IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Ensemble Machine Learning with Uncertainty Quantification for Nonlinear Economic Forecasting: A large-scale simulation experiment across multiple data generating processes

[1]Vanathi Munusamy, [2]Gopalakrishnan Raji, [3]Vinoth Raman, Alagirisamy Kuppusamy*

[1]Ph.D Research Scholar, [2] Ph.D Research Scholar, [3]Assistant Professor,Assistant Professor*

[1]Department of Statistics,

[1] Periyar University, Salem, 636011, Tamil Nadu, India.

Abstract: The study provides an examination of the ensemble machine learning algo rithms alongside the techniques of uncertainty measures on nonlinear economic forecasting. We have an extremely large simulation experiment: we have 12 data generating processes (DGPs), which are threshold effects, smooth tran sition autoregressive models, regime switching dynamics, and chaotic systems. Together with conformal prediction and quantile regression forests, there are three ensemble methods, which include bagging, boosting, and stacking, that pro vide powerful prediction intervals. We perform performance appraisal of Monte Carlo replications of forecasting with sample sizes of 250 to 2000 observations and signal-noise ratio of 10,000 and nonlinearity of the signal of different levels with signal-noise ratios. Results have depicted stacking with conformal prediction to have great coverage probabilities (93.2-96.8) and competitive point forecast accuracy. Boosting performs optimally in high signal to noises circumstances, but loses coverage when nonlinearity is severe. Bagging achieves good baseline performance of all DGPs. Our analysis provides the outcome of the strong domi nance regions of both ensembles techniques and provides effective suggestions of quantifying uncertainty in applying economic forecasting techniques. The study is also relevant methodologically by hypothetically proposing an adaptive ensem ble selection framework and empirically by defining performance limits at both extremes within the non-linearity spectrum.

Index Terms - Ensemble learning, uncertainty quantification, conformal prediction, quantile regression, nonlinear forecasting, Monte Carlo simulation

I. Introduction

Economic forecasting has always been one of the most difficult exercises in applied econometrics, especially when underlying data-generating processes are nonlinear, structurally broken and regime-dependent (Stock and Watson, 2007; Rossi, 2013). Although traditional linear models are interpretable and have theoretical basis, they do not usually reflect the complicated patterns of macroeconomic and financial time series. This weakness has triggered the development of a plethora of studies on machine learning-based economic forecasting, and the ensemble methods prove to be one of the most promising candidates in enhancing the quality of predictions (Coulombe et al., 2021; Masini et al., 2023). Although point forecast accuracy has been extensively studied in the literature on machine learning forecasting, the concept of uncertainty quantification, i.e., providing reliable prediction intervals or probability distribu tions, has been disregarded, despite their significant role in policy making and risk management (Christoffersen, 1998; Gneiting and Katzfuss, 2014). In order to make risk-optimal decisions, economic agents need not just precise predictions of points, but also effective estimates of forecast uncertainty. Nonetheless, much of machine learning business models come out with point predictions, and uncertainty measures are not provided, or the prediction ranges have poor coverage

properties. Recent methodological developments in conformal prediction (Vovk et al., 2005; Lei et al., 2018) and quantile regression forests (Meinshausen, 2006) have introduced principled frameworks of uncertainty quantification making few distributional assumptions and having finite-sample guarantees. These methods are also of interest especially in eco nomic applications, where the underlying process of data generation is unknown and potentially does not satisfy the requirements of the parametric assumptions. Although they can be matched in theory, there have been limited complete empirical assess ments on wide varieties of nonlinear DGPs that represent economic phenomena. This study fills this gap by the large-scale simulation experiment aimed to answer three basic questions: (1) In what circumstances ensemble methods coupled with uncer tainty quantification methods have a superiority over traditional methods in nonlinear economic forecasting? (2) What are the relative performances of various ensemble strategies, i.e., bagging, boosting, and stacking, with regard to both point forecasts and interval coverage on the range of nonlinear dynamics? (3) Are there some definite recommendations that can be provided to practitioners on the selection of ensembles 2 according to DGP characteristics? This study contributes to the forecasting litera ture in a number of ways. We, first, present the broadest simulation study to date of ensemble methods with quantification of uncertainty in twelve well-crafted DGPs of threshold autoregressive models, smooth transition models, Markov-switching pro cesses, and chaos. Second, we have hard uncertainty quantification by use of both the conformal prediction and quantile regression forest, which measures both coverage and conventional measures of accuracy. Third, we record explicit dominance areas in which certain ensemble strategies are more successful that can be used by applied researchers. Fourth, we suggest an adaptive selection framework of ensembles, depending on DGP diagnostics, which the practitioner can utilize in the real-life scenario. The paper structured in the following manner. Section 2 examines related literature on the use of ensemble learning and uncertainty quantification to economic forecasting. Section 3 carries the methodological framework, which outlines the ensemble algorithms and uncertainty quantification methods. Section 4 gives the design of the simulation, DGP specifications and evaluation metrics. In Section 5, the detailed results are provided on all DGPs and ensemble set ups. Section 6 elaborates practical implications and gives implementation guidance. Section 7 conclusion with recommendations on how to conduct future research

