Macroeconomic Forecasting with Large Language Models

Andrea Carriero[a] Davide Pettenuzzo[b] Shubhranshu Shekhar[b]

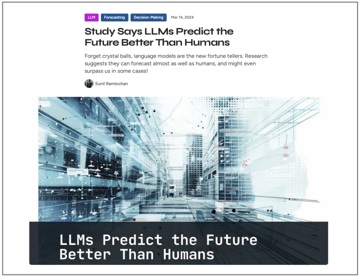
[a]Queen Mary University of London
[b]Brandeis University

NBER Summer Institute

Forecasting and Empirical Methods

July 10 2025

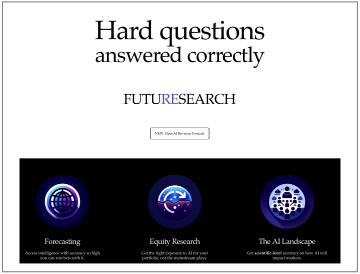
Can LLMs predict the future?

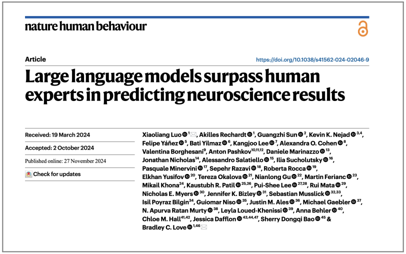

LLM Forecasting Decision-Making Mar 14, 2024

Study Says LLMs Predict the Future Better Than Humans

Forget crystal balls, language models are the new fortune tellers. Research suggests they can forecast almost as well as humans, and might even surpass us in some cases!

Sunil Ramlochan




nature human behaviour

Article https://doi.org/10.1038/s41562-024-02046-9

Large language models surpass human experts in predicting neuroscience results

Received: 19 March 2024

Accepted: 2 October 2024

Published online: 27 November 2024

Check for updates

Xiaoliang Luo[1], Akilles Rechardt[1], Guangzhi Sun[2], Kevin K. Nejad[3,4], Felipe Yáñez[5], Bati Yilmaz[6], Kangjoo Lee[7], Alexandra O. Cohen[8], Valentina Borghesani[9], Anton Pashkov[10,11,12], Daniele Marinazzo[13], Jonathan Nicholas[14], Alessandro Salatiello[15], Ilia Sucholutsky[16], Pasquale Minervini[17], Sepehr Razavi[18], Roberta Rocca[19], Elkhan Yusifov[20], Tereza Okalova[21], Nianlong Gu[22], Martin Ferianc[23], Mikail Khona[24], Kaustubh R. Patil[25,26], Pui-Shee Lee[27,28], Rui Mata[29], Nicholas E. Myers[30], Jennifer K. Bizley[31], Sebastian Musslick[32,33], Isil Poyraz Bilgin[34], Guiomar Niso[35], Justin M. Ales[36], Michael Gaebler[37], N. Apurva Ratan Murty[38], Leyla Loued-Khenissi[39], Anna Behler[40], Chloe M. Hall[41,42], Jessica Dafflon[43,44,47], Sherry Dongqi Bao[45] & Bradley C. Love[1,46]

Introduction

- Large Language Models (LLMs) have reshaped natural language processing
  - Have demonstrated proficiency in capturing linguistic nuances and semantic meanings
  - Used routinely for content generation, information extraction, code generation and completion, Conversational AI and chatbots
  - Increasingly shown to be useful for forecasting in various scientific and technical disciplines
- This paper focuses on a more recent development
  - Time Series Foundational Models (TSFMs) or Time Series Language Models (TSLMs)
  - Large-scale, general-purpose neural networks pre-trained on large amounts of diverse data across various frequencies and domains
  - TSLMs build on LLM architecture to uncover intricate nonlinear relationships in time series data

Introduction

- Large Language Models (LLMs) have reshaped natural language processing
  - Have demonstrated proficiency in capturing linguistic nuances and semantic meanings
  - Used routinely for content generation, information extraction, code generation and completion, Conversational AI and chatbots
  - Increasingly shown to be useful for forecasting in various scientific and technical disciplines
- This paper focuses on a more recent development
  - Time Series Foundational Models (TSFMs) or Time Series Language Models (TSLMs)
  - Large-scale, general-purpose neural networks pre-trained on large amounts of diverse data across various frequencies and domains
  - TSLMs build on LLM architecture to uncover intricate nonlinear relationships in time series data

Many TSLMs out there...

- Several TSLMs have already been productionalized and are publicly available:
  - LagLlama (February 2024)
  - Chronos (Amazon, March 2024)
  - Moirai (Salesforce, March 2024)
  - Tiny Time Mixers (IBM, April 2024)
  - TimesFM (Google, April 2024)
  - Time-GPT (Nixtla, May 2024)[1]
- These models are used to accomplish a variety of time series related tasks, ranging from prediction and classification to anomaly detection and data imputation

[1] Use of Time-GPT is done through a proprietary API, with some limitations.

This paper

1. Are TSLMs actually useful for Macro Forecasting?
2. How do TSLMs fare relative to state-of-the-art macro time series methods?
    - Bayesian Vector Autoregressions
    - Factor models

- Note: challenges in using TSLMs for forecasting
    - "Data leakages"
    - Not trivial to run a proper out of sample exercise

This paper

1. Are TSLMs actually useful for Macro Forecasting?
2. How do TSLMs fare relative to state-of-the-art macro time series methods?
    - Bayesian Vector Autoregressions
    - Factor models

- Note: challenges in using TSLMs for forecasting
    - "Data leakages"
    - Not trivial to run a proper out of sample exercise

Results

The setup

- We carried out a pseudo real-time forecasting exercise using the FRED monthly database and an evaluation sample ranging from 1985 to 2023
- We forecast up to 12 months out and focused on point forecast accuracy (*RMSFE*)

Main take-aways

1. Two out of five TSLMs are competitive against the AR(1) benchmark (Moirai and TimesFM)
2. Moirai and TimesFM perform generally on par with BVARs and factor models, but are more erratic in their performance
3. In general, TSLMs show less reliability, and are prone to produce occasional unreasonable forecasts
4. TSLMs forecasting performance varies wildly for persistent series

Results

The setup

- We carried out a pseudo real-time forecasting exercise using the FRED monthly database and an evaluation sample ranging from 1985 to 2023
- We forecast up to 12 months out and focused on point forecast accuracy (*RMSFE*)

