Research on Deep Learning-based Economic Forecasting Algorithms

Yuanqi Geng[1,*], Tingting Yang[2]
[1]College of Business and Economics, Australian National University, Canberra 2600, Australia
[2]School of Electronic Engineering, Jiangsu Ocean University, Lianyungang 222005, China
*Corresponding email: 240065760@qq.com

Abstract

Deep learning models are increasingly being applied to macroeconomic forecasting. However, most studies report aggregate accuracy without isolating the contribution of individual architectural components. This paper proposes a multi-scale attention-augmented bidirectional gated recurrent unit with a direct linear residual branch and introduces a controlled walk-forward ablation protocol to quantify the marginal value of each component. Experiments on eleven monthly United States (U.S.) indicators from the Federal Reserve Economic Data - monthly database (FRED-MD), spanning March 1959 to September 2019, show that the linear residual branch accounts for most of the predictive gain. In contrast, multi-scale convolution and temporal attention each contribute less than a 2.0% additional reduction in root mean square error (RMSE). These results suggest that the deep architectural components offer only limited marginal benefit on this dataset. This diagnostic finding clarifies the practical role of deep learning components in low-signal macroeconomic data and establishes a reproducible evaluation standard for future architectural studies.

Keywords

Deep learning, Economic forecasting, Multi-scale convolution, Temporal attention, Walk-forward validation

Introduction

Macroeconomic forecasting supports monetary policy, fiscal planning, and portfolio allocation. Linear autoregressive models remain standard tools because they are transparent and perform reliably on stationary aggregate series. In parallel, a growing body of literature applies machine learning to the same task. Large-scale forecasting competitions show that no single model family dominates across series and horizons [1]. Kant et al. reported that machine learning improves GDP nowcasting relative to econometric benchmarks, with the magnitude of improvement depending on the predictor set and the evaluation window [2]. Chapman and Desai utilized payments data with machine learning and found that additional high-frequency information improves macroeconomic predictions [3]. In asset pricing, Chen et al. demonstrated that deep networks price assets more accurately than linear models, although the improvement is modest and concentrated in nonlinear interactions [4]. Despite this activity, most studies report the aggregate accuracy of the final model without isolating the marginal contribution of each architectural component. A reader cannot determine whether the gain originates from multi-scale convolution, attention, the recurrent encoder, or the added linear path. Schlieper et al. benchmarked neural architectures for univariate forecasting and found that simpler recurrent variants often match more elaborate designs when training is controlled [5]. Analyses of the M6 competition reached a similar conclusion: hybrid and ensemble designs often win, but the source of the improvement is rarely decomposed [6].

This paper addresses that gap by proposing a dual-branch architecture and reporting a component-wise ablation under a controlled walk-forward protocol. The architecture, termed multi-scale attention bidirectional gated recurrent unit (MSA-BiGRU), combines three deep components with a direct linear residual branch: multi-scale causal convolution with kernel sizes of 3, 7, and 13. A bidirectional gated recurrent unit with 32 units per direction; and additive temporal attention over the encoded sequence. The linear branch receives the same input window, and the final prediction sums the deep and

linear outputs.

The contributions are threefold. First, the dual-branch design separates nonlinear and linear pathways, fitting macroeconomic mean reversion through the linear branch while the deep branch captures residual nonlinear dynamics. Second, a walk-forward ablation protocol removes one component at a time and re-estimates the model on identical rolling windows for controlled attribution of performance. Third, experiments on FRED-MD show that the linear residual branch accounts for most of the predictive gain, with multi-scale convolution and temporal attention each contributing less than a 2.0% additional reduction in RMSE.

Related work and research gap

Machine learning methods for macroeconomic forecasting have been evaluated primarily through benchmark comparisons against linear and classical models. Kant et al. and Chapman and Desai reported gains in nowcasting and short-horizon predictions, but both studies treated the learning model as a single block. Studies on inflation reach similar conclusions. Stoneman and Duca found that deep learning improved U.S. inflation forecasts during the coronavirus disease 2019 period [7]. Vancsura et al. reported comparable gains for euro area inflation and its subcomponents [8]. These studies establish that nonlinear models can be beneficial but do not isolate which architectural features produce the benefits.