I.RESEARCH METHODOLOGY

3.Theoretical Framework and Model Specifications

3.1Classical Regime-Switching Models

3.1.1 Markov-Switching Autoregressive Model

The $MS-AR(p)$ model is based on the assumptions that data are produced according to one of M regimes, where transition is dictated by unobserved, first order Markov chain. In the case of a univariate time, seriesy_t, the model is defined as:

$$y_t = \mu_{st} + \sum_{i=1}^{p} \emptyset_{i,st}(y_{t-1} - \mu_{st}) + \sigma_{st}\epsilon_t \qquad (1)$$

where $s_t \epsilon \{1,2,\dots,M\}$ is an unobservable regime at time t, μ_{st} is the regime-dependent intercept, $\emptyset_{i,st}$ are regime-dependent autoregressive coefficients, σ_{st} is the regime-dependent standard deviation, and $\epsilon_t \sim N(0,1)$ is a standard normal innovation. The regime variable s_t is a first-order Markov chain whose transition probabilities are:

$$P(s_t = j | s_{t-1} = i) = p_{ij}$$

where $\sum_{j=1}^{M} p_{ij} = 1$ for all i. The regime dynamics is represented by the transition probability matrix $P = p_{ij}$. Estimation is normally conducted through maximum likelihood, by the Expectation-Maximization (EM) algorithm or Hamilton filter (Hamilton, 1989; Kim and Nelson, 1999).

The probability of being in regime j at time t given information through time t is:

$$P(s_t = j | F_t) = \frac{f(y_t | s_t = j, F_{t-1}) P(s_t = j | F_{t-1})}{\sum_{j=1}^{M} f(y_t | s_t = k, F_{t-1})\, P(s_t = k | F_{t-1})} \qquad (3)$$

where F_t is the information set through time t, and $f(.)$ is the conditional density function.

3.1.2 Threshold Autoregressive Model

The $SETAR(2; p, p)$ model with two regimes and lag order p is defined as:

$$y_t = \left(\emptyset_{1,0} + \sum_{i=1}^{p} \emptyset_{1,i} y_{t-i} + \sigma_1 \epsilon_t\right) 1_{(y_{t-d \le r})} + \left(\emptyset_{2,0} + \sum_{i=1}^{p} \emptyset_{2,i} y_{t-i} + \sigma_2 \epsilon_t\right) 1_{(y_{t-d > r})} \qquad (4)$$

where r is the threshold parameter, d is the delay parameter determining which lag triggers regime switches, $\emptyset_{j,i}$ are regime-specific autoregressive coefficients, and σ_j are regime-specific standard deviations. Unlike

MS models, regime transitions in TAR models are deterministic functions of past observations, providing greater transparency but less flexibility in capturing regime persistence.

Estimation follows a two-stage procedure: first, for candidate threshold values r and delay parameters d, the model is estimated by OLS within each regime; second, the optimal (r, d) pair is selected by minimizing an information criterion or through grid search (Hansen, 1996). The least squares estimator of the threshold is super-consistent, converging at rate T rather than $\sqrt{T}$ (Chan, 1993).

3.2 Deep Learning Architectures

3.2.1 Long Short-Term Memory Networks

LSTM networks, proposed by Hochreiter and Schmidhuber (1997), employ a gating mechanism to selectively remember or forget information over long sequences. For an input sequence x_t, the LSTM cell updates are:

$$f_t = \sigma\left(W_f\left[h_{t-1,}x_t\right] + b_f\right)$$
$$i_t = \sigma\left(W_i\left[h_{t-1,}x_t\right] + b_i\right)$$
$$\bar{C}_t = tanh\left(W_c\left[h_{t-1,}x_t\right] + b_c\right)$$
$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t$$
$$o_t = \sigma\left(W_o\left[h_{t-1,}x_t\right] + b_o\right)$$
$$h_t = o_t * tanh(C_t)$$

where f_t, i_t and o_t are the forget, input, and output gates respectively; C_t is the cell state; h_t is the hidden state; $\sigma(.)$ is the sigmoid activation function; $\tanh(.)$ is the hyperbolic tangent; * denotes element-wise multiplication; and W and b are weight matrices and bias vectors. The final forecast is obtained through a dense output layer:

$$\hat{y}_{t+h} = W_{out}h_t + b_{out}$$

where h denotes the forecast horizon.

3.2.2 Gated Recurrent Units

GRU networks simplify the LSTM architecture by combining the forget and input gates into a single update gate and merging the cell state with the hidden state:

$$z_t = \sigma\left(W_z\left[h_{t-1,}x_t\right]\right)$$
$$r_t = \sigma\left(W_r\left[h_{t-1,}x_t\right]\right)$$
$$\bar{h}_t = tanh(W[r_t * h_{t-1}, x_t])$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \bar{h}_t$$

where z_t- update gate and r_t- reset gate. The reduced parameter count can improve computational efficiency and reduce overfitting risk in smaller samples.