Main take-aways

1. Two out of five TSLMs are competitive against the AR(1) benchmark (Moirai and TimesFM)
2. Moirai and TimesFM perform generally on par with BVARs and factor models, but are more erratic in their performance
3. In general, TSLMs show less reliability, and are prone to produce occasional unreasonable forecasts
4. TSLMs forecasting performance varies wildly for persistent series

Outline of the talk

Literature Review: Time-series & LLMs

- **Time series for LLMs (data centric approaches):**
  - PromptCast (Xue and Salim, 2023)
  - One fits all (Zhou et al., 2023)
  - TEST (Sun et al., 2024)
  - LLaTA (Liu et al., 2024)
  - LLTIME (Gruver et al., 2024)
  - LLM4TS (Chang et al., 2024)
  - TEMPO (Cao et al., 2024)
- **LLMs for time series (model centric approaches):**
  - Lag-Llama (Rasul et al., 2024), Moirai (Woo et al., 2024), Time-GPT (Garza and Mergenthaler-Canseco, 2023), TimesFM (Das et al., 2024), Tiny Time Mixers (Ekambaram et al., 2024)
  - Chronos (Ansari et al., 2024)
  - Moment (Goswami et al., 2024)
  - WaveToken (Masserano et al., 2024)

Literature Review: Time-series & LLMs

- **Time series for LLMs (data centric approaches):**
  - PromptCast (Xue and Salim, 2023)
  - One fits all (Zhou et al., 2023)
  - TEST (Sun et al., 2024)
  - LLaTA (Liu et al., 2024)
  - LLTIME (Gruver et al., 2024)
  - LLM4TS (Chang et al., 2024)
  - TEMPO (Cao et al., 2024)
- **LLMs for time series (model centric approaches):**
  - Lag-Llama (Rasul et al., 2024), Moirai (Woo et al., 2024), Time-GPT (Garza and Mergenthaler-Canseco, 2023), TimesFM (Das et al., 2024), Tiny Time Mixers (Ekambaram et al., 2024)
  - Chronos (Ansari et al., 2024)
  - Moment (Goswami et al., 2024)
  - WaveToken (Masserano et al., 2024)

Literature Review: Macro and Finance Applications

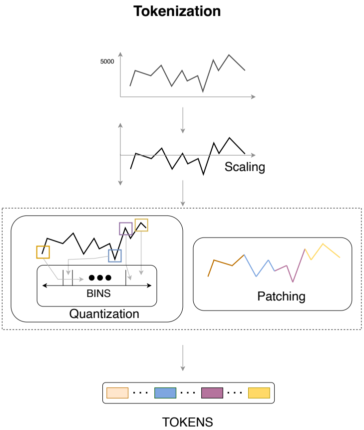
- Chen et al. (2022): **BERT**, **RoBERTa**, **OPT** + Thomson Reuters Real-time News Feed (RTRS) → Predicting firm-level daily returns.
- Kim et al. (2024): **ChatGPT-4** + financial statements → Predicting firm-level earnings.
- Bybee (2023): **ChatGPT-3.5** + WSJ articles → Predicting financial and macroeconomic variables.
- Chen et al. (2025): **ChatGPT-3.5, DeepSeek** + WSJ articles → Predicting aggregate stock market returns.
- Faria-e Castro and Leibovici (2024): **Google AI's PaLM** → Predicting inflation.
- Araujo et al. (2025): **Word2Vec** + ECB monetary policy press conferences → Predicting inflation.
- Alonso (2024): **ChatGPT-4o**, **Gemini Advanced**, **Claude 3.5 Sonnet** + FOMC minutes & policy statements → Forecasting GDP growth, inflation, and unemployment.

Time Series Language Models (TSLMs)

- TSLMs bridge the gap between LLMs original text data training and the numerical nature of time series data
- Main idea:
  1. Train a foundational model on a very large set of time series, $\boldsymbol{X}_{1:T} = (x_{1,1:T}, \ldots, x_{N,1:T})$ and obtain very large set of coefficients $\widehat{\theta}$ which describe a mapping function $f_{\widehat{\theta}}$
  2. Use $f_{\widehat{\theta}}$ along with the current and past values of time series of interest $y_{1:T}$, to forecast future values of y,

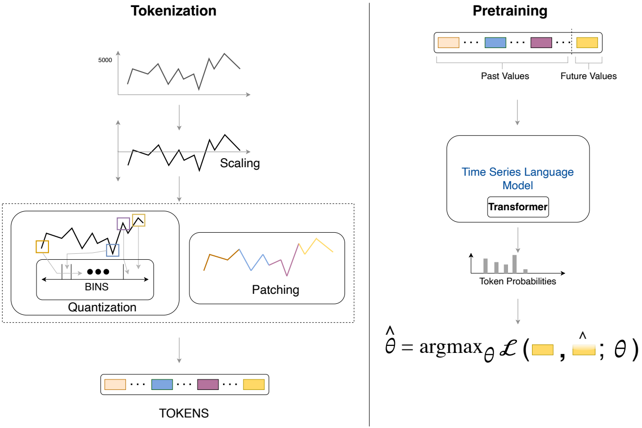
$$\widehat{y}_{T+1:T+H|T} = f_{\widehat{\theta}}(y_{1:T}) = f\left(y_{1:T}|\boldsymbol{X}_{1:T}, \widehat{\theta}\right)$$

Components of Time Series Language Models



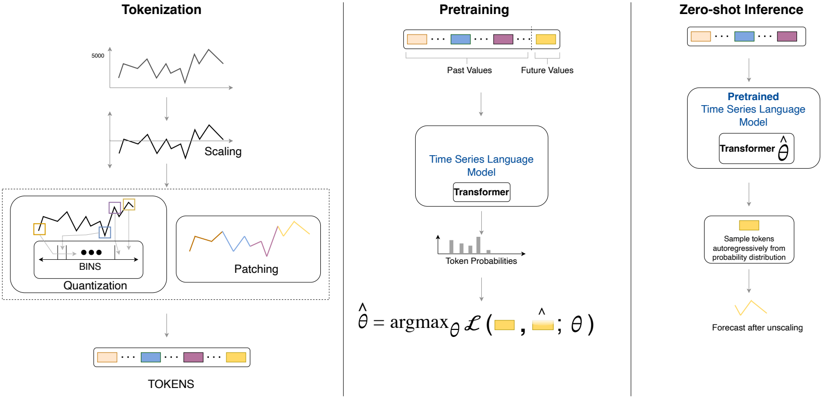
1. Raw time series is tokenized into discrete tokens Scaling Patching Quantization
2. Tokens are used to pre-train a Transformer-based model
3. The pretrained model can make zero-shot forecasts of a new time series