Within deep learning for time series, convolutional, recurrent, and attention-based architectures have each been proposed. Semenoglou et al. reframed time series as images and applied convolutional networks to forecasting [9]. Schlieper et al. benchmarked recurrent and convolutional architectures under a common protocol and found that gated recurrent units are competitive on short, univariate series. Xiao et al. designed a hierarchical spatio-temporal attention mechanism that weights relevant time steps and channels [10]. Yang and Gonsalves proposed a cross-time and cross-dimensional transformer for multivariate forecasting [11]. Jiang et al. combined a temporal fusion transformer with a penalised quantile loss for probabilistic price forecasting [12]. de Vilmarest and Werge developed an adaptive volatility method for probabilistic forecasts [13]. Lin introduced a progressive network for multi-horizon forecasting [14]. Dowlut and Gobin-Rahimbux reviewed deep learning for tourism demand and noted uneven benchmarking practices [15]. Graph neural networks extend these ideas to relational structure [16].

Interpretability research provides tools to inspect what a model has learned [17]. While uncertainty quantification methods estimate confidence around deep forecasts [18]. However, these tools address post-hoc explanation or confidence estimation, not the forward-looking question of which design choices should be included at construction time. Few macroeconomic forecasting studies perform a controlled ablation in which components are removed one at a time and all models are retrained on identical rolling windows. Therefore, predictive accuracy differences are assessed here with the Diebold-Mariano test, following standard practice in recent forecasting competitions. Ensemble and forecast-combination strategies can further obscure component contributions, which motivates the component-wise design used in this study [19,20].

Proposed method

Problem formulation

Let x_t denote the m-dimensional vector of transformed macroeconomic indicators at month t. The model receives a sliding window of T months and produces an h-step-ahead forecast. The input matrix is

$$X_t = [x_{t-T+1}, x_{t-T+2}, \dots, x_t] \in R^{m \times T} \quad (1)$$

Where m = 11 and T = 24. The goal is to learn a mapping f such that

$$\hat{y}_{t+h} = f(X_t; \theta) \quad (2)$$

Where θ denotes all trainable parameters and h ∈ {1, 3, 6}.

Architecture overview

Figure 1 illustrates the overall architecture of the proposed model. Specifically, the input window is fed simultaneously into two parallel processing branches. The upper nonlinear branch first applies multi-scale causal convolution to capture local patterns across different receptive fields. Followed by a bidirectional gated recurrent unit (GRU) encoder that models sequential dependencies in both directions, and then a temporal attention mechanism that adaptively weights the encoded representations over time. In parallel, the lower linear branch consists of a single linear layer applied to the

flattened input window, which preserves the direct linear relationship between the input and the target. Finally, the outputs from both branches are summed within a fusion layer to generate the final forecast.

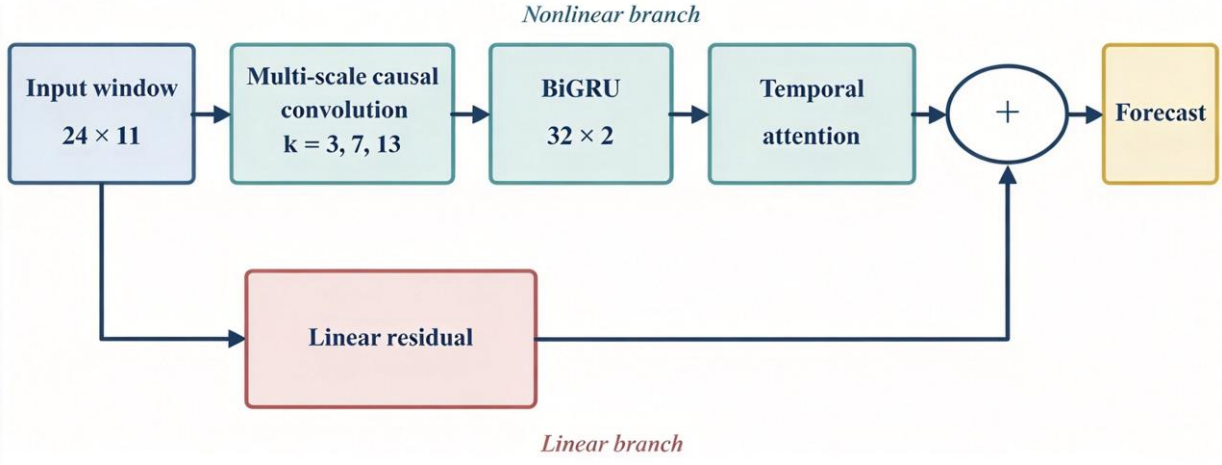


Figure 1. Architecture of the proposed MSA-BiGRU model.