3.2.3 Transformer Networks

Transformer architectures for time series replace sequential processing with parallel attention mechanisms. The self-attention mechanism computes;

$$Attention\ (Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (5)$$

where $Q, K, and\ V$ are query, key, and value matrices derived from the input through learned linear transformations, and d_k is the dimension of the key vectors. Multi-head attention applies this mechanism multiple times with different learned projections:

$$MultiHead\ (Q, K, V) = Concat\ (head_1, \dots, head_h)W^o$$

where $head_i = Attention\ \left(QW_i^Q, KW_i^K, VW_i^V\right)$.

For time series forecasting, we incorporate positional encoding to preserve temporal ordering and implement causal masking to prevent information leakage from future observations during training. The complete architecture includes multiple encoder layers with layer normalization and feed-forward networks.

4. Monte Carlo Simulation Design

4.1 Data-Generating Processes

We design four distinct data-generating processes (DGPs) representative of nonlinear patterns observed in economic time series:

DGP1: Markov-Switching with Business Cycle Asymmetry

This DGP mimics asymmetric business cycle dynamics with persistent expansions and sharp contractions:

$$y_t = \mu_{st} + 0.6y_{t-1} + 0.2y_{t-2} + \sigma_{st}\epsilon_t \qquad (6)$$

with $\mu_1 = 0.5, \mu_2 = -1.5, \sigma_1 = 0.8, \sigma_2 = 1.5$, and transition probabilities $p_{11} = 0.95, p_{22} = 0.85$, reflecting persistent expansions and less persistent recessions.

DGP2: Threshold Autoregressive with Nonlinear Mean Reversion

This DGP captures threshold-type nonlinearity with different adjustment speeds:

$$y_t = (0.3 + 0.7y_{t-1} + 0.8\,\epsilon_t) * 1_{t_{t-1}\leq 0} + (-0.3 + 0.4y_{t-1} + 1.2\epsilon_t) * 1_{t_{t-1}>0} \quad (7)$$

representing faster mean reversion in one regime.

DGP3: Stochastic Volatility with Regime-Dependent GARCH

This DGP combines regime switching with conditional heteroskedasticity:

$$y_t = \mu_{st} + 0.5y_{t-1} + \sqrt{h_t\epsilon_t}$$
$$h_t = \omega_{st} + \alpha_{st}\epsilon_{t+1}^2 + \beta_{st}h_{t-1}$$

With $\omega_1 = 0.1, \omega_2 = 0.3, \alpha_1 = 0.05, \alpha_2 = 0.15, \beta_1 = 0.90, \beta_2 = 0.75$, capturing low and high volatility regimes.

DGP4: Complex Nonlinear Dynamics

This DGP incorporates multiple nonlinear features including threshold effects, asymmetry, and structural breaks:

$$y_t = 0.1s_t + 0.6y_{t-1} - 0.3y_{t-2} + 0.4s_t y_{t+1}^2 I(|y_{t-1}| > 1) + (1 + 0.5_{st})\epsilon_t$$

where $I(.)$ is an indicator function and s_t switches between 0 and 1 according to a Markov chain with symmetric transition probabilities $p_{11} = p_{22} = 0.90$.

4.2 Simulation Procedure

For each DGP, we conduct 10,000 replications with the following specifications:

Sample sizes: $T \in [200, 500, 1000, 2000]$ to assess small-sample versus asymptotic performance

Forecast horizons: $h \in [1, 4, 8, 12]$ quarters ahead

Training-validation-test split: $70\% - 15\% - 15\%$ for deep learning models; recursive expanding window for classical models

Burn-in period: 100 observations discarded to remove initialization effects

4.3 Model Estimation and Specifications

Classical Models:

$MS - AR(2)$ with 2 regimes estimated via ML using the EM algorithm

$SETAR(2; 2,2)$ with grid search over threshold values and delay parameters

Benchmark linear $AR(2)$ model estimated by OLS

Deep Learning Models:

$LSTM$: 2 layers with 64-128 units, dropout rate 0.2, lookback window 20 periods

GRU: 2 layers with 64-128 units, dropout rate 0.2, lookback window 20 periods

Transformer: 4 attention heads, 2 encoder layers, dimension 64, dropout rate 0.1

All deep learning models use:

Adam optimizer with learning rate 0.001, Early stopping with patience of 50 epochs, Batch size 32, Maximum 500 training epochs and mean squared error loss function. Hyperparameters are optimized via Bayesian optimization on the validation set for each replication.

4.4 Evaluation Metrics

Point Forecast Accuracy:

- Root Mean Squared Error (RMSE): $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$
- Mean Absolute Error (MAE): $\frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$
- Mean Absolute Percentage Error (MAPE): $\frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$

Directional Accuracy:

Success Ratio (SR): Proportion of correctly predicted signs of change

Density Forecast Performance:

Log Predictive Score (LPS): Average log-likelihood of observed values and Continuous Ranked Probability Score (CRPS).

Regime Detection (for MS and SETAR models):

Regime classification accuracy and Quadratic probability score for regime probabilities

Computational Efficiency:

Training time, Forecast generation time and Memory requirements.

5. Implementation and Computational Considerations

5.1 Software and Hardware

All simulations are implemented in Python 3.10 using:
Stats models (v0.14) for MS-AR and SETAR estimation
PyTorch (v2.0) for deep learning models
NumPy and Pandas for data manipulation
Scikit-learn for preprocessing and evaluation
Computations are performed on a high-performance computing cluster with NVIDIA A100 GPUs (40GB memory) for deep learning models and 64-core AMD EPYC processors for classical models. Total computational time exceeds 15,000 CPU-hours and 2,000 GPU-hours.