Components of Time Series Language Models



1. Raw time series is tokenized into discrete tokens Scaling Patching Quantization
2. Tokens are used to pre-train a Transformer-based model
3. The pretrained model can make zero-shot forecasts of a new time series

Components of Time Series Language Models



1. Raw time series is tokenized into discrete tokens Scaling Patching Quantization
2. Tokens are used to pre-train a Transformer-based model
3. The pretrained model can make zero-shot forecasts of a new time series

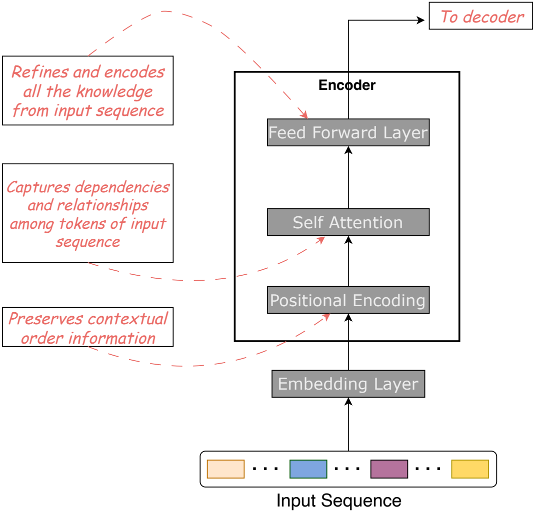
Tokenization: Considerations

- Choice of scaling and tokenization method depends on the specific TSLM
- Scaling:
  - Can be applied at global level, context window level, or patch level
- Patching:
  - Larger patch sizes for high-frequency time series
  - Smaller patch sizes for low-frequency time series
- Quantization:
  - Uniform quantization: equal-sized bins
  - Data-dependent quantization: adjusts bin sizes based on data distribution

Model Architecture

- Transformer-based architecture. It is a multilayered ANN that transforms an input sequence into an output sequence by understanding the context and relationships between the elements within the sequence.
- Embed the tokens into vectors (each dimension in the vector space represents some relevant dimension of meaning)
- Encode information via several layers of self-attention and perceptors
  - Self-attention mechanisms capture context and long-range dependencies (e.g. what the word "row" means depending on context)
- A nice resource to visualize all of this: https://www.3blue1brown.com/lessons/gpt

Encoder Architecture



Time Series Augmentation

- Helps mitigate scarcity of time series data
- Used to generate more diverse training data
- Techniques include:
  - Convex combinations of existing time series
  - Combinations of ARMA processes, seasonal patterns, trends
  - Combining frequency spectrum of sequences

TSLM Training Details

Model	Release date	Training datasets (domains)	Data size	Model size
LagLlama	Feb 2024	Traffic, Uber TLC, Electricity, London Smart Meters, Solar power, Wind farms, KDD Cup 2018, Sunspot, Beijing Air quality, Air Quality UC Irvine Repository, Huawei cloud, Econ/Fin*	352M tokens	2.45M
Moirai	Mar 2024	Energy, Transport, Climate, CloudOps, Web, Sales, Nature, Econ/Fin*, Healthcare	27B obs.	311M
TTM	April 2024	Electricity, Web traffic, Solar power, Wind farms, Energy consumption, KDD Cup 2018, Sunspot, Australian weather, US births, Bitcoin, Econ/Fin*	1B obs.	4M
Time-GPT	May 2024	Finance, economics, Demographics, Healthcare, Weather, IoT sensor data, Energy, Web traffic, Sales, Transport, and Banking	100B obs.	
TimesFM	May 2024	Google Trends, Wiki Page views, M4 Competion, Electricity and the Traffic data, Weather data, Synthetic Time Series Data	100B obs.	200M

Table: Training datasets for the various TSLMs considered in this paper. * indicates that the training data include the Monash forecast repository, and therefore includes a large part of the FRED-MD dataset.

1. Bayesian VAR with Natural Conjugate Priors

- Collect all N time series of interest in $y_t = (y_{1t}, ..., y_{Nt})$ and write a VAR(p) model as:

$$y_t = \Phi_c + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + ... + \Phi_p y_{t-p} + \varepsilon_t;\ \varepsilon_t \sim i.i.d.N(0, \Sigma)$$

- Natural conjugate N-IW prior $+$ Minnesota-style layout

$$\Phi|\Sigma \sim N(\Phi_0, \Sigma \otimes \Omega_0),\ \Sigma \sim IW(S_0, v_0)$$

- Augment prior with "sum of coefficients" and "single unit root" priors
- Conjugacy and Kronecker structure in the priors keep computations manageable even in large systems, and yield marginal likelihood in closed form $\Rightarrow$ Exploit this result to optimize prior hyperparameters

2. Bayesian VAR with Asymmetric Conjugate Priors

- Natural conjugate prior rules out cross-variable shrinkage
- Chan (2022) extended this setup to allows for asymmetry in the prior while maintaining conjugacy. Starting point is the BVAR in its structural form:

$$Ay_t = b + B_1 y_{t-1} + B_2 y_{t-2} + \ldots + B_p y_{t-p} + u_t; \quad u_t \sim i.i.d.\ N(0, D)$$

  where A is a lower triangular matrix and D is diagonal. This allows for estimation recursively, one equation at a time

- Prior assumes that all the BVAR parameters are a priori independent across equations
- As in the previous case, prior is augmented with "sum of coefficients" and "single unit root" priors

3. Factor Model

- Factor models are another class of models that has repeatedly shown to be well suited for macroeconomic forecasting
- We proceeed in two steps:
  1. A set of static factors is estimated from the whole cross section of available data
  2. Use the extracted first factor to augment an autoregression of the *i*-th series with its lag values, i.e.:

$$y_{i,t} = \alpha_h + \beta_h(L)\widehat{f}_{1,t-h} + \gamma_h(L)y_{i,t-h} + \varepsilon_{i,t}$$

