Applied Intelligence (2025) 55:395
https://doi.org/10.1007/s10489-025-06268-w

Mind the naive forecast! a rigorous evaluation of forecasting models for time series with low predictability

Nico Beck[1,2] · Jonas Dovern[2] · Stefanie Vogl[3]

Accepted: 4 January 2025
© The Author(s) 2025

Abstract
In the field of time series forecasting, numerous machine learning studies have assessed the performance of new methods on highly volatile data from macroeconomics and finance. Unlike in other domains, where models are also compared to simpler statistical or naive baselines, they mostly compare the performance solely relative to other complex models. This approach may lead to limited conclusions and reduce the practical significance of the results, as it overlooks the unpredictability of some highly volatile time series in the datasets used. We apply state-of-the-art methods from time-series econometrics and machine learning, including autoregressive integrated moving average (ARIMA), exponential smoothing (ETS), Bayesian vector autoregressive model (BVAR), long-short term memory neural networks (LSTM), historical consistent neural networks (HCNN), deep vector autoregressive neural networks (DeepVAR), temporal fusion transformers (TFT), and extreme gradient boosting (XGBoost). Our results demonstrate that no method consistently outperforms the naive (no-change) forecast for highly volatile time series from two popular datasets containing exchange rates and stock prices, rendering comparative analysis between complex models less meaningful. In contrast, when applied to more predictable macroeconomic price indices, many of the methods significantly outperform naive forecasts. We find that the performance of machine learning models deteriorates more than that of statistical models for high-volatility time series. This study highlights the critical importance of using appropriate benchmark models, including cost-effective, simple approaches, on datasets that permit meaningful conclusions.

Keywords Time series · Benchmark · Forecasting methods · Naive benchmark · No-Change · Random walk

1 Introduction

Since deep learning became a trend, researchers have been proposing increasingly complex models. In the field of time series forecasting, large-scale models such as transformer-based neural networks have become increasingly popular. These models are typically benchmarked against a selected number of datasets from diverse fields, including macroeconomics and finance. Popular datasets include exchange rates, as published by [17], and stock prices, such as those from the National Association of Securities Dealers Automated Quotations (NASDAQ100), as used by [22]. Such macroeconomic and financial time series are of particular interest to decision makers in banks and other financial intermediaries, firms, government bodies, and international organizations, such as the International Monetary Fund. Many forward-looking investment and policy decisions require predictions for such time series. However, forecasting these time series is particularly challenging because they contain a lot of random fluctuations (noise), exhibit little to no autocorrelation. In addition to their relevance for practitioners, the datasets that we chose are popular for benchmarking, as it is possible to train complex models for them due to the availability of a sufficient data history for constructing training periods.

✉ Nico Beck
nico.beck@iis.fraunhofer.de

Jonas Dovern
jonas.dovern@fau.de

Stefanie Vogl
stefanie.vogl@oth-regensburg.de

1 Fraunhofer IIS, Fraunhofer-Institut für Integrierte Schaltungen IIS, Nordostpark 84, Nürnberg 90411, Germany

2 Friedrich-Alexander-Universität Erlangen-Nürnberg, Lange Gasse 20, Nürnberg 90403, Germany

3 Ostbayerische Technische Hochschule Regensburg, Galgenbergstraße 32, Regensburg 93053, Germany

Published online: 03 February 2025

Table 1 Overview of publications that apply a benchmark on at least one of the exchange rates or NASDAQ100 datasets

	Publication	Data sets	Models	Naive forecast
a)	[17]	Exchange-Rate	AR, Lridge, LSVR, TRMF, Gaussian Process, VAR-MLP, RNN-GRU, LSTNet (2 variants)	No
	[28]	Exchange-Rate	Autoformer, Informer, LogTrans, Reformer, LST-Net, LSTM, TCN	No
	[32]		Informer (2 variants), LogTrans, Reformer, LSTMa, LSTNet, DeepAR, ARIMA, Prophet	No
	[33]	Exchange-Rate	FEDformer (2 variants), Autoformer, Informer, LogTrans, Reformer	No
	[29]	Exchange-Rate	AR, VAR-MLP, Gaussian Process, RNN-GRU, LSTNet, TPA-LSTM, MTGNN (2 variants)	No
	[24]	Exchange-Rate	VAR, GARCH, Vec-LSTM (5 variants) DeepVAR, GP-RNN (3 variants)	No
	[34]	Exchange-Rate	FiLM, FEDformer, Autoformer, S4, Informer, Log-Trans, Reformer, N-Beats, N-HiTS, Snaive	Yes
	[22]	NASDAQ-100	ARIMA, NARX RNN, Encoder-Decoder (2 variants), Attention RNN (2 variants), Input-Attn-RNN (2 variants), DA-RNN (2 variants)	No
	[31]	NASDAQ-100	ARIMA, LSTM, Encoder-Decoder (2 variants), Input-Attn-RNN, DA-RNN, GeoMan, MARNN_1	No
	[8]	Exchange-Rate, NASDAQ-100	DeepTCN, forecastNet, IMV_LSTM, LSTNet, RNN-GRU, Transformer, SATNet	No
	[3]	Exchange-Rate	DeepAR, CNN-QR, Transformer, ETS, ARIMA, SNaive	Yes
b)	[18]	Exchange-Rate	MLP, ERNN, LSTM, RNN-GRU, ESN, CNN, TCN	No
	[36]	Exchange-Rate, NASDAQ-100	DA-RNN, LSTNet, TPA-LSTM	No
c)	[19]	Exchange-Rate	LSTNet, LSTMa, Reformer, LogTrans, Informer, TCN, SCINet, SNaive, Linear	Yes
	[12]	Exchange-Rate	Naive, Autoformer, Film	Yes