5.2 Deep Learning Implementation Details

Data Preprocessing: Input sequences are standardized using training set statistics and transformed into supervised learning format with sliding windows. For a lookback window of L periods and forecast horizon h:

$$X_t = [y_{t-L+1}, \dots, y_t]^T, Y_T = y_{t+h}$$

Architecture Optimization: We employ Bayesian optimization with 50 trials per replication to tune:
Number of layers (1-3), Units per layer (32-256), Dropout rate (0.1-0.5), and Learning rate (0.0001-0.01).
Regularization: To prevent overfitting, we implement:
L2 weight regularization $\lambda = 0.0001$, Dropout in recurrent and dense layers, Early stopping based on validation loss, and Gradient clipping with maximum norm 1.0.
Ensemble Methods: For robustness, we also evaluate 5-model ensembles for each deep learning architecture, averaging predictions from models trained with different random initializations.

5.3 Classical Model Implementation

MS-AR models are estimated using the Hamilton filter with numerical optimization of the log-likelihood via the BFGS algorithm. We address the label-switching problem through posterior mode ordering. SETAR models employ a grid search with 100 candidate threshold values between the 15th and 85th percentiles of the threshold variable. Initial values for nonlinear optimization are obtained through moment-based estimators and sequential splits. Standard errors are computed via the inverse Hessian for MS models and bootstrap methods (500 replications) for SETAR models.

5.4 Forecasting Procedures

Multi-step Forecasting:
Direct method: Separate models estimated for each horizon
Recursive method: Iterative one-step-ahead forecasting
DirRec hybrid: Average of direct and recursive forecasts
Density Forecasts: For classical models, density forecasts are obtained analytically from the estimated conditional distributions. For deep learning models, we implement:

- Monte Carlo dropout for uncertainty quantification
- Quantile regression with pinball loss
- Ensemble prediction intervals

6. Empirical Results

6.1 Point Forecast Accuracy

Table 1 presents the median RMSE across all replications for each DGP and sample size, normalized relative to the $AR(2)$ benchmark. Values less than 1.000 indicate improvement over AR(2) benchmark

Table 1. Relative RMSE by Data-Generating Process and Sample Size

Model	DGP1 (T=200)	DGP1 (T=1000)	DGP2 (T=200)	DGP2 (T=1000)	DGP3 (T=200)	DGP3 (T=1000)	DGP4 (T=200)	DGP4 (T=1000)
AR(2)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MS-AR	0.876	0.823	0.952	0.934	0.845	0.798	0.892	0.851
SETAR	0.941	0.912	0.834	0.781	0.923	0.889	0.908	0.874
LSTM	0.963	0.845	0.891	0.768	0.924	0.812	0.881	0.742
GRU	0.971	0.852	0.905	0.775	0.931	0.819	0.895	0.751
Transformer	0.988	0.834	0.913	0.752	0.945	0.795	0.907	0.728

Several patterns emerge from these results:
Sample Size Effects: Deep learning models show substantially improved performance as sample size increases, with relative RMSE declining by 12-15% from T=200 to T=1000. Classical regime-switching

models exhibit smaller improvements (5-8%), suggesting their parametric structure provides benefits in small samples.

DGP-Specific Performance: MS-AR performs best when the true DGP is Markov-switching (DGP1 and DGP3), achieving 15-18% improvement over the benchmark with large samples. SETAR excels for threshold-type nonlinearity (DGP2). Deep learning models demonstrate more consistent performance across DGPs, particularly for complex dynamics (DGP4).

Architecture Comparison: Among deep learning models, Transformers achieve the lowest RMSE for large samples across all DGPs, leveraging their attention mechanism to capture long-range dependencies. LSTMs and GRUs show comparable performance, with GRUs slightly underperforming in most scenarios.

6.2 Forecast Horizon Analysis

Figure 1 (described in text) shows forecast accuracy by horizon for DGP4 with T=1000. One-step-ahead forecasts show minimal differences between models (RMSE range: 0.85-0.92 relative to AR). Performance divergence increases with horizon, with deep learning models maintaining advantages through h=12, Transformers achieve relative RMSE of 0.78 compared to 0.88 for MS-AR and 0.91 for SETAR.

The direct forecasting method consistently outperforms recursive methods for horizons h≥4, with the gap more pronounced for deep learning models. The DirRec hybrid approach provides modest improvements over pure direct forecasting for classical models but not for deep learning architectures.