The dual-branch design is motivated by the observation that macroeconomic series are dominated by persistent trends and mean reversion, which a linear model captures well. The deep branch is therefore freed to model short-lived nonlinear deviations without being forced to reproduce the trend itself. This separation also enables a clean ablation: Removing the deep branch leaves the linear model intact, and removing the linear branch exposes the deep model alone.

Multi-scale temporal convolution

The convolution layer applies three parallel one-dimensional causal convolutions with kernel sizes k∈{3, 7, 13} to the input matrix. For kernel size k, the output at position i is

$$z_i^{(k)} = \sigma(W_k * x_{i:i+k-1} + b_k) \quad (3)$$

Where W_k and b_k are the weight and bias for kernel size k, $*$ denotes causal convolution, and σ is the rectified linear unit. The three outputs are concatenated along the channel dimension. Figure 2(a) illustrates how the three kernel spans cover different temporal extents of the 24-month input window.

Multi-scale convolution serves a dual purpose. Short kernels capture monthly momentum and noise, medium kernels capture quarterly dynamics, and long kernels capture annual seasonality and business-cycle persistence. Consequently, this layer yields a richer temporal representation than a single fixed kernel, without the need for explicit feature engineering.

BiGRU encoding and temporal attention

The concatenated multi-scale features are passed through a bidirectional gated recurrent unit with 32 hidden units per direction. For each direction, the GRU computes a reset gate, an update gate, and a candidate hidden state at the current time step s:

$$r_s = \sigma(W_r[h_{s-1}, x_s] + b_r) \quad (4)$$

$$z_s = \sigma(W_z[h_{s-1}, x_s] + b_z) \quad (5)$$

$$\tilde{n}_s = tanh(W_n[r_s \odot h_{s-1}, x_s] + b_n) \quad (6)$$

$$h_s = (1 - z_s) \odot h_{s-1} + z_s \odot \tilde{n}_s \quad (7)$$

Where ⊙ denotes element-wise multiplication and [a, b] denotes concatenation. The bidirectional encoder produces a hidden state sequence $h_1, h_2, \ldots, h_T$, each of dimension 64 after concatenating forward and backward states.

Temporal attention then computes a scalar score for each position s:

$$e_s = tanh(W_a h_s + b_a) \quad (8)$$

The scores are normalised to obtain attention weights:

$$a_s = exp(e_s) / \sum_{j=1}^{T} exp(e_j) \quad (9)$$

The context vector is the weighted sum of hidden states:

$$c = \sum_{s=1}^{T} a_s h_s \quad (10)$$

Figure 2(b) shows how the temporal attention mechanism assigns weights α_s to individual time steps. A larger weight indicates that the corresponding hidden state contributes more to the context vector. As can be seen, the attention weights are not uniformly distributed across the 24-month window. Instead, they exhibit a clear peak around the 19th time step (α_s=0.31), indicating that the model focuses most strongly on recent but not necessarily the most recent observations, while earlier time steps receive comparatively smaller weights. This non-uniform weighting allows the model to selectively emphasize the most informative temporal positions when forming the context representation.