We mention all benchmark models. Furthermore, we mark the publications that include a naive forecast in their studies. The table features three sections consisting of studies that a) propose a new model, b) review multiple models and c) mention the difficulties in forecasting highly volatile time series

However, these time series are often difficult to predict. Despite this, studies typically claim state-of-the-art performance by comparing themselves with other complex models. Occasionally, benchmarks include established statistical models such as autoregressive integrated moving average (ARIMA) and vector autoregressive (VAR). However, from an econometric point of view, the main question is: Are these datasets even predictable? To ascertain predictability, econometricians include a naive forecast in the set of benchmark models.[1] In some areas of machine learning, the simplest baselines are also commonly included, such as random guessing for classification tasks. This paper underscores a significant flaw in the field of time series machine learning: most studies neglect a crucial benchmark—the naive forecast. Especially in the macroeconomic and financial context, where decisions have far-reaching consequences, a thorough forecast evaluation is essential that does not only look at relative forecast performance among a set of highly complex models but considers comparison with naive predictions, too. The paper contributes to the literature by demonstrating that, in two commonly considered applications, the naive benchmark outperforms complex machine learning models, thus challenging the practical relevance of their claimed superiority over other complex models.

The literature contains many studies that suffer from the mentioned flaw. Table 1 collects studies that use the exchange rates or NASDAQ100 datasets to compare model performances and summarizes the models used as benchmarks. One type of study (see Table 1a) proposes new models [8, 17, 22, 24, 28, 29, 31–34] or new model implementations [3]. The proposed models are all neural networks with high complexity, and the benchmark models are mostly neural networks as well. Only two studies—those by [3] and [34]—consider a naive forecast as a benchmark.

[1] See [2] for an early example and [1, 4, 10, 11, 23] for more recent forecasting applications in different fields of economics that all compare more complex models to naive benchmarks.

A second type of study considering these datasets is reviews. For instance, [18] and [36] assess whether deep neural networks are effective in time series forecasting in general. Although these reviews aim to provide a comprehensive overview, they do not include a statistical or naive benchmark.

Previous work closely related to our study, as detailed in Table 1c, already focuses on the challenges associated with forecasting highly volatile datasets. [12] demonstrate that, contrary to the claims in [34], the models proposed in [28] and [34] do not outperform the naive forecast on the specified exchange rates. However, this evidence of the naive forecast's superiority is limited to only two models, precluding broader comparisons between methods or generalizations to other statistical or machine learning methods. [19] test linear regression and seasonal naive forecast on eight datasets against neural networks and transformer-based models, such as long- and short-term time-series network (LSTNet) [17]. They found that linear regression and a seasonal naive forecast outperformed the neural networks. The study focuses on long forecast horizons ($\geq$ 96 days for exchange rates) without addressing shorter ones. Consequently, their results offer limited insights for short-term forecasting, as long-term forecasting is more about capturing trends than changes between periods. Additionally, [12] and [17] do not consider statistical time series models. We conclude that a significant gap in literature exists, highlighting the inability of various statistical and machine learning models to effectively predict highly volatile datasets, thereby underscoring the necessity of including the naive forecast as a benchmark.