- Direct *h*-step ahead forecasts are given by

$$\widehat{y}_{i,t+h} = \widehat{\alpha}_h + \widehat{\beta}_h(L)\widehat{f}_{1,t} + \widehat{\gamma}_h(L)y_{i,t}$$

4. DeepAR (Salinas et al 2019)

Model Structure:

- Autoregressive RNN that takes $(y_{i,1:t}, \mathbf{X}_{-i,1:t-1})$ as inputs
- Outputs parameters of the predictive distribution for $y_{i,t+1:t+h}$

Training Objective: Maximize (Gaussian) likelihood over all time periods and all series:

$$\mathcal{L}_t = \sum_{i=1}^{N} \sum_{\tau=1}^{t} \log p(y_{i,\tau} \mid \theta(h_{i,\tau}, \Theta))$$

where

$$h_{i,\tau} = h\left(h_{i,\tau-1}, y_{i,\tau-1}, \mathbf{X}_{-i,\tau-1}; \Theta\right)$$

is a function implemented by a multi-layer recurrent neural network with Long Short-Term Memory (LSTM) cells.

DeepAR Training and Tuning

Empirical Application

- US monthly macro time series spanning January 1959 to December 2023
- Data source: Federal Reserve Economic Data Monthly Dataset (FRED-MD) at https://fred.stlouisfed.org.
- FRED-MD covers 120+ key macroeconomic variables (output, prices, interest rates, etc.) $\rightarrow$ standard variable transformations applied
- Forecast from January 1985 to December 2019 (2023) using both pre-trained TSLMs and econometrics models
- BVARs, factor models, and DeepAR estimated using an expanding window approach and predictive simulation for $h = 1$ to 12 months ahead
- Model sizes: Medium (19 variables), Large (39 variables), **X-large (120 variables)**

Measuring Predictive Accuracy

- Measure the accuracy of the h-step-ahead point forecasts for model i and variable j, relative to that from the univariate AR(1), using relative Root MSFEs:

$$RMSFE_{ijh} = \sqrt{\frac{\sum_{\tau=\underline{t}}^{\overline{t}-h} e^2_{i,j,\tau+h}}{\sum_{\tau=\underline{t}}^{\overline{t}-h} e^2_{bcmk,j,\tau+h}}},$$

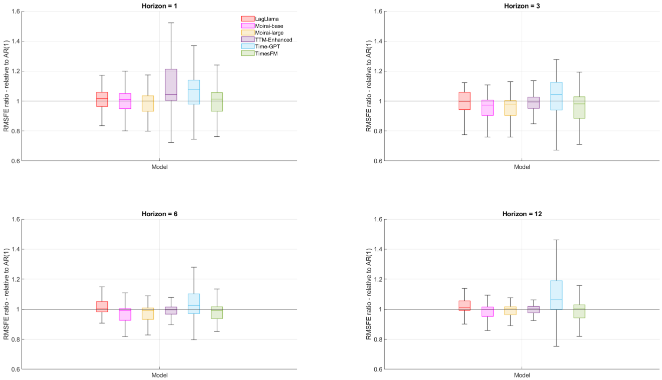
  where $\underline{t}$ and $\overline{t}$ denote the start and end of the out-of-sample period and model $i \in \{\text{BVAR v1}, \text{BVAR v2}, \text{DeepAR}, \text{Factor model}\} \cup \text{TSLMs}$ and the AR(1) model

- For BVARs and DeepAR, point forecasts is computed as the median of predictive densities

- Focus on point forecast accuracy due to complexity in constructing density forecasts with TSLMs

TSLMs Performance Comparison

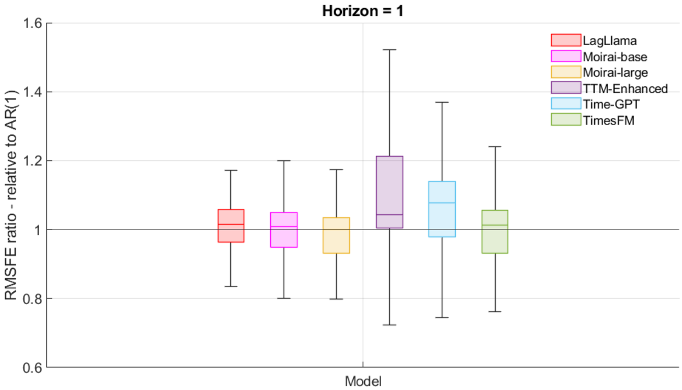
Distribution of RMSFEs



[Zoom on $h = 1$ panel]

TSLMs Performance Comparison

Distribution of RMSFEs



[Back to full figure]

TSLMs Detailed Performance Statistics

RMSFE by model type and forecast horizon

	h=1				h=3			
	Median	**Std**	**Min**	**Max**	**Median**	**Std**	**Min**	**Max**
LagLlama	1.015	1.057	0.726	7.271	0.997	0.787	0.633	4.843
Moirai-base	1.008	0.097	0.704	1.204	0.973	0.100	0.634	1.107
Moirai-large	0.999	0.102	0.703	1.436	0.978	0.099	0.637	1.158
TimesFM	1.014	0.129	0.706	1.482	0.980	0.127	0.635	1.318
TTM-Enhanced	1.044	0.352	0.723	2.959	0.993	0.108	0.718	1.448
Time-GPT	1.077	0.124	0.745	1.531	1.044	0.134	0.672	1.278
	h=6				**h=12**			
	Median	**Std**	**Min**	**Max**	**Median**	**Std**	**Min**	**Max**
LagLlama	1.002	0.461	0.568	3.577	1.009	0.260	0.597	2.431
Moirai-base	0.990	0.093	0.567	1.109	0.999	0.098	0.594	1.168
Moirai-large	0.991	0.096	0.600	1.159	1.001	0.113	0.619	1.324
TimesFM	0.990	0.142	0.593	1.629	1.001	0.158	0.482	1.440
TTM-Enhanced	0.995	0.097	0.643	1.400	1.002	0.198	0.663	2.257
Time-GPT	1.025	0.140	0.600	1.363	1.063	0.169	0.611	1.515