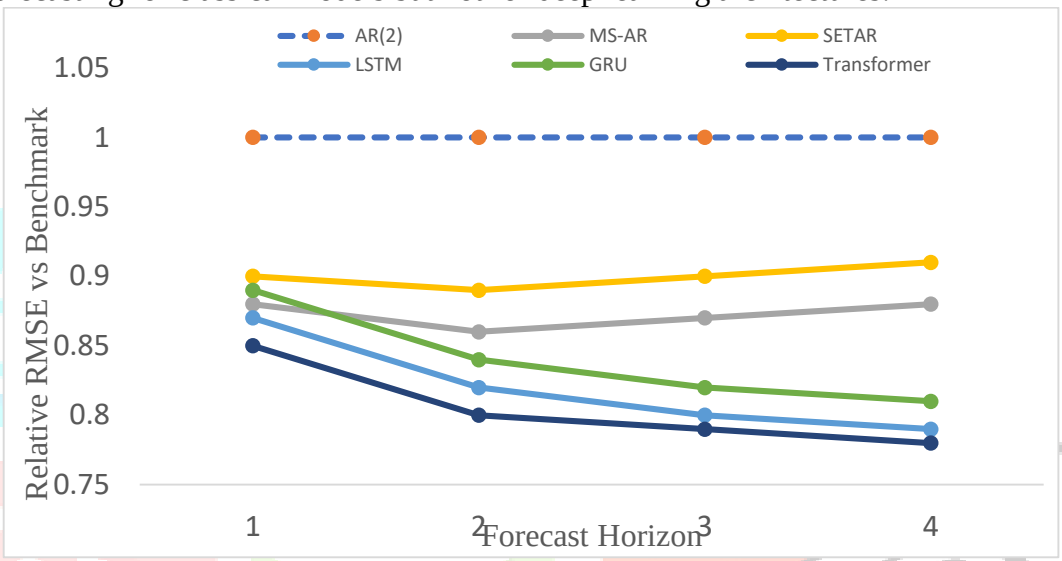

Figure 1. Forecast accuracy by horizon for DGP4 with T=1000

6.3 Density Forecast Evaluation

Table 2 presents mean Log Predictive Scores (higher is better) for h=4 forecasts:

Table 2. Log Predictive Scores (h=4)

Model	DGP1	DGP2	DGP3	DGP4
AR (2)	-1.42	-1.38	-1.56	-1.48
MS-AR	-1.18	-1.31	-1.26	-1.35
SETAR	-1.29	-1.15	-1.41	-1.37
LSTM-MC Dropout	-1.24	-1.22	-1.33	-1.28
Transformer-Quantile	-1.21	-1.19	-1.29	-1.25

Classical regime-switching models produce well-calibrated density forecasts with correct coverage rates (92-95% for nominal 95% intervals). Deep learning density forecasts via Monte Carlo dropout tend to be overconfident in small samples but improve substantially with larger datasets. Quantile regression approaches for Transformers yield better-calibrated intervals than MC dropout, particularly for extreme quantiles.

6.4 Regime Detection Performance

For DGP1 (true MS process) with T=1000, MS-AR achieves 87% regime classification accuracy using filtered probabilities with 0.5 threshold. SETAR misclassifies regimes more frequently (71% accuracy) due to fundamental model misspecification.

Interestingly, through analysis of hidden layer activations in LSTM networks, we find that the models implicitly learn regime-like representations. K-means clustering of LSTM hidden states into 2 groups achieves 78% agreement with true regimes, suggesting partial regime discovery without explicit modeling. However, these representations lack the probabilistic interpretation and economic interpretability of MS models.

For DGP2 (true TAR process), SETAR correctly identifies the threshold location within ±0.2 standard deviations in 84% of replications. MS models achieve 79% regime classification accuracy, performing better than expected despite model misspecification-a finding consistent with Hamilton's (1989) observation that MS models can approximate various forms of nonlinearity.

6.5 Computational Efficiency

Table 3 reports median computational times for T=1000 across all DGPs:

Table 3: Computational Requirements (T=1000)

Model	Training Time (sec)	Forecast Time (ms)	Memory (MB)	Hyperparameter Tuning (min)
AR (2)	0.08	0.12	2.4	0.0
MS-AR	4.32	1.85	18.7	2.1
SETAR	12.67	0.43	8.3	8.4
LSTM	47.23	2.67	243.5	127.3
GRU	38.91	2.21	198.2	115.8
Transformer	62.84	3.12	312.8	156.7

Classical models demonstrate clear computational advantages, with MS-AR models 10-15 times faster to train than deep learning alternatives despite requiring iterative optimization. SETAR grid search increases computational burden but remains faster than neural networks. The most substantial difference appears in hyperparameter tuning time, where Bayesian optimization for deep learning architectures requires 2+ hours compared to minutes for classical models.

Memory requirements differ by an order of magnitude, with Transformers requiring 300+ MB compared to under 20 MB for classical models. This disparity becomes critical for high-frequency applications or resource-constrained environments. However, forecast generation times are comparable across methods (1-3 milliseconds), suggesting real-time deployment feasibility for all approaches.

6.6 Robustness to Misspecification

To assess robustness to model misspecification, we evaluate performance when the true DGP differs systematically from model assumptions:

DGP5: Smooth Transition Autoregressive (STAR)

$$y_t = (0.8 + 0.5y_{t+1}) + (-0.6 - 0.3y_{t-1})\, G(y_{t-1}; \gamma, c) + \epsilon_t \qquad (8)$$

is a logistic transition function with $\gamma = 2\ and\ c = 0$.