We contribute to the criticism of the common practice of evaluating forecast performance of complex models only relative to that of other complex models. By performing a pseudo-out-of-sample analysis for the most common statistical and machine learning models on the highly volatile datasets of exchange rates and NASDAQ100, we demonstrate the importance of including a naive forecast in the benchmarks. This enhances studies in the time series machine learning area, making their conclusions more significant. Furthermore, we propose using datasets with higher granularities in the macroeconomic and financial area to test models. These time series are usually very noisy at small granularities, but this noise is reduced by aggregating. One example is the Federal Reserve Economic Data monthly database (FRED-MD) [21]. While common in macroeconomic studies, it is rarely utilized in the machine learning community. We show that it is difficult, but possible, to forecast these indices. Thus, model comparison is more meaningful there, as significant improvements over the naive forecast are achievable, including through machine learning methods. Additionally, to analyze the forecast performances, we prove that they are generalizable and that error differences between machine learning and statistical methods are significantly correlated with volatility for a large proportion of highly volatile variables. Finally, for the first time, we rigorously benchmark historical consistent neural networks (HCNN) against other state-of-the-art methods.

We compare state-of-the-art statistical and machine learning models with the naive forecast across three applications. The study uses publicly available, large, real-world datasets that allow long test periods. We split the datasets sequentially into a training period (70%), a validation period (10%), and a test period (20%). For artificial neural networks, other machine learning models, and the Bayesian vector autoregressive model (BVAR), we utilize the validation period to perform hyperparameter optimization. We determine the hyperparameters of the statistical models in-sample based on the Akaike information criterion (AIC). During the test period, we employ rolling origins to ensure a sufficient sample size of test data. To measure forecast performance, we utilize the mean squared error (MSE), the mean absolute error (MAE), and apply the Diebold-Mariano test to assess the statistical significance of differences in performance.

We present the performance of various models on three datasets: exchange rates, NASDAQ100, and FRED-MD. The models fail to outperform the naive forecast on both the exchange rates and NASDAQ100 datasets. However, on the FRED-MD dataset, long-short term memory neural networks (LSTM), HCNN, deep vector autoregressive neural networks (DeepVAR), temporal fusion transformers (TFT), exponential smoothing (ETS), and ARIMA outperform the naive forecast.

The remainder of this paper is structured as follows. Section 2 outlines the models used in our benchmarks. Section 3 describes the design of the empirical study employed to evaluate the forecast performance of these models. Section 4 summarizes and analyzes the benchmark results, and Section 5 concludes.

2 Models

To fix notation, let $\mathbf{y}_t \in \mathbb{R}^n$, for $t = 1, \ldots, T$, represent a multivariate time series of dimension n. The parameter T denotes both the sequence length and the forecast origin. Our aim is to forecast all n variables in $\mathbf{y}_t$ for future periods $t = T + 1, \ldots, T + \tau$. We employ a variety of statistical/econometric and machine learning methods for forecasting.

2.1 Univariate statistical models

As statistical univariate models, we use ETS, ARIMA models, and a naive forecast.

The naive forecast assumes no changes for all forecast steps, predicting the last observed value, $\hat{\mathbf{y}}_t = \mathbf{y}_T$, for each

step $t = T + 1, \ldots, T + \tau$. Alternatively, it predicts zero changes $\hat{\mathbf{y}}_t = 0$, when forecasting integrated time series.

ETS forecasts values as a weighted sum of previous periods. The weights decrease exponentially for earlier periods, thereby assigning greater impact to more recent observations. The smoothing factor models the degree of exponential decay and determines how much more weight is given to recent periods compared to periods further in the past [15].

Similarly, ARIMA models forecast as a weighted sum of earlier periods. It includes a moving average component that incorporates previous forecasting errors into the predictions. The weights are estimated by Maximum Likelihood Estimation. Further, ARIMA models the stationary d-th differences of time series that are integrated of order d.

The modeling assumptions of the naive forecast, ETS, and ARIMA for the data generation process are very strict. The naive forecast does not attempt to approximate the data generation process at all. This has the advantage that no spurious dependencies are modeled if the data generation process is very noisy or changes frequently, for example due to frequent structural breaks. ETS and ARIMA only have a few parameters and do not allow for non-linear intertemporal dependencies. This is clearly restrictive for time series with complex dependencies. However, both approaches have the advantage of not being prone to overfitting and have been shown to perform very well when time series can be well approximated by linear processes—as is often the case in the fields of macroeconomics and finance. Thus, overall they are potentially good forecasting models for time series with low predictability.

ETS and ARIMA are estimated for each variable and each rolling origin separately. For ARIMA, we limit the sequence length to the last 200 time periods for each rolling origin to reduce computational load. We use the GluonTS framework [3] for the univariate statistical models, which relies on the RForecast package [14] for ARIMA and ETS.