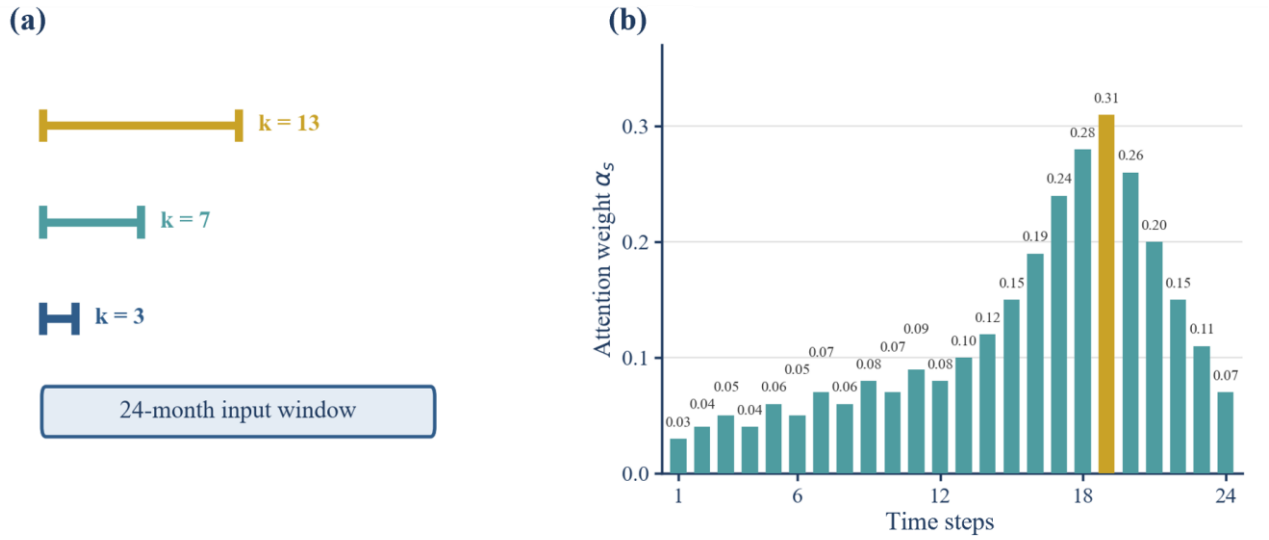


Figure 2. Overview of multi-scale convolution and temporal attention. (a) Multi-scale kernel spans over the 24-month input window; (b) temporal attention weighting over hidden states.

Residual fusion

The deep branch output is the context vector c. The linear residual branch applies a single weight vector to the flattened input window. The final forecast is

$$\hat{y}_{t+h} = w_d{}^T c + b_d + w_r{}^T vec(X_t) + b_r \quad (11)$$

where vec(X_t) flattens the input matrix into a column vector, w_d and b_d are the deep branch output weights, and w_r and b_r are the linear residual weights. Additive fusion is chosen over gated fusion because it guarantees that the model always contains a pure linear forecast as a special case, preventing the deep branch from displacing the trend signal during training. The model is trained by minimising mean squared error over the training window using the Adam optimiser with a learning rate of 0.001 and early stopping after 15 epochs without validation loss improvement.

Experimental protocol

Data and preprocessing

The dataset is drawn from the FRED-MD database. Eleven monthly United States indicators are selected to cover production, employment, prices, money, housing, consumption, financial markets and labour market flows. Table 1 lists the variables and their stationarity transformations. All series are transformed to achieve stationarity, then standardised to zero mean and unit variance using statistics computed on each estimation window only.

Table 1. Variables and stationarity transformations.

Variable	Description	Transformation
INDPRO	Industrial production index	Δln
PAYEMS	Nonfarm payroll employment	Δln
UNRATE	Civilian unemployment rate	Δ
CPIAUCSL	Consumer price index	Δ²ln
FEDFUNDS	Effective federal funds rate	Δ
GS10	10-year Treasury yield	Δ
M2SL	M2 monetary aggregate	Δ²ln
HOUST	New housing starts	ln
RETAILx	Retail sales ex. food services	Δln
S&P 500	S&P 500 composite index	Δln
CLAIMSx	Initial unemployment claims	Δln

After transformation, the dataset contains 727 usable monthly observations from March 1959 to September 2019. The target variable is the transformed industrial production index INDPRO. Each observation consists of a 24-month input window and an h-step-ahead target, where h ∈ {1, 3, 6}.

Walk-forward validation

Figure 3 illustrates the walk-forward validation protocol. Five non-overlapping forecast blocks are used, starting in 2005, 2008, 2011, 2014 and 2017, each spanning 36

months. For each block, the model is estimated on all data available before the block start and then produces forecasts for the 36 months within that block. The estimation window thus grows with each successive block, reflecting the expanding-window convention commonly used in macroeconomic forecasting.