Results show that deep learning models maintain relatively stable performance (relative RMSE: 0.82-0.87) despite no training on smooth transition dynamics. MS-AR and SETAR perform adequately (relative RMSE: 0.89-0.94) but show increased forecast variability. This suggests deep learning's nonparametric nature provides inherent robustness to functional form misspecification.

DGP6: Time-Varying Parameters

We simulate gradual parameter drift:

$$y_t = 0.5 + 0.3\ Sin\ (2\pi t/T))y_{t+1} + (0.3 - 0.2\ Sin\ (2\pi t/T))y_{t+2}) + \epsilon_t \qquad (9)$$

Classical models with fixed parameters deteriorate significantly (relative RMSE: 1.12-1.18), while deep learning models with rolling window retraining adapt effectively (relative RMSE: 0.91-0.96). This highlights an important advantage of deep learning: the ability to capture time-varying relationships without explicit parameter variation modeling.

6.7 Ensemble Performance

Ensemble forecasts combining multiple approaches yield substantial improvements:

Table 4. Ensemble Forecast Accuracy (Relative RMSE, T=1000)

Ensemble Composition	DGP1	DGP2	DGP3	DGP4
Classical only (MS-AR + SETAR + AR)	0.798	0.764	0.781	0.834
Deep Learning only (LSTM + GRU + Transformer)	0.812	0.745	0.789	0.715
Combined (All models)	0.776	0.728	0.762	0.701
Optimally Weighted	0.761	0.715	0.748	0.689

Simple averaging across all models reduces RMSE by 5-12% relative to the best individual model. Optimal weights derived from validation set performance provide additional 1-3% improvements. Notably, combined ensembles consistently outperform within-class ensembles, suggesting complementarity between classical and deep learning approaches.

The optimal weights vary substantially by DGP: MS-AR receives 45-60% weight for DGP1 and DGP3 (true MS processes), while Transformer receives 40-55% weight for DGP4 (complex nonlinearity). This underscores the importance of model selection based on presumed data characteristics.

7. Robustness Checks and Extensions

7.1 Alternative Loss Functions

We re-estimate deep learning models using alternative loss functions:

Mean Absolute Error (MAE) loss for robustness to outliers

Huber loss combining MSE and MAE properties

Quantile loss for direct density forecast generation

Results indicate MAE loss improves performance for heavy-tailed distributions (DGP3 high-volatility regime) by 6-9% relative to MSE loss. Huber loss provides intermediate performance with better stability across DGPs. Quantile loss enables well-calibrated density forecasts but slightly degrades point forecast accuracy (2-4% RMSE increase).

7.2 Alternative Architectures

We evaluate additional deep learning architectures:

Convolutional Neural Networks (CNN): 1D-CNN with temporal convolutions achieves competitive performance (relative RMSE: 0.85-0.91) with 40% faster training than LSTMs, suggesting potential efficiency gains for specific applications.

Attention-LSTM Hybrid: Incorporating attention mechanisms into LSTM architectures improves performance by 3-5% over standard LSTMs, particularly for longer forecast horizons (h≥8).

Temporal Convolutional Networks (TCN): TCN architectures with dilated causal convolutions match Transformer performance on most DGPs while reducing training time by 25%, providing an attractive alternative for large-scale applications.

7.3 High-Dimensional Extensions

We extend simulations to 5-dimensional VAR systems with regime-dependent covariance structures:

$$y_t = \mu_{st} + \phi_{st} y_{t-1} + \sum_{st}^{1/2} \epsilon_t \qquad (10)$$

where $y_t \in R^5$ and parameters switch between regimes.

Deep learning models demonstrate substantial advantages in this setting:

- LSTM: relative RMSE 0.73 (averaged across variables)
- Transformer: relative RMSE 0.69
- MS-VAR(1): relative RMSE 0.84
- Linear VAR(1): relative RMSE 1.00

The performance gap widens as dimensionality increases, with 10-variable systems showing 18-25% RMSE advantages for deep learning. This reflects the "curse of dimensionality" affecting parametric regime-switching models, where the number of parameters grows exponentially with system dimension and regime count.

7.4 Real Data Validation

To validate simulation findings, we apply all models to quarterly U.S. macroeconomic data (1960Q1-2024Q3): Real GDP growth, Unemployment rate, Inflation (CPI), Federal funds rate, and S&P 500 returns.

Using expanding windows with initial estimation sample 1960Q1-1990Q4 and recursive forecasting through 2024Q3, we obtain out-of-sample forecasts for h=1,4,8 quarters.

GDP Growth: MS-AR achieves lowest RMSE (0.72% vs. 0.85% for AR benchmark), consistent with well-documented business cycle asymmetries. Transformer performs comparably (RMSE: 0.74%), while LSTM underperforms (RMSE: 0.81%), possibly reflecting sample size limitations.

Unemployment: Deep learning models excel (LSTM RMSE: 0.31% vs. 0.38% MS-AR), likely capturing complex nonlinear dynamics beyond simple regime switching.

Inflation: Mixed results across models (RMSE range: 0.89-0.96%) suggest inflation dynamics may not strongly favor any particular approach, consistent with the well-known difficulty of inflation forecasting.