2.2 Multivariate statistical models

Additionally, we employ a BVAR model, widely used in macroeconomic forecasting. It jointly forecasts all variables as a weighted sum of all variables from previous periods. The model can be summarized by

$$\hat{\mathbf{y}}_{t+1} = \mathbf{w} + \sum_{i=0}^{p-1} \mathbf{W}_i \mathbf{y}_{t-i} + \epsilon,$$

with $\mathbf{W}_i \in \mathbb{R}^{n \times n}$ for $i = 0, \ldots, p-1$ and $\mathbf{w}, \epsilon \in \mathbb{R}^n$. The equation is applied recursively to predict multiple forecast horizons. For the entries in $\mathbf{w}, \mathbf{W}_0, \ldots, \mathbf{W}_{p-1}$ we assume independently normally distributed priors, each with mean $\mu = 0$ and standard deviation $\tilde{\sigma} = 1$. Residuals ϵ are drawn from a multivariate normal distribution $\mathcal{N}(\mathbf{0}, \Sigma)$, where $\Sigma = \mathbf{I}_n \sigma$. The entries of σ are drawn independently from a half-normal distribution: $\sigma_i \sim \mathcal{N}^+(0, 1)$ for $i = 1, \ldots, n$. We estimate the model parameters by using the no-U-turn sampler (NUTS), a form of Markov Chain Monte Carlo (MCMC) simulation. From each of four chains, we sample 1000 burn-in draws and 1000 draws that we keep. Since we estimate the model for each rolling origin, and it is very time-consuming to do this on the large datasets, we truncate the sequence length before the last 2000 time steps for each rolling origin. We implemented the model using PyMc [25] and integrated it into GluonTS [3].

2.3 Univariate machine learning models

For univariate machine learning, we have selected extreme gradient boosting (XGBoost) [7]. Within the machine learning community, it is a popular method based on an ensemble of regression trees. XGBoost is a computationally efficient end-to-end model suitable for large datasets. As input features for the model, we utilize time windows consisting of lag q observations from the time series. We independently estimate an XGBoost model for each variable and each forecast period using the implementation provided in GluonTS [3].

Given the considerable success of transformer-based models in numerous studies, we include TFT [20] in our analysis. It uses a self-attention mechanism to detect long-range dependencies between past time steps and the target future. This is implemented with multi-head attention, which uses multiple attention heads to learn different dependencies. In addition, TFT is adapted to suit many applications. It includes a gating mechanism that allows for dataset specific preprocessing. The gating mechanism is able to preprocess past time steps non-linearly, but it also allows to skip the non-linearities, if it is more beneficial for a dataset. This may be the case for small or noisy datasets. We use the GluonTS [3] implementation of TFT.[2]

2.4 Multivariate machine learning models

To compare with multivariate machine learning models, we incorporate a multivariate LSTM into our benchmark [13]. The LSTM is a type of recurrent neural network that includes an additional memory state for retaining information over multiple periods. In each period, the model deletes unnecessary information and adds new information to this state. It allows long-term memory and can improve predictive

[2] Note that we do not include another popular transformer-based approach, the Mamba model, as it has been shown already to underperform relative to other transformer-based models and linear models for the exchange rates dataset [27].

performance. For the lag q, we use the multivariate input $\mathbf{y}_{t-q}, \ldots, \mathbf{y}_t$ and encode the information contained with a LSTM in the state $\mathbf{s}_t$ of the model. With an affine linear transformation, we directly forecast $\hat{\mathbf{y}}_{t+1}, \ldots, \hat{\mathbf{y}}_{t+\tau}$ from $\mathbf{s}_t$. The model is trained on MSE using a sliding window approach to forecast $\hat{\mathbf{y}}_{t+1}, \ldots, \hat{\mathbf{y}}_{t+\tau}$. Furthermore, we train ten identical models with different initialization as an ensemble to guarantee stability of the results. The final forecast is the mean of all the forecasts from the ensemble members.

We include the HCNN because it is specifically designed for macroeconomic and financial time series forecasting [35]. It jointly models and forecasts all variables, similar to VAR. This enables the model to learn the temporal relationships between variables. Like LSTM, HCNN uses the inputs $\mathbf{y}_{T-q}, \ldots, \mathbf{y}_T$ for lag q. However, q can be larger compared to LSTM. The following τ time periods $\hat{\mathbf{y}}_{T+1}, \ldots, \hat{\mathbf{y}}_{T+\tau}$ are recursively forecast by the HCNN. It is trained without a sliding window approach, and each error-backpropagation step minimizes the in-sample MSE of $\hat{\mathbf{y}}_t - \mathbf{y}_t$ for $t = T-q, \ldots, T$. We use ensembles of 50 HCNNs and the implementation of Prosper_nn [5].