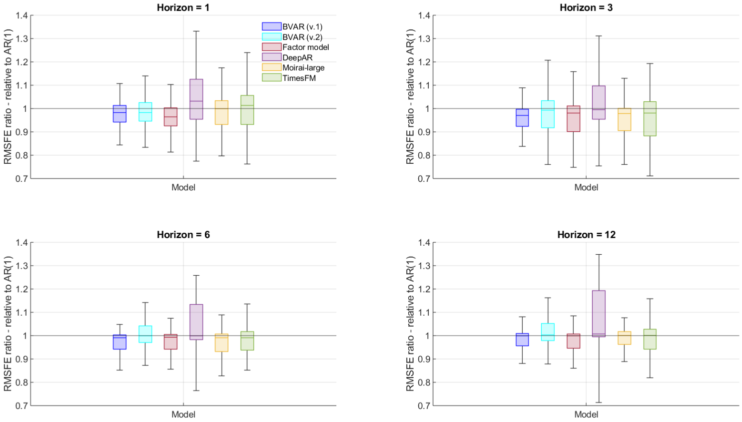
Table: Selected RMSFE statistics by TSLM model type and forecast horizon

Key Takeaways

- Heterogeneity: Significant performance differences among TSLMs
- Top performers: Moirai Large and TimesFM
- Underperformers: TTM-enhanced and Time-GPT
- Outliers: TSLMs can produce extremely inaccurate forecasts in some cases
- Recommendation: Use TSLMs with caution and monitor results carefully

Comparing TSLMs and Econometric Models

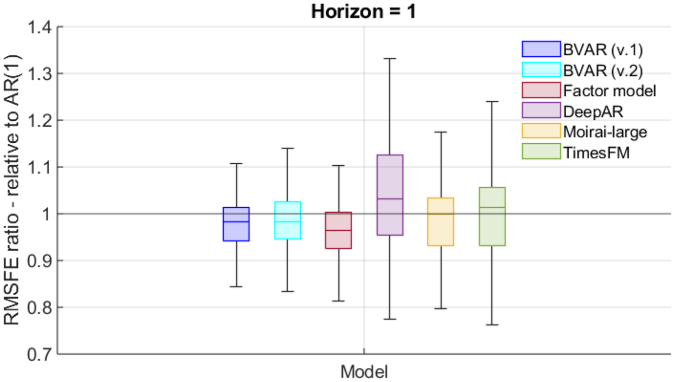
Distribution of RMSFEs



[Zoom on $h = 1$ *panel]*

Comparing TSLMs and Econometric Models

Distribution of RMSFEs



[Back to full figure]

Comparing TSLMs and Econometric Models

RMSFE by model type and forecast horizon

	h=1				**h=3**			
	Median	**Std**	**Min**	**Max**	**Median**	**Std**	**Min**	**Max**
BVAR (v.1)	0.983	0.060	0.827	1.158	0.970	0.060	0.767	1.089
BVAR (v.2)	0.982	0.081	0.800	1.315	0.992	0.099	0.732	1.243
Factor model	0.965	0.065	0.766	1.103	0.980	0.102	0.682	1.183
DeepAR	1.031	0.296	0.7774	2.341	0.995	0.340	0.667	2.536
Moirai-large	0.999	0.102	0.703	1.436	0.978	0.099	0.637	1.158
TimesFM	1.014	0.129	0.706	1.482	0.980	0.127	0.635	1.318
	h=6				**h=12**			
	Median	**Std**	**Min**	**Max**	**Median**	**Std**	**Min**	**Max**
BVAR (v.1)	0.991	0.068	0.725	1.106	0.999	0.080	0.671	1.162
BVAR (v.2)	1.000	0.095	0.714	1.222	1.004	0.111	0.611	1.343
Factor model	0.992	0.087	0.602	1.105	0.999	0.089	0.633	1.125
DeepAR	0.999	0.375	0.641	2.762	1.007	0.470	0.712	3.213
Moirai-large	0.991	0.096	0.600	1.159	1.001	0.113	0.619	1.324
TimesFM	0.990	0.142	0.593	1.629	1.001	0.158	0.482	1.440

Table: Selected RMSFE statistics for econometric models and best performing TSLMs

Key Takeaways

- Both the best TSLMs and econometric models generally outperform AR benchmark
- Econometric models offer more stable and reliable performance (RMSFE distribution consistently below 1)
- TSLMs show higher variance in the RMSFE distribution (higher likelihood of observing large and small forecast errors)

How do TSLMs work with persistent series?

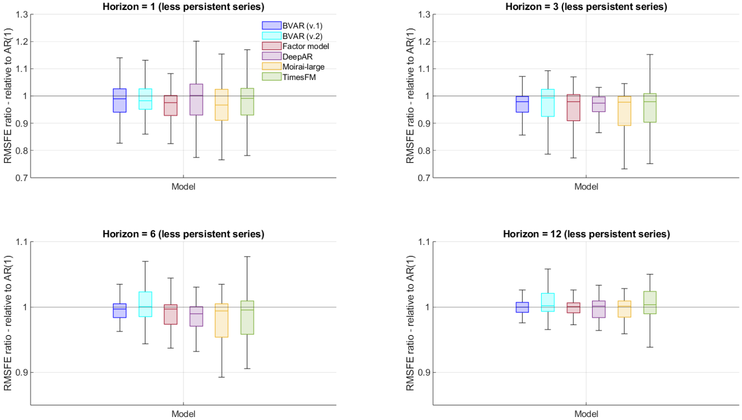


Figure: Distribution of RMSFEs (relative to an AR benckmark), focusing on variables with low persistence. The evaluation sample is January 1985 to December 2019.

How do TSLMs work with persistent series?

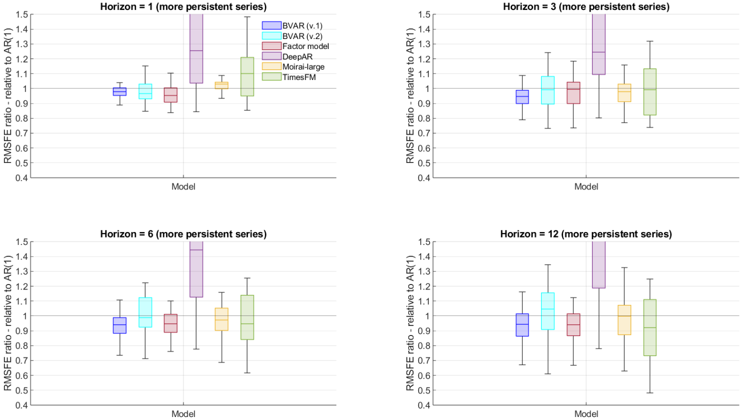


Figure: Distribution of RMSFEs (relative to an AR benckmark), focusing on variables with high persistence. The evaluation sample is January 1985 to December 2019.

