extreme, short-lived disruptions that differ from historical patterns. LSTM and GRU models show improved robustness for some information sets, though not uniformly.

These results demonstrate that architecture performance is highly regime-dependent: the model that performs best during stable conditions may not generalize to turbulent periods. Consequently, relying on a single model may be insufficient for real-time GDP nowcasting, motivating approaches that adapt model choice to economic regime or combine architectures through ensembling.

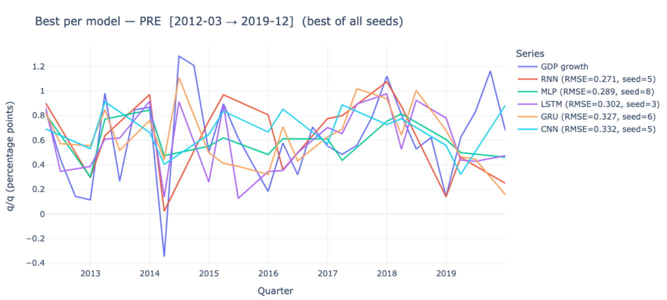

Fig. 9. Best-performing model per architecture during the PRE evaluation window (2012–2019). Under stable economic conditions, several architectures perform comparably, with CNNs showing strong performance across sequence lengths.

VIII. Discussion

Overall, I am satisfied with the visualization results produced in this project. My original goal was to create a dashboard that could clearly communicate the findings of my GDP nowcasting research, and the line plots succeeded in making model behavior and leading–lagging dynamics easy for users to explore and understand. Because I began with a

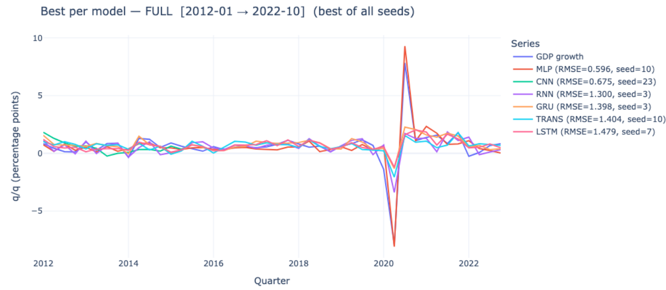

Fig. 10. Best-performing model per architecture during the FULL evaluation window (2012–2024), which includes the COVID-19 recession. Model performance varies substantially across architectures, illustrating the difficulty of capturing extreme macroeconomic shocks.

TABLE I
RELATIVE RMSE PERFORMANCE DURING PRE EVALUATION WINDOW (2012–2024)

Model	$\Omega m1$ **Naive**	$\Omega m1$ **DFM**	$\Omega m2$ **Naive**	$\Omega m2$ **DFM**	$\Omega m3$ **Naive**	$\Omega m3$ **DFM**
MLP (ℓ=8)	0.576	0.823	0.569	0.809	0.582	0.812
MLP (ℓ=18)	0.547	0.782	0.512	0.728	0.626	0.873
MLP (ℓ=36)	0.596	0.851	0.554	0.788	0.667	0.930
MLP (ℓ=48)	0.684	0.977	0.649	0.924	0.720	1.004
CNN (ℓ=8)	0.647	0.924	0.586	0.834	0.607	0.846
CNN (ℓ=18)	0.638	0.912	0.734	1.044	0.665	0.928
CNN (ℓ=36)	0.665	0.949	0.832	1.184	0.677	0.943
CNN (ℓ=48)	0.681	0.973	0.781	1.112	0.724	1.009
RNN (ℓ=8)	0.511	0.730	0.479	0.681	0.641	0.894
RNN (ℓ=18)	0.542	0.774	0.522	0.743	0.668	0.931
RNN (ℓ=36)	0.536	0.766	0.493	0.702	0.694	0.968
RNN (ℓ=48)	0.603	0.861	0.593	0.844	0.660	0.921
LSTM (ℓ=8)	0.586	0.837	0.583	0.830	0.674	0.940
LSTM (ℓ=18)	0.572	0.817	0.594	0.844	0.623	0.869
LSTM (ℓ=36)	0.572	0.816	0.563	0.801	0.643	0.896
LSTM (ℓ=48)	0.534	0.763	0.601	0.855	0.612	0.854
GRU (ℓ=8)	0.643	0.919	0.607	0.864	0.670	0.935
GRU (ℓ=18)	0.624	0.892	0.586	0.834	0.621	0.866
GRU (ℓ=36)	0.583	0.832	0.630	0.896	0.647	0.902
GRU (ℓ=48)	0.579	0.827	0.615	0.875	0.646	0.900

strong focus on illustrating the modeling pipeline—sequence lengths, information sets, and prediction outputs—many of the visuals naturally took the form of time-series line plots, which remain the most intuitive and informative choice for this type of data.