As a multivariate model from Table 1, we benchmark DeepVAR [24]. For each variable, DeepVAR uses a global LSTM to calculate a model state. Transformations are used to map the time series $\mathbf{y}_t$ to a multivariate Gaussian distribution. The model is then trained to predict the mean and the covariance matrix of the joint Gaussian distribution of all time series. To reduce the parameters that have to be predicted in the covariance matrix and to allow training on sub-samples, DeepVAR uses a low-rank Gaussian process parametrization. For prediction, samples are drawn from the Gaussian distribution and mapped back to the original distribution with inverse transformations.

We utilize GluonTS [3] for the implementation of DeepVAR and have developed wrappers within GluonTS to facilitate the integration of both LSTM and HCNN.

3 Study design

This section introduces the datasets and describes the quasi-real-time forecasting setup employed.

3.1 Data

We examine the forecast performance across three distinct applications: exchange rates, stock prices, and macroeconomic price indicators. For exchange rates, our analysis utilizes daily data of eight US dollar exchange rates (vis-à-vis Australia, the UK, Canada, Switzerland, China, Japan, New Zealand, and Singapore). We use data from [17] and use it until November 4, 2013.

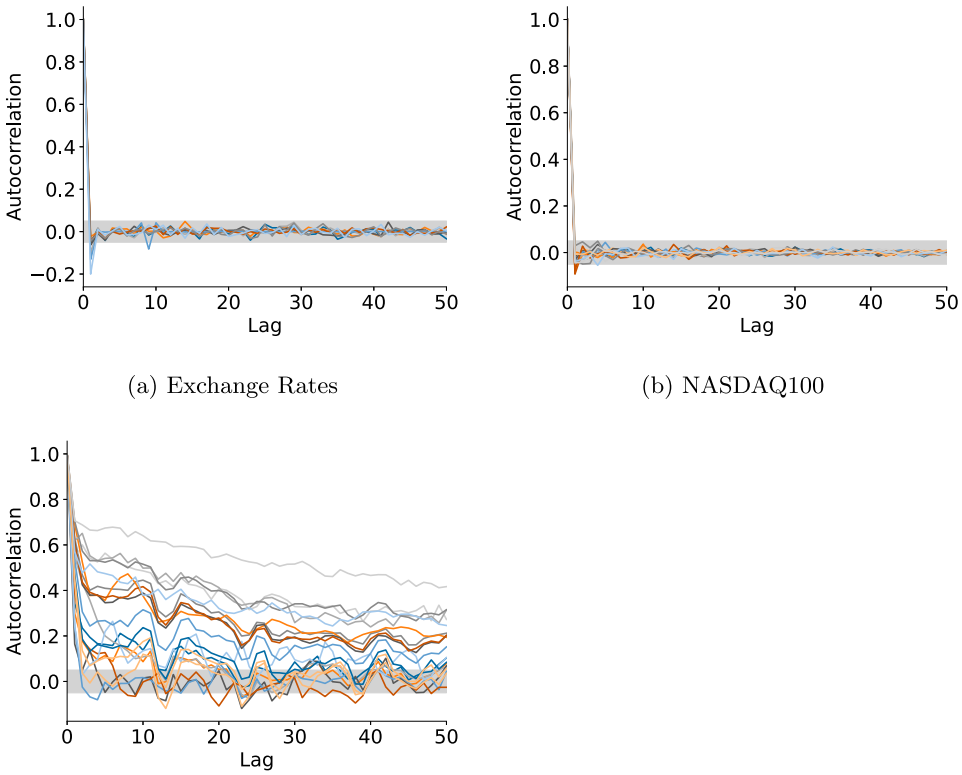
In the stock price application, we employ minute-by-minute data of 81 NASDAQ100 stocks provided by [22]. To decrease the dimension, we select the first 20 stocks listed alphabetically. The sample spans from July 26, 2016, to December 22, 2016, with 390 data points recorded daily.

For macroeconomic price indicators, we utilize all indicators relating to prices from the FRED-MD database published by [21]. This set contains 20 monthly indices, spanning from January 1959 to April 2022 (760 observations). Table 2 represents the key points of the three datasets.