Examples: an exceptionally good TSLM forecast

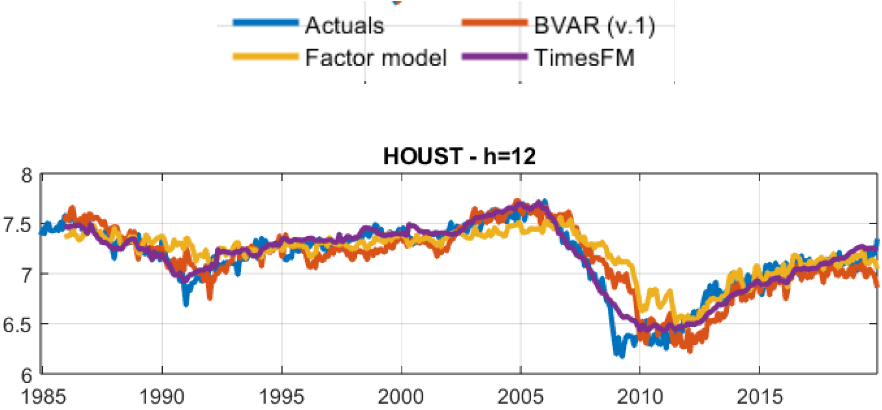


Figure: Housing Starts, 12-step ahead forecasts

Examples: a not so good TSLM forecast (hallucination)

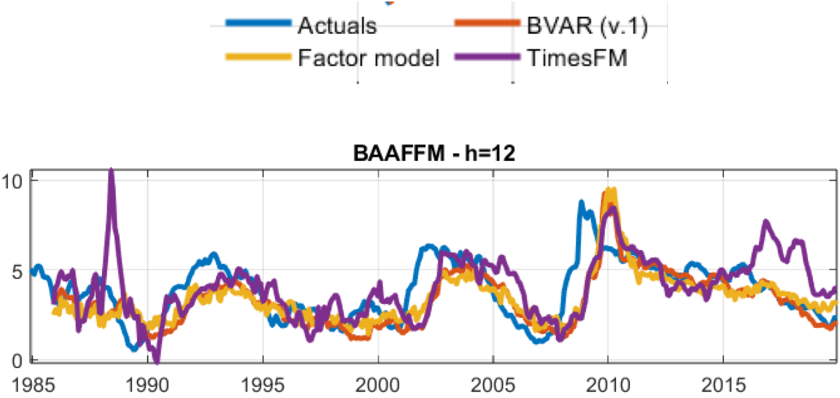


Figure: Moodys Baa Corporate Bond Minus FEDFUNDS, 12-step ahead forecasts

Results for Medium VAR

Go to: Medium VAR

Fine tuning

- Fine-tuning means updating the pretrained model's parameters on a task-specific dataset
- We selected the two best models (TimesFM and Moirai-large) – and fine-tuned them on FRED-MD data.
- The FRED-MD dataset is relatively small, risk of overfitting
- We employ a simple parameter-efficient fine-tuning strategy as opposed to continued-pretraining
- In particular, we freeze the transformer layers of the two TSLMs, thereby reducing the number of trainable parameters.

Fine tuning

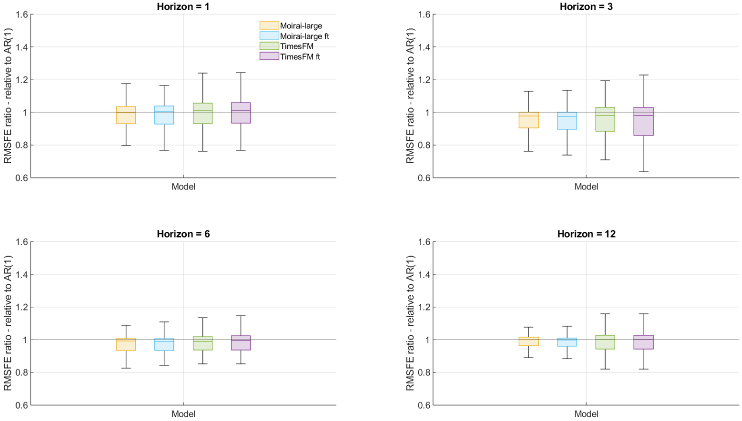


Figure: Distribution of RMSFEs (relative to an AR benckmark) for TimesFM and Moirai-large for zero shot and fine-tuning.

Key takwaways

- Fine-tuning yields results that are similar or slightly better than the zero-shot performance for both models.
- Given the marginal performance gains achieved through fine-tuning in our context, the inherent "plug-and-play" simplicity of zero-shot forecasting presents itself as a potentially more practical approach.

Conclusions

- We evaluated the forecasting performance of Time Series Language Models (TSLMs) for macroeconomic forecasting.
- Among five models tested, only Salesforce's Moirai and Google's TimesFM consistently outperform a simple AR benchmark—but not traditional methods like BVARs and Factor Models.
- A major challenge is limited control over training data—e.g., models pretrained on FRED-MD may lead to biased pseudo out-of-sample forecasts.
- Despite limitations, TSLMs show promise in handling nonlinearities and structural breaks, particularly in the post-Covid-19 period.
- Future research could explore hybrid models combining TSLM flexibility with econometric structure and theory

References I

Alonso, M. N. I. (2024, September). Large language models as macroeconomists. SSRN Working Paper SSRN 4951445, Artificial Intelligence in Finance Institute, New York, NY. Available at https://ssrn.com/abstract=4951445.

Ansari, A. F., L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor, et al. (2024). Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

Araujo, D., N. Bokan, F. Comazzi, and M. Lenza (2025). Word2Prices: Embedding Central Bank Communications for Inflation Prediction. ECB Working Paper No. 2025/3047, Available at SSRN: https://ssrn.com/abstract=5201335.

Bybee, L. (2023). Surveying generative ai's economic expectations. *arXiv preprint arXiv:2305.02823*.

Cao, D., F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, and Y. Liu (2024). TEMPO: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*.