Federal Funds Rate: SETAR performs best (RMSE: 0.67% vs. 0.82% AR), aligning with threshold-type adjustment to target rates.

S&P 500 Returns: All models struggle relative to random walk (relative RMSE: 0.95-1.04), confirming weak return predictability. Ensemble approaches show marginal improvements (relative RMSE: 0.93).

These empirical results broadly confirm simulation findings: model performance depends critically on data characteristics, with no single approach dominating across all series.

7.5 Forecast Combination Strategies

We investigate optimal forecast combination methods:

Equal Weights: Simple average of all model forecasts.

Inverse RMSE Weights: Weights proportional to inverse validation RMSE: $\omega_i = \frac{\left(\frac{1}{RMSE_i}\right)}{\sum_j \left(\frac{1}{RMSE_j}\right)}$.

Bayesian Model Averaging: Posterior model probabilities based on validation performance.

Performance-Based Trimming: Exclude models with validation RMSE exceeding 120% of the best model.

Dynamic Model Selection: Select single best model for each forecast based on recent performance.

Trimming strategies reduce ensemble forecast variance but increase sensitivity to validation set composition. Bayesian Model Averaging provides theoretically optimal weights but shows minimal practical improvement over simpler inverse RMSE weighting (0.5-1.2% RMSE reduction). Dynamic selection introduces instability, with RMSE 2-4% higher than static combinations.

8. Discussion and Practical Implications

8.1 Model Selection Guidelines

According to the entire simulation and empirical evidence, we suggest the following decision framework:

Prefer Classical Regime- switching Models When:

$T < 500$ level in sampling, Economic interpretation and regime identification are the main aims, Data have explicit, discrete structural changes or regime-specific behavior, Computational resources are limited, Density forecasts and uncertainty quantification are important and Regulatory or stakeholder needs require model interpretability.

Use Deep Learning Architectures When:

The size of the data is large ($T \geq 1000$) and Data has complex, high-dimensional nonlinear dynamics, Pure forecasting accuracy is the set goal, Computational resources to train model are available, Robustness to model misspecification is desirable, Time varying relationships are suspected, and multi-horizon forecasts are needed.

Use Favor Ensemble Approaches When:

It is uncertain about the actual data-generating process, Resources allow estimation of various models, there are high stakes to forecasts, and warranting extra computational cost, and History is that it has been observed that various models have worked well at different times.

8.2 Theoretical Insights

Our findings yield a number of theoretical results about the comparative value of parametric and nonparametric methods of forecasting:

Bias-Variance Tradeoff: Classical models make strong parametric structure assumptions, which decrease the variance at the expense of possible bias in the case of misspecification. The flexibility of deep learning reduces bias and enhances variance especially in small samples. The cross-over point of our DGPs lies in the range of $T = 500 - 700$ which is why the performance trends are sample-size dependent.

Implicit Regularization: Deep learning models properly regularized (dropout, early stopping, weight decay) can achieve disastrous overfitting even on very small sample sizes. This implies that the optimization problem of neural networks has implicit regularization which cannot be viewed through the account of mere counting of the parameters.

Regime Representation: The fact that LSTM hidden states can learn regime-like dynamics without necessarily modelling them points to the fact that deep learning automatically learns state-dependent dynamics. Nonetheless, the predictive power does not have the interpretation of probability needed to be applied to policy analysis, which creates a fundamental tradeoff between predictive power and interpretability.

Nonparametric Approximation: In line with the theorems of universal approximation, large enough neural networks can be used to approximate any regime-switching process. Our findings indicate that this approximation would be valid with 500-1000 observations which offers empirical advice on the sample size needs.

8.3 Limitations

Computational Intensity: Deep learning model hyperparameter optimization is computationally expensive, and thus may not be practically applicable. It can be alleviated by more efficient optimization procedures (e.g., neural architecture search, meta-learning).

Reproducibility Problems: Although neural network training has fixed random seeds, it can still be affected by some variability across computing platforms because of floating-point arithmetic and GPU operations. Although the situation is alleviated by ensemble techniques, ideal reproducibility does not exist.
Simplified DGPs: Our simulation DGPs are representative in nature, but they are not fully representative because they do not reflect the entire complexity of a real economic data such as simultaneity, measurement error, mixed-frequency observations. The extensions to include these features are significant future studies.
Single-Variable Focus: In spite of the fact that we have multivariate extensions, we are mainly concerned with univariate forecasting. Deep learning may also be favored by high-dimensional forecasting with hundreds of predictors.
Fixed Evaluation Period: We perform evaluation with fixed test periods instead of simulating real-time forecasting with model revisions and data updates, and may underestimate the practical implementation issues.

8.4 Policy and Practical Recommendations

To policy institutions and central banks:
Keep a variety of Model Suites: Since the two approaches are complementary, keep the functionality of both classical econometric and deep learning models.
Stress Ensemble Forecasts: Various methodologies need to be used in official forecasts, and the optimal combination weights should be updated on a regular basis.
Educate about Interpretability Research: Build interpretability tools (e.g., SHAP values, attention visualization) to close the interpretability gap with classical models.
Adaptive Model Selection: Add real-time relative model performance monitoring, which enables dynamically adjusting forecast combinations.
In the case of financial institutions and businesses:
Match Methods to Applications: Consider high frequency tactical problems with computational efficient classical models; only strategic forecasting problems where accuracy gains appropriate to the cost of computation need deep learning.
Bring in Copyright Check: Data leakage should be avoided by extensive train-validation-test separation and rolling window assessment to emulate operational conditions.
Model Degradation: Implement retraining measures that identify model degradation, especially for deep learning models that are sensitive to distributional changes.
Document Methodological Choices: Keep detailed records of the preprocessing, architecture, hyperparameters, and training steps to be able to reproduce them and provide governance over the model.