Data analysis on the transformed time series (applying logarithm and first differences as described in Section 3.2) shows the properties of the datasets. Jarque-Bera tests [16] applied to the three datasets show that they are all not normally distributed. This suggests that skewness, outliers, and potentially non-linearities are important. Furthermore, the average standard deviations of these time series are 0.0065, 0.0009, and 0.0112 for the exchange rates, NASDAQ100, and FRED-MD, respectively. Thus, our three analyses cover very different volatility setups. Figure 1 shows the autocorrelation of the variables included in the three datasets to show linear dependencies. All exchange rates and NASDAQ100 stocks

Table 2 Overview over key properties of the used datasets (top) and settings for the forecasting setup (bottom)

	Exchange Rate	NASDAQ100	FRED-MD
Source	[17]	[22]	[21]
Frequency	Day	Minute	Month
Variables	8	20	20
Time steps	6,221	40,560	760
First period	1990-01-01	2016-07-26	1959-01
Last period	2013-11-04	2016-12-22	2022-04
Forecast horizon	20	30	6
Step size validation	36	240	4
Step size test	20	30	1
Number rolling origins	61	268	146
Start validation set	2006-09-08	2016-11-07 05:11:00	2003-03
Start test set	2009-01-27	2016-11-22 01:30:00	2009-08



(a) Exchange Rates

(b) NASDAQ100

(c) FRED-MD

Fig. 1 Autocorrelation functions for exchange rates, NASDAQ100 and FRED-MD are shown in Fig. 1a), b), and c) respectively. Each line in a figure shows the autocorrelation of one variable from zero to fifty lags. The grey area depicts the area of low autocorrelation between -5% and 5%

have low autocorrelations. Only for lag one are a few outside the low autocorrelation region (indicated by the grey color). In contrast, in the FRED-MD dataset all variables have higher autocorrelations for lag one and for most of the variables the high autocorrelation persists over multiple lags. Therefore, the potential for forecasting variables with linear models using only lagged values is much higher for the FRED-MD dataset. For exchange rates and the NASDAQ100, only non-linear lagged information may be available.

3.2 Quasi-real-time forecasting setup

The flow chart in Fig. 2 illustrates the forecasting setup described in this section. The process includes data prepro-

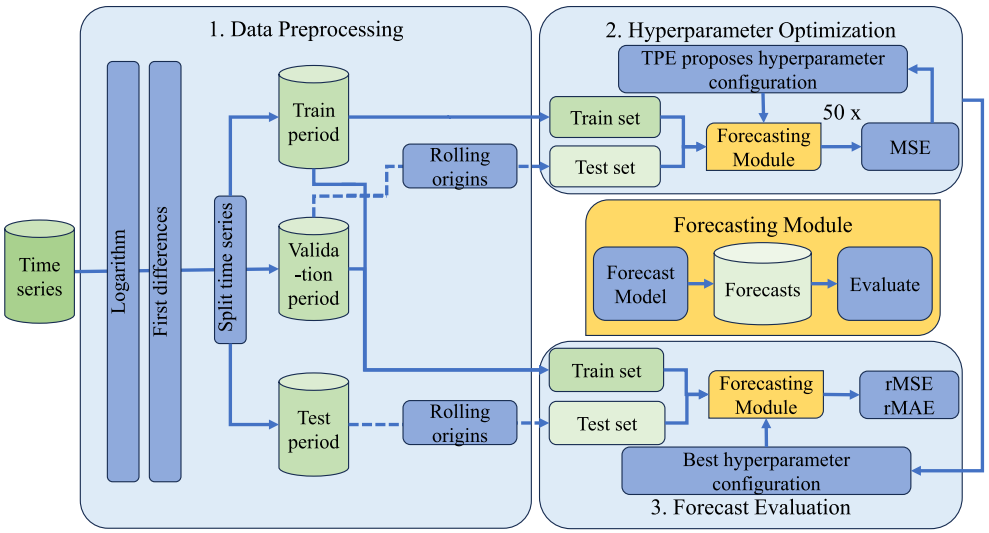


Fig. 2 Flow chart depicting the quasi-real-time forecasting setup

cessing, hyperparameter optimization, and forecast evaluation. Additionally, it illustrates how the training, validation, and test periods are utilized. Specifically, the validation period is employed twice: with rolling origins as a test set in the hyperparameter optimization, and in total as part of the final training period for the out-of-sample forecast evaluation. The following describes these steps in detail.

To ensure stationarity of all time series, we model the first logarithmic differences as percentages, calculated as

$$\Delta \mathbf{y}_{t+1} = 100 \cdot (\log(\mathbf{y}_{t+1}) - \log(\mathbf{y}_t)) .$$

We divide all datasets into a training period (first 70%), a validation period (next 10%), and a test period (last 20%).