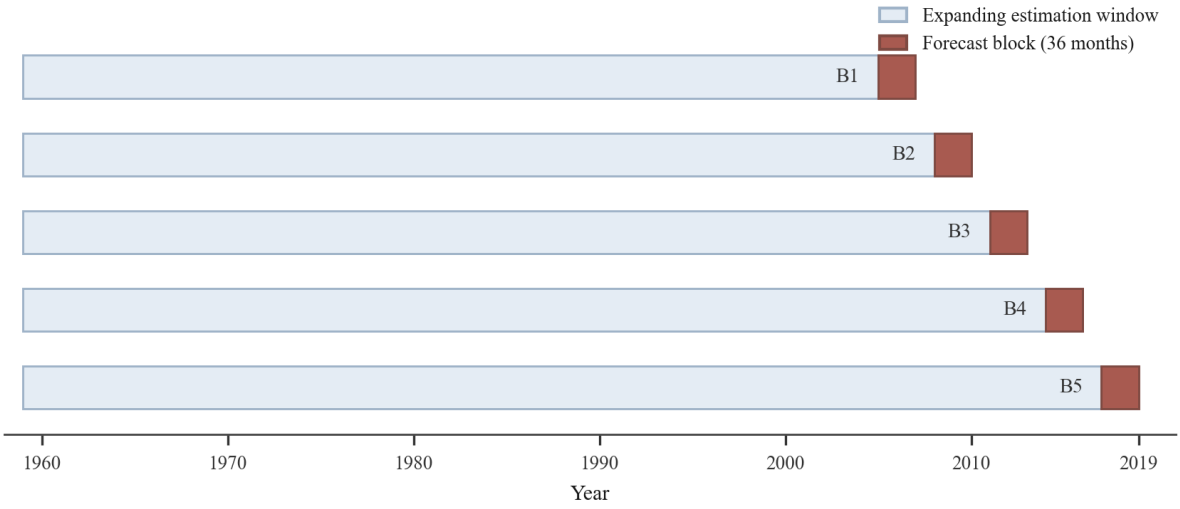


Figure 3. Walk-forward validation with five forecast blocks of 36 months each.

This protocol yields 176 test origins at h=1, and 171 origins at h=3 and h=6 because the trailing observations of the final block are exhausted at longer horizons. All models, including baselines, are re-estimated from scratch on every block using the same estimation window. No data from the forecast block is used for tuning or standardisation.

Baseline models and implementation

Five baselines are compared. autoregressive integrated moving average (ARIMA) (1,0,1) with a constant is estimated by maximum likelihood on each training window. Ridge regression with the regularisation parameter selected by 5-fold cross-validation on each estimation window serves as the linear benchmark. A two-layer stacked long short-term memory (LSTM) and a two-layer GRU each use 64 hidden units. A two-layer Transformer encoder uses 4 heads and a model dimension of 64. The proposed MSA-BiGRU contains 31,138 trainable parameters, compared with 39,809 for the GRU, 53,057 for the LSTM and 69,441 for the Transformer. The lower parameter count reduces overfitting risk on a dataset of this size.

All deep models use the Adam optimiser with a learning rate of 0.001, a batch size of 32 and a maximum of 200 epochs. Early stopping monitors validation loss computed on the last 20.0% of each estimation window. Evaluation metrics are RMSE, mean absolute error (MAE), the coefficient of determination (R^2), and directional accuracy (DA). Statistical significance of loss differences is assessed with the Diebold-Mariano test using a two-sided alternative at the 5% level.

Results and discussion

One-month forecast accuracy

Table 2 reports one-month-ahead (h=1) forecast accuracy on the pooled test set. Ridge regression achieves the lowest RMSE of 0.658, followed by GRU at 0.682 and ARIMA at 0.689. The proposed MSA-BiGRU records an RMSE of 0.712, which is higher than the linear benchmark but lower than the standalone LSTM and Transformer. Standard deviations across the five forecast blocks are reported in parentheses for the deep learning models.

Table 2. One-month-ahead forecast accuracy on the pooled test set.

Model	RMSE	MAE	R^2	DA (%)
ARIMA	0.689	0.472	0.089	67.8
Ridge	0.658	0.470	0.151	65.3
LSTM	0.726 (0.001)	0.493 (0.004)	-0.034 (0.003)	60.8 (0.4)
GRU	0.682 (0.011)	0.474 (0.006)	0.089 (0.029)	67.0 (3.6)
Transformer	0.744 (0.015)	0.504 (0.008)	-0.085 (0.043)	59.7 (4.4)
MSA-BiGRU	0.712 (0.030)	0.487 (0.014)	0.005 (0.084)	60.2 (4.8)

The negative R^2 values for LSTM and Transformer indicate that these models perform worse than a constant-mean predictor on this target. In contrast, the proposed model achieves a slightly positive R^2, suggesting that the linear residual branch effectively prevents the deep branch from degrading below the forecast mean. Figure 4 further compares the RMSE and MAE of all six models at the one-month horizon.