References II

Chan, J. C. C. (2022). Asymmetric conjugate priors for large bayesian vars. *Quantitative Economics 13*(3), 1145–1169.

Chang, C., W.-Y. Wang, W.-C. Peng, and T.-F. Chen (2024). LLM4TS: Aligning pre-trained llms as data-efficient time-series forecasters. *arXiv preprint arXiv:2308.08469*.

Chen, J., G. Tang, G. Zhou, and W. Zhu (2025). ChatGPT and Deepseek: Can They Predict the Stock Market and Macroeconomy? Available at https://arxiv.org/pdf/2502.10008.

Chen, Y., B. T. Kelly, and D. Xiu (2022). Expected returns and large language models. Available at SSRN: https://ssrn.com/abstract=4416687.

Das, A., W. Kong, R. Sen, and Y. Zhou (2024). A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*. PMLR.

References III

Ekambaram, V., A. Jati, N. H. Nguyen, P. Dayama, C. Reddy, W. M. Gifford, and J. Kalagnanam (2024). Ttms: Fast multi-level tiny time mixers for improved zero-shot and few-shot forecasting of multivariate time series. *arXiv preprint arXiv:2401.03955*.

Faria-e Castro, M. and F. Leibovici (2024). Artificial intelligence and inflation forecasts. Federal Reserve Bank of St. Louis Working Paper Series.

Garza, A. and M. Mergenthaler-Canseco (2023). TimeGPT-1. *arXiv preprint arXiv:2310.0358*.

Goswami, M., K. Szafer, A. Choudhry, Y. Cai, S. Li, and A. Dubrawski (2024). Moment: A family of open time-series foundation models. In *International Conference on Machine Learning*. PMLR.

Gruver, N., M. Finzi, S. Qiu, and A. G. Wilson (2024). Large language models are zero-shot time series forecasters. *arXiv preprint arXiv:2310.07820*.

Harvey, D., S. Leybourne, and P. Newbold (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting 13*(2), 281 – 291.

References IV

Kim, A. G., M. Muhn, and V. V. Nikolaev (2024). Financial statement analysis with large language models. Available at SSRN: https://ssrn.com/abstract=4835311.

Liu, P., H. Guo, T. Dai, N. Li, J. Bao, X. Ren, Y. Jiang, and S.-T. Xia (2024). Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. *arXiv preprint arXiv:2403.07300*.

Masserano, L., A. F. Ansari, B. Han, X. Zhang, C. Faloutsos, M. W. Mahoney, A. G. Wilson, Y. Park, S. Rangapuram, D. C. Maddix, and Y. Wang (2024). Enhancing foundation models for time series forecasting via wavelet-based tokenization.

Rasul, K., A. Ashok, A. R. Williams, H. Ghonia, R. Bhagwatkar, A. Khorasani, M. J. D. Bayazi, G. Adamopoulos, R. Riachi, N. Hassen, M. Biloš, S. Garg, A. Schneider, N. Chapados, A. Drouin, V. Zantedeschi, Y. Nevmyvaka, and I. Rish (2024). Lag-llama: Towards foundation models for probabilistic time series forecasting.

References V

Sun, C., H. Li, Y. Li, and S. Hong (2024). TEST: Text prototype aligned embedding to activate llm's ability for time series. *arXiv preprint arXiv:2308.08241*.

Woo, G., C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo (2024). Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*.

Xue, H. and F. D. Salim (2023). Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 1–14.

Zhou, T., P. Niu, X. Wang, L. Sun, and R. Jin (2023). One fits all:power general time series analysis by pretrained lm. *arXiv preprint arXiv:2302.11939*.

Tokenization: Scaling

- To ensure consistent processing, data going into TSLM are typically re-scaled
- Helps in optimization/learning for deep learning models
- General scaling formula:

$$\tilde{x}_t = \frac{x_t - M}{S}$$

  $\tilde{x}_t$ is the scaled value, M is a measure of central tendency, and S is a measure of spread.

- Example (LagLlama):
  - M = median of the context window
  - S = inter-quartile range within the context window

Back

Tokenization: Patching

- Patching divides time series into fixed-length segments (patches)
- Patching allows to capture local patterns
- Patch size is a hyper-parameter
- Patches can be overlapping or non-overlapping
- Example: $x_{1:T} = \{4.7, 4.76, 6.8, 7.2, 6.1\}$:
    - Patch size = 3, overlap = 2:
        - $\{4.7, 4.76, 6.8\}$
        - $\{4.76, 6.8, 7.2\}$
        - $\{6.8, 7.2, 6.1\}$

Back

Tokenization: Quantization

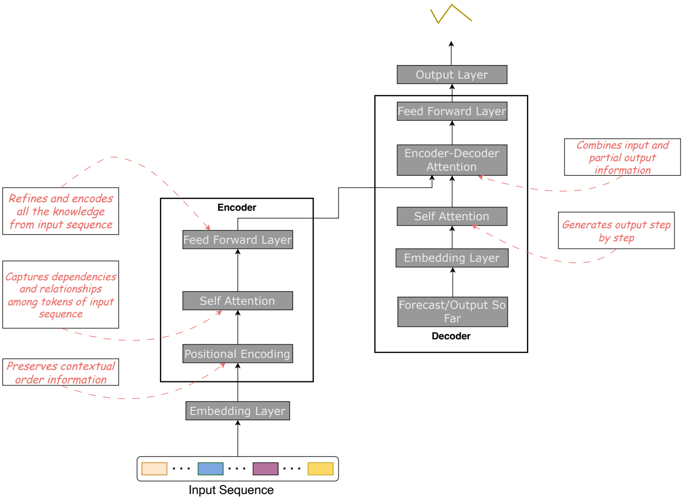
- Quantization is used to convert numerical values into discrete tokens
- It divides the time series into a predefined number of bins $\mathbb{B}$
- Each data point assigned a token (a number between 1 and $\mathbb{B}$) based on its bin
- Example for $x_{1:T} = \{4.7, 4.76, 6.8, 7.2, 6.1\}$ with $\mathbb{B} = 4$ and uniform binning:

$$q_t = \begin{cases} 1 & \text{if } 4 \leq x_t < 5 \\ 2 & \text{if } 5 \leq x_t < 6 \\ 3 & \text{if } 6 \leq x_t < 7 \\ 4 & \text{if } 7 \leq x_t < 8 \end{cases}$$