However, this initial approach limited me to many line plots, which pushed me to expand the scope of the dashboard to include covariance-based visuals such as the correlation plot and heatmap. These figures helped justify the use of ANN architectures by showing the limitations of relying solely

TABLE II
RELATIVE RMSE PERFORMANCE DURING FULL EVALUATION WINDOW (2012–2024)

Model	$\Omega m1$ **Naive**	$\Omega m1$ **DFM**	$\Omega m2$ **Naive**	$\Omega m2$ **DFM**	$\Omega m3$ **Naive**	$\Omega m3$ **DFM**
MLP (ℓ=8)	0.802	1.029	0.797	0.805	0.645	0.880
MLP (ℓ=18)	0.758	0.974	0.806	0.815	0.754	1.029
MLP (ℓ=36)	0.690	0.887	0.762	0.770	0.627	0.855
MLP (ℓ=48)	0.779	1.000	0.835	0.844	0.674	0.919
CNN (ℓ=8)	0.814	1.046	0.842	0.851	0.714	0.974
CNN (ℓ=18)	0.768	0.986	0.882	0.892	0.834	1.137
CNN (ℓ=36)	0.879	1.129	0.963	0.973	0.780	1.063
CNN (ℓ=48)	0.843	1.082	1.034	1.045	0.819	1.117
RNN (ℓ=8)	0.791	1.016	0.896	0.905	0.707	0.963
RNN (ℓ=18)	0.760	0.976	0.891	0.901	0.669	0.912
RNN (ℓ=36)	0.752	0.966	0.818	0.826	0.595	0.811
RNN (ℓ=48)	0.714	0.918	0.840	0.849	0.667	0.910
LSTM (ℓ=8)	0.821	1.054	0.876	0.885	0.731	0.996
LSTM (ℓ=18)	0.751	0.965	0.867	0.876	0.698	0.952
LSTM (ℓ=36)	0.759	0.975	0.845	0.854	0.612	0.835
LSTM (ℓ=48)	0.760	0.976	0.856	0.865	0.660	0.899
GRU (ℓ=8)	0.747	0.959	0.794	0.802	0.728	0.992
GRU (ℓ=18)	0.760	0.977	0.836	0.845	0.697	0.951
GRU (ℓ=36)	0.763	0.981	0.816	0.825	0.564	0.770
GRU (ℓ=48)	0.795	1.021	0.829	0.838	0.582	0.794

on correlation for feature selection. While these additions strengthened the conceptual foundation of the project, they were developed later in the process and therefore received less time for planning and design.

If beginning this project again from scratch, I would frame the visualization goals more broadly from the start, not only to illustrate the modeling methodology, but also to visualize the motivation for the models themselves. Doing so would allow for a more balanced exploration of visualization types, such as dimensionality-reduction maps and feature-importance visuals, and would provide an even clearer bridge between the problem context, the modeling approach, and the resulting forecasts.

IX. Conclusion

This paper presents an interactive dashboard and modeling framework for visualizing FRED-MD indicators and evaluating neural-network architectures for real-time GDP nowcasting. Through a combination of exploratory visualizations, sequence constructions, and model evaluations under realistic information-set constraints, the dashboard highlights the limitations of traditional covariance-based methods and demonstrates the benefits of models capable of learning feature importance.

The results show that no single architecture performs best across all economic regimes. CNNs excel in stable environments, whereas recurrent models show greater robustness during turbulent periods. These findings underscore the importance of regime awareness and suggest that ensemble-based forecasting strategies may offer substantial benefits.

References

[1] B. Bok, D. Caratelli, D. Giannone, A. Sbordone, and A. Tambalotti, "Macroeconomic nowcasting and forecasting with the NY Fed staff nowcast," *Federal Reserve Bank of New York Staff Reports*, no. 967, 2023, accessed: December 8, 2025. [Online]. Available: https://fedinprint.org/item/fednls/96733

[2] Y.-L. Stricker, L. Bögelein, S. Nardo, S. Raviv, and E. Rolf, "Gdp nowcasting with pmi text using large language models," *arXiv preprint arXiv:2304.05805*, 2024. [Online]. Available: https://arxiv.org/abs/2304.05805

[3] Statsmodels Developers, "Dynamic factor models: Coincident index example," https://www.statsmodels.org/dev/examples/notebooks/generated/statespace_dfm_coincident.html, 2024, accessed: December 8, 2025.