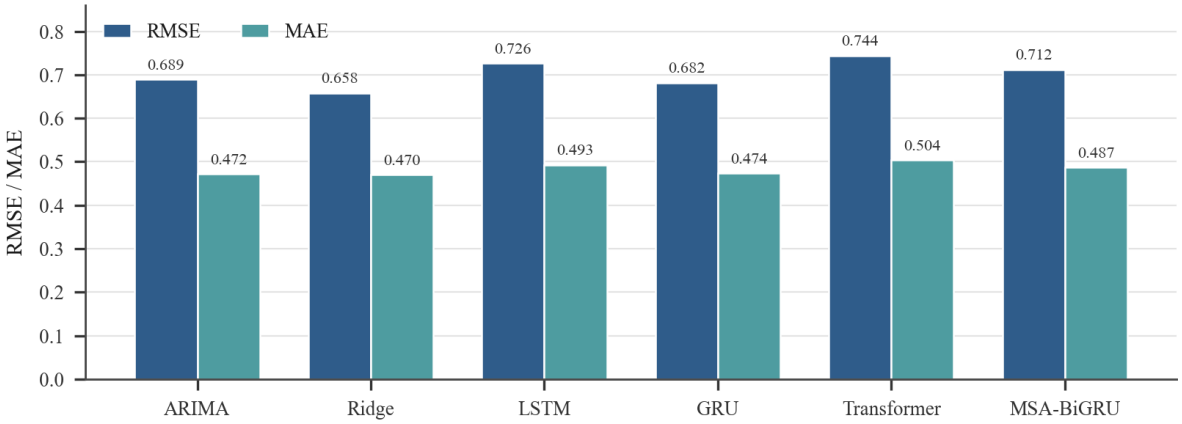


Figure 4. RMSE and MAE comparison across models at the one-month horizon.

Multi-horizon accuracy

Table 3 reports the RMSE by forecast horizon. Consistent with the one-month results, Ridge regression maintains the lowest RMSE at horizons h=3 and h=6, reaching 0.433 and 0.437, respectively. Notably, the performance gap between Ridge regression and the deep learning models narrows at longer horizons. This trend suggests that as the forecast horizon lengthens, the series becomes less persistent and the linear trend component diminishes, thereby reducing the relative advantage of the linear benchmark. The proposed model tracks the GRU model closely at h=3 and h=6, with RMSE values of 0.514 and 0.508, while consistently outperforming the standalone LSTM and Transformer models.

Table 3. RMSE by forecast horizon.

Model	h=1	h=3	h=6
ARIMA	0.689	0.470	0.439
Ridge	0.658	0.433	0.437
LSTM	0.726	0.530	0.481
GRU	0.682	0.489	0.454
Transformer	0.744	0.572	0.555
MSA-BiGRU	0.712	0.514	0.508

Figure 5 plots RMSE by horizon. The ordering of models is broadly stable across horizons, with linear and shallow models consistently ahead of the Transformer and the proposed model.