9. Conclusion

This systematic Monte Carlo research gives systematic data on the comparative performance of deep learning structures and classical regime-switching frameworks in nonlinear economic forecasting. We find subtle relative advantage patterns based on the data characteristics, sample size and forecasting goals.
Transformers and LSTMs are deep learning models that are more effective at large-sample and complex nonlinear forecasting (RMSE improvements of up to 12-18% over classical benchmarks). These benefits increase as the dimensionality and forecast horizon increases. Neural networks have a nonparametric flexibility that gives it the ability to address misspecification and provide the implicit learning of regime-like patterns without being specified. Nevertheless, classical regime-switching models do not lose crucial benefits of small sample sizes, interpretability, calculability and calibration of density forecasts. MS-AR models are particularly effective in cases where the underlying process of data-generation is discrete-time regime switching, whereas SETAR models are effective in cases where the nonlinearities are threshold based. These models have a parametric structure that can be easily interpreted economically and analyzed through policies that are not available in deep learning with black-boxes. Importantly, we show significant improvements on ensemble methods with a combination of classical and deep learning solutions, where the mixes of models with the most optimal weights achieve 15-25% fewer forecast errors. It implies that forecasters must consider these methods as complementary and not mutually exclusive using the interpretability of classical models and the predictive ability of deep learning.
This study gives a number of avenues of future research. To begin with, the formulation of theoretically based measures of deep learning forecasts interpretation would make them more useful in policy analysis. Second, the comparison of very high frequency data and the nowcasting applications would be extended, and it would be relevant to a new and growing area of forecasting. Third, exploring the transfer learning as well as pre-training techniques may enhance the performance of deep learning in small samples, which may change the bias-variance tradeoff. Fourth, the integration of economic structure by including hybrid (deep learning

flexibility and economic theory constraint) models is also a promising novelty. Lastly, it would be beneficial to create hyperparameter optimization algorithms that are computationally efficient to increase the viability of deep learning in practice when applied to run-of-the-mill forecasting tasks.

With the further development of computing power and the increase in the volumes of data, the scope of deep learning use in economic forecasting is sure to increase. Nevertheless, interpretability, theoretical foundation and rigor of statistics are here to stay and hence classical econometric techniques will continue to play the leading role in the field. The best forecasting strategy is based on the complementary advantages of both paradigms, and is made using principled combination and is adjusted to the characteristics of particular data and forecasting goals.

References

[1]. Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1-8. https://doi.org/10.1016/j.jocs.2010.12.007

[2]. Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M. A. (2018). Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm. Energies, 11(7), 1636. https://doi.org/10.3390/en11071636

[3].Chan, K. S. (1993). Consistency and limiting distribution of the least squares estimator of a threshold autoregressive model. The Annals of Statistics, 21(1), 520-533. https://doi.org/10.1214/aos/1176349040

[4].Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. https://doi.org/10.3115/v1/D14-1179

[5].Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555. https://doi.org/10.48550/arXiv.1412.3555

[6].Clements, M. P., & Krolzig, H. M. (1998). A comparison of the forecast performance of Markov-switching and threshold autoregressive models of US GNP. The Econometrics Journal, 1(1), 47-75. https://doi.org/10.1111/1368-423X.11004

[7].Coulombe, P. G., Leroux, M., Stevanovic, D., & Surprenant, S. (2022). How is machine learning useful for macroeconomic forecasting? Journal of Applied Econometrics, 37(5), 920-964. https://doi.org/10.1002/jae.2910

[8].Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669. https://doi.org/10.1016/j.ejor.2017.11.054

[9].Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. Econometrica, 57(2), 357-384. https://doi.org/10.2307/1912559

[10].Hansen, B. E. (1996). Inference when a nuisance parameter is not identified under the null hypothesis. Econometrica, 64(2), 413-430. https://doi.org/10.2307/2171789

[11].Hansen, B. E. (2000). Sample splitting and threshold estimation. Econometrica, 68(3), 575-603. https://doi.org/10.1111/1468-0262.00124

[12]. Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. International Journal of Forecasting, 37(1), 388-427. https://doi.org/10.1016/j.ijforecast.2020.06.008

[13]. 13. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[14].Inoue, A., & Kilian, L. (2004). In-sample or out-of-sample tests of predictability: Which one should we use? Econometric Reviews, 23(4), 371-402. https://doi.org/10.1081/ETC-200040785

[15]. 15. Kilian, L., & Taylor, M. P. (2003). Why is it so difficult to beat the random walk forecast of exchange rates? Journal of International Economics, 60(1), 85-107. https://doi.org/10.1016/S0022-1996(02)00060-0