- Resulting tokenized sequence: $q_{1:T} = \{1, 1, 3, 4, 3\}$

Back

Model Architecture



back

DeepAR: Training and Tuning Considerations

Hyperparameters to Tune:

- **Context length:** Number of past time periods used for conditioning
- **RNN type:** Vanilla RNN, LSTM, GRU
- **RNN depth:** Number of layers and units
- **Learning rate and batch size:** Standard NN tuning

Likelihood function

- Gaussian: $\mathcal{N}(\mu_\tau, \sigma_\tau^2)$ with

$$\mu_\tau = \boldsymbol{w}_\mu' h_{i,\tau} + \boldsymbol{b}_\mu$$

and

$$\sigma_\tau = \log\left(1 + \exp\left(\boldsymbol{w}_\sigma' h_{i,\tau} + \boldsymbol{b}_\sigma\right)\right)$$

at each step

Forecasting:

- Produces density forecasts
- Roll out forecasts by sampling recursively from predicted distributions

Back to Main

Results for Medium model

- Focus: 19 variables from Medium model
- Display *RMSFE* ratios relative to AR(1) benchmark
- Include Diebold-Mariano (DM) t-statistic
  - Serial correlation robust std. errors
  - Harvey et al. (1997) small-sample adjustment
- Evaluation period: January 1985 to December 2019

Variables in Medium VAR

Abbreviation	Description	Transformation
PAYEMS	All Employees: Total nonfarm	5
INDPRO	IP Index	5
FEDFUNDS	Effective Federal Funds Rate	1
UNRATE	Civilian Unemployment Rate	1
RPI	Real personal income	5
DPCERA3M086SBEA	Real PCE	5
CMRMTSPLx	Real Manu. and TradeIndustries Sales	5
CUMFNS	Capacity Utilization: Manufacturing	1
CES0600000007	Avg Weekly Hours: Goods-Producing	1
HOUST	Housing Starts, Total	4
S&P 500	S&P's Common Stock Price Index: Composite	5
T1YFFM	1-Year Treasury C Minus FEDFUNDS	1
T10YFFM	10-Year Treasury C Minus FEDFUNDS	1
BAAFFM	Moodys Baa Corporate Bond Minus FEDFUNDS	1
EXUSUKx	U.S.-UK Foreign Exchange Rate	5
WPSFD49207	PPI: Final Demand: Finished Goods	5
PPICMM	PPI: Metals and metal products	5
PCEPI	Personal Consumption Expenditures	5
CES0600000008	Avg Hourly Earnings: Goods-Producing	6

Table: Variables and transformations in the Medium model. (1) no transformation, (2) Δx_t, (5) $\Delta \log (x_t)$, (6) $\Delta^2 \log (x_t)$ with Δ^i indicating i th differences.

RMSFE ratios, $h = 1$

	$h=1$									
	BVAR(v.1)		BVAR(v.2)		Factor model		Moirai Large		TimesFM	
PAYEMS	0.84	***	0.83	***	0.87	***	0.83	***	0.82	***
INDPRO	0.90	***	0.93	**	0.97		0.95	**	0.94	**
FEDFUNDS	1.21		0.97		0.87	**	1.00		1.23	
UNRATE	0.88	***	0.84	***	0.89	***	1.04		1.11	
RPI	0.98		0.98		0.99		1.00		1.04	
DPCERA3M086SBEA	1.00		0.99		0.98	*	1.02		1.06	
CMRMTSPLx	0.96	*	0.95	**	0.98		1.01		1.04	
CUMFNS	0.90	*	0.91	**	0.95		1.09		1.48	
CES0600000007	0.90	***	0.91	***	0.91	**	0.93	*	0.94	**
HOUST	0.95	***	0.95	**	0.93	***	1.00		0.92	***
S&P 500	1.04		1.03		1.00		1.03		1.02	
T1YFFM	1.19		1.18		1.06		1.04		1.13	
T10YFFM	1.06		1.00		1.00		1.04		1.16	
BAAFFM	1.02		0.96		0.94		1.01		1.28	
EXUSUKx	1.02		1.03		1.01		1.02		1.01	
WPSFD49207	0.97		1.00		1.03		1.01		1.01	
PPICMM	1.00		1.01		0.99		1.03		1.04	
PCEPI	0.94	***	0.94	**	0.98		0.99		0.95	*
CES0600000008	0.85	***	0.85	***	0.78	***	0.77	***	0.78	***

Table: Differences in accuracy that are statistically significant at 10%, 5%, and 1% levels are denoted by one, two, or three stars, respectively.

RMSFE ratios, $h = 12$

Return to Main

	$h=12$									
	BVAR(v.1)		BVAR(v.2)		Factor model		Moirai Large		TimesFM	
PAYEMS	1.00		0.99		1.00		1.00		0.99	
INDPRO	1.01		1.01		1.01		1.01		0.99	
FEDFUNDS	0.96		1.01		0.96		1.00		1.08	
UNRATE	0.89	***	0.88	***	0.86	**	0.96		0.89	*
RPI	1.00		1.00		1.00		1.00		1.00	
DPCERA3M086SBEA	1.00		1.00		1.00		1.00		1.00	
CMRMTSPLx	1.01		1.01		1.01		1.02		1.02	
CUMFNS	0.91		0.93		1.06		0.92		1.02	
CES0600000007	0.70	***	0.75	***	0.81	***	0.76	***	0.78	***
HOUST	1.03		1.11		1.02		0.96		0.48	**
S&P 500	1.01		1.01		1.00		1.01		1.05	
T1YFFM	1.25		1.50		1.01		1.11		1.25	
T10YFFM	0.87		1.05		0.90	**	1.03		1.15	
BAAFFM	0.94		1.01		0.96		0.96		1.22	
EXUSUKx	1.00		1.01		1.00		1.03		1.00	
WPSFD49207	1.02		1.05		1.03		1.00		1.03	
PPICMM	1.00		1.00		1.01		1.01		1.03	
PCEPI	0.90	*	0.97		0.90	*	0.82	***	0.83	***
CES0600000008	0.88	***	0.88	***	0.81	***	0.78	***	0.80	***

Table: Differences in accuracy that are statistically significant at 10%, 5%, and 1% levels are denoted by one, two, or three stars, respectively.