During the validation period, predictions are made using XGBoost, DeepVAR, LSTM, TFT, and HCNN across 16 rolling origins. To obtain 16 rolling origins in the validation period, we use step sizes of 36, 4, and 240 between two rolling origins for exchange rates, NASDAQ100, and FRED-MD, respectively. We employ Tree-Structured Parzen Estimators (TPE) [6] for hyperparameter optimization to determine the optimal settings for each of the three applications. Table A1 in the appendix describes the search spaces in detail. It includes the most important hyperparameters of each model and allows values in the range of the default values. In particular, we let TPE evaluate 50 settings and search for the smallest MSE over all forecast steps. Appendix D analyzes the sensitivity of the models' forecast performance over the validation set to the hyperparameters. To reduce the computational burden during the hyperparameter search in the case of BVAR, we sample only from 2 chains and restrict the set of potential lag order p (exchange rates: $p \in \{5, 10\}$, NASDAQ100: $p \in \{10, 20\}$, FRED-MD: $p = 12$).

For the exchange rates dataset, the maximum forecast horizon is set at 20 working days—approximately one month. For NASDAQ100, we analyze half-hour-ahead forecasts, which correspond to 30-step-ahead forecasts for the minute-by-minute data. We focus on a maximum forecast horizon of six months for FRED-MD.

Forecast performance over the test period is assessed using two distinct methodologies. For the exchange rates and stock-price applications, we use non-overlapping, lined-up rolling origins. The first forecast origin is at the beginning of the test period, and we produce a forecast with the selected forecast horizon, which are 20 and 30, respectively, for the exchange rates and stock-price applications. Then we use the period after the last forecast period as the next forecast origin. We continue this way until we reach the end of the sample. Since the FRED-MD dataset is considerably shorter, we allow the rolling origins to overlap in the price indices application, moving the forecast window by only one period. Given that the stock market is open for only a few hours each day, the NASDAQ100 time series intervals are not equidistant. To avoid rolling origins that overlap from one day to another, we start with the opening of the stock market. A step size of 30 minutes precisely covers the stock market's opening hours ($390 \bmod 30 \equiv 0$). Following this approach, we achieve 61, 268, and 146 rolling origins for exchange rates, NASDAQ100, and FRED-MD, respectively.

During the test period, BVAR, XGBoost, LSTM, TFT, and HCNN, continue to utilize the hyperparameters optimized in the validation period. For ETS and ARIMA, we determine the most suitable lag order according to the Akaike Information Criterion (AIC) separately for each rolling origin.

4 Results

This section presents the benchmark results. First, the forecasting performance is depicted, followed by an analysis of the results.

4.1 Forecasting performance

To assess model competitiveness, [12, p.798] propose to use an error relative to a simple benchmark method. We calculate the MSE relative to the naive forecast (rMSE)

$$rMSE_\tau(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_r (e_{i,r,\tau}(\mathbf{y}, \hat{\mathbf{y}}))^2}{\sum_r (e^{naive}_{i,r,\tau}(\mathbf{y}))^2},$$

where $e_{i,r,\tau}(\mathbf{y}, \hat{\mathbf{y}}) = \mathbf{y}^i_{r+\tau} - \hat{\mathbf{y}}^i_{r+\tau}$ is the error of variable i in the τ-th forecast step of the rolling origin with start time r, and $e^{naive}_{i,r,\tau}(\mathbf{y}) = \mathbf{y}^i_{r+\tau}$ the corresponding error of the naive forecast. We first sum over the squared errors of the rolling origins to avoid division by zero, since the error of the naive forecast is zero for particular periods. Similarly, we calculate the MAE relative to the naive forecast (rMAE)

$$rMAE_\tau(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_r |e_{i,r,\tau}(\mathbf{y}, \hat{\mathbf{y}})|}{\sum_r |e^{naive}_{i,r,\tau}(\mathbf{y})|}.$$

A model outperforms the naive forecast with respect to the error if rMSE or rMAE are smaller than one. For the three datasets described in Section 3.1, we look at the rMSE and rMAE for different forecast horizons. Additionally, we evaluate the normalized root mean squared error (NRMSE) in Appendix B with similar results.