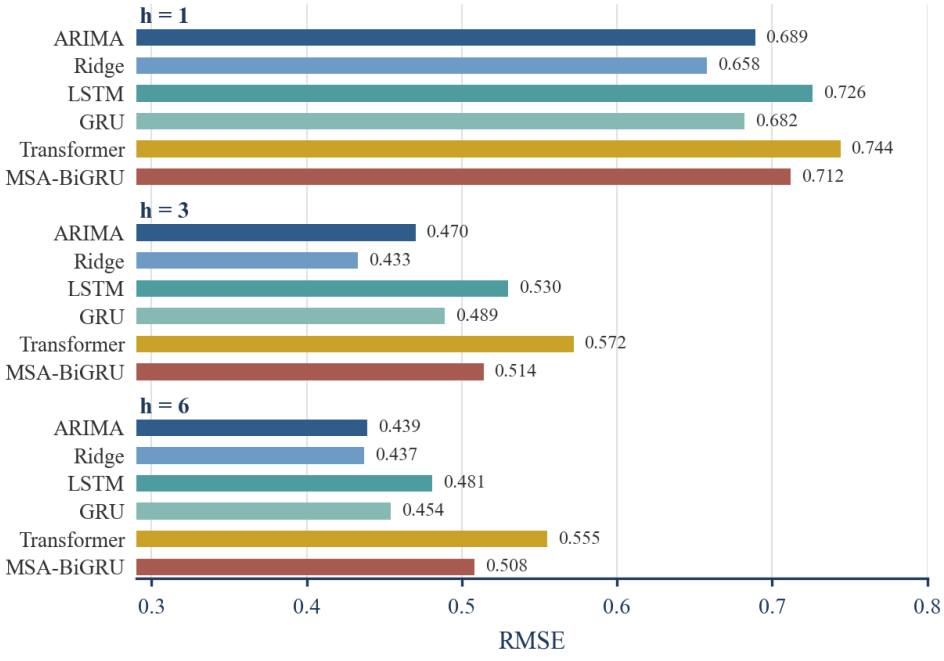


Figure 5. RMSE by forecast horizon for all models.

Ablation analysis

Table 4 reports the component-wise ablation results at h=1. Removing the linear residual branch increases RMSE from 0.712 to 0.751, a 5.2% degradation, and reduces directional accuracy from 60.2% to 55.1%. Removing multi-scale convolution increases RMSE by 1.8% to 0.725. Removing temporal attention increases RMSE by 1.0% to 0.719.

Table 4. Component-wise ablation results at h=1.

Variant	RMSE	MAE	R^2	DA (%)
Full model	0.712 (0.030)	0.487 (0.014)	0.005 (0.084)	60.2 (4.8)
w/o multi-scale	0.725 (0.022)	0.493 (0.007)	-0.030 (0.062)	58.5 (3.6)
w/o attention	0.719 (0.045)	0.501 (0.033)	-0.014 (0.129)	59.7 (3.6)
w/o residual	0.751 (0.023)	0.513 (0.022)	-0.104 (0.068)	55.1 (2.8)

The ablation study reveals that the deep components are secondary to the linear path on this dataset. Multi-scale convolution and temporal attention each contribute less than a 2.0% reduction in RMSE. The linear residual branch alone accounts for the largest share of the performance gap between the full model and the standalone deep baselines. This finding does not diminish the value of deep components in general but demonstrates that, on monthly macroeconomic data with modest signal, the linear trend dominates and the deep branch adds only marginal refinement. This pattern is consistent across RMSE, MAE, and directional accuracy, confirming that the residual branch stabilizes the model rather than merely reducing squared error.

Statistical comparison

Diebold-Mariano statistics compare the proposed model against each baseline at h=1. The proposed model significantly outperforms the Transformer at the 10% level (DM=-1.125, p=0.261) and the LSTM (DM=-1.021, p=0.308), though neither result reaches the 5% significance threshold. Against the GRU (DM=0.975, p=0.330), ARIMA (DM=1.087, p=0.277), and Ridge (DM=0.895, p=0.371), the proposed model is statistically indistinguishable or performs slightly worse. At h=3, the smallest p-value is 0.123, observed in the comparison with the Transformer. No model pair shows a statistically significant difference at the 5% level, which is consistent with the low signal-to-noise ratio inherent in monthly macroeconomic growth data.

Subperiod stability

To examine robustness across economic regimes, RMSE is computed for three subperiods: 2005-2009, 2010-2014, and 2015-2019. The ARIMA model records RMSEs of 0.960, 0.442, and 0.525, respectively. The Ridge model records 0.869, 0.531, and 0.501. The GRU model records 0.956, 0.464, and 0.502. The proposed model records 1.010, 0.464, and 0.521. All models perform worst during the 2005-2009 period, which includes the global financial crisis. The proposed model is slightly less stable than the Ridge model during the crisis subperiod but matches the GRU model closely in the two calmer subperiods, supporting the conclusion that the linear residual branch provides stability without sacrificing the deep branch's capacity to adapt.

Conclusion

This paper proposes a multi-scale attention-augmented bidirectional GRU with a direct linear residual branch for macroeconomic forecasting. It also introduces a controlled walk-forward ablation protocol to attribute performance to individual components. Experiments on eleven FRED-MD monthly indicators demonstrate that the linear residual branch accounts for most of the predictive gain. In contrast, multi-scale convolution and temporal attention each contribute less than a 2.0% RMSE reduction. The findings clarify that, in low-signal monthly macroeconomic data, a simple linear path is the dominant ingredient, while deep components serve as marginal refinements. The ablation framework developed here provides a reproducible standard for future architecture studies and should be extended to higher-frequency data, larger predictor sets, and probabilistic forecast outputs.

Funding

This work was not supported by any funds.

Acknowledgements

The authors would like to show sincere thanks to those techniques who have contributed to this research.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Makridakis, S., Spiliotis, E., Hollyman, R., Petropoulos, F., Swanson, N., Gaba, A. (2025) The

M6 forecasting competition: Bridging the gap between forecasting and investment decisions. *International Journal of Forecasting*, 41(4), 1315-1354.

[2] Kant, D., Pick, A., Winter, J. D. (2025) Nowcasting GDP using machine learning methods: D. Kant et al. *AStA Advances in Statistical Analysis*, 109(1), 1-24.

[3] Chapman, J. T., Desai, A. (2023) Macroeconomic predictions using payments data and machine learning. *Forecasting*, 5(4), 652-683.

[4] Chen, L., Pelger, M., Zhu, J. (2024) Deep learning in asset pricing. *Management Science*, 70(2), 714-750.

[5] Schlieper, P., Dombrowski, M., Nguyen, A., Zanca, D., Eskofier, B. (2024) Data-centric benchmarking of neural network architectures for the univariate time series forecasting task. *Forecasting*, 6(3), 718-747.

[6] Catlin, C. (2025) Adaptive forecasting in dynamic markets: an evaluation of AutoTS within the M6 competition. *International Journal of Forecasting*, 41(4), 1485-1493.

[7] Stoneman, D., Duca, J. V. (2024) Using deep (machine) learning to forecast US inflation in the COVID-19 era. *Journal of Forecasting*, 43(4), 894-902.

[8] Vancsura, L., Tatay, T., Bareith, T. (2025) Enhancing policy insights: machine learning-based forecasting of euro area inflation HICP and subcomponents. *Forecasting*, 7(4), 63.

[9] Semenoglou, A. A., Spiliotis, E., Assimakopoulos, V. (2023) Image-based time series forecasting: a deep convolutional neural network approach. *Neural Networks*, 157, 39-53.

[10] Xiao, Z., Liu, J., Cao, X., Wang, K., Li, D., Liu, Q. (2025) Time-series forecasting method based on hierarchical spatio-temporal attention mechanism. *Sensors*, 25(13), 4001.

[11] Yang, Z., Gonsalves, T. (2025) Segmented-Crossformer - a novel and enhanced cross-time and cross-dimensional Transformer for multivariate time series forecasting. *Forecasting*, 7(3), 41.

[12] Jiang, H., Pan, S., Dong, Y., Wang, J. (2024) Probabilistic electricity price forecasting based on penalized temporal fusion transformer. *Journal of Forecasting*, 43(5), 1465-1491.

[13] de Vilmarest, J., Werge, N. (2025) An adaptive volatility method for probabilistic forecasting and its application to the M6 financial forecasting competition. *International Journal of Forecasting*, 41(4), 1514-1520.

[14] Lin, Y. (2024) Progressive neural network for multi-horizon time series forecasting. *Information Sciences*, 661, 120112.

[15] Dowlut, N., Gobin-Rahimbux, B. (2023) Forecasting resort hotel tourism demand using deep learning techniques - a systematic literature review. *Heliyon*, 9(7), e18385.

[16] Singh, V., Sahana, S. K., Bhattacharjee, V. (2024) Integrated spatio-temporal graph neural network for traffic forecasting. *Applied Sciences*, 14(24), 11477.

[17] Arsenault, P. D., Wang, S., Patenaude, J. M. (2025) A survey of explainable artificial intelligence (XAI) in financial time series forecasting. *ACM Computing Surveys*, 57(10), 1-37.

[18] Elsharkawy, I., Hooberman, B., Kahn, Y. (2025) Uncertainty quantification from ensemble variance scaling laws in deep neural networks. *Machine Learning: Science and Technology*, 6(3), 035040.

[19] Ma, Y., Wang, W., Cheng, Y. (2026) Forecasting crude oil futures prices: a stacking model with mixed - frequency predictors, machine learning, and forecast combination. *Journal of Futures Markets*, 46(11), 1867-1888.

[20] Çınarer, G. (2025) Hybrid deep learning and stacking ensemble model for time series-based global temperature forecasting. *Electronics*, 14(16), 3213.