To statistically determine if the forecast errors significantly differ from those of the naive forecast, we employ the Diebold-Mariano test [9]. This test calculates the loss differential d for each variable i, rolling origin r, and forecast horizon τ

$$d_{i,r,\tau} = e_{i,r,\tau}(\mathbf{y}, \hat{\mathbf{y}})^2 - e^{naive}_{i,r,\tau}(\mathbf{y})^2$$

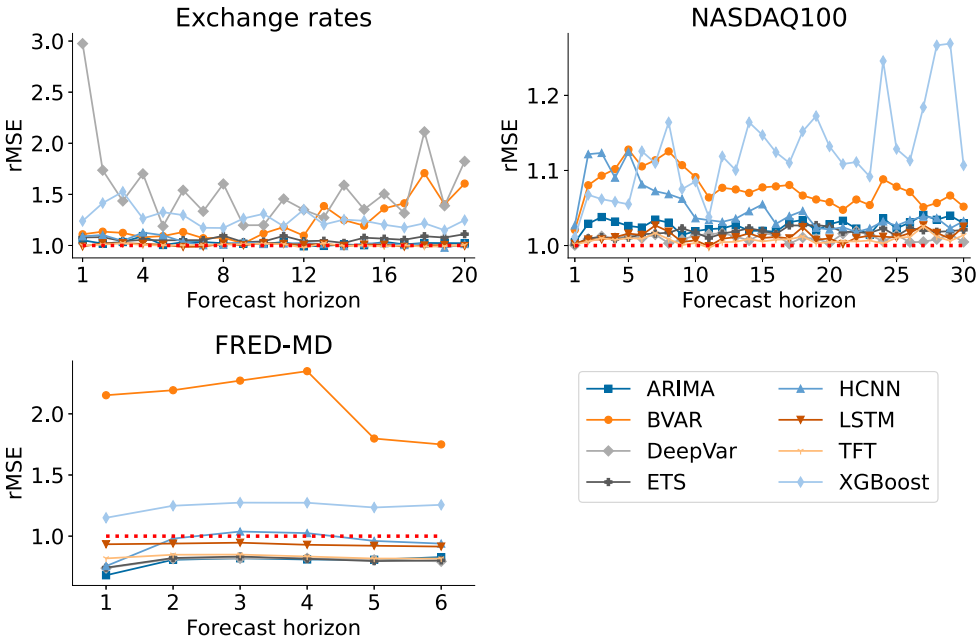
Afterwards the test statistic DM is calculated by

$$DM_{i,\tau} = \frac{\overline{d}_{i,\tau}}{\frac{1}{R}\sum_{r=1}^{R}(d_{i,r,\tau} - \overline{d}_{i,\tau})^2},$$

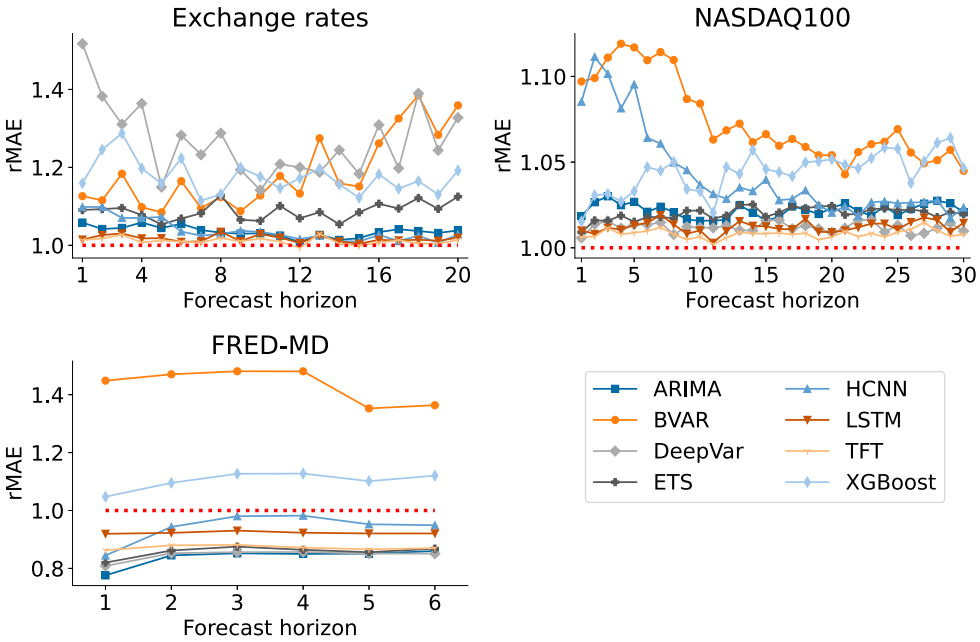
where $\overline{d}_{i,\tau} = \frac{1}{R}\sum_{r=1}^{R} d_{i,r,\tau}$ and R is the number of rolling origins. The test assumes that $DM_{i,\tau} \sim \mathcal{N}(0, 1)$ and the null hypothesis is $DM_{i,\tau} = 0$.

4.1.1 Exchange rates

Figure 3 shows the forecast performance across the models for 20 forecast steps, aggregating results over all exchange rates. LSTM, HCNN, TFT, ETS, and ARIMA exhibit rMSE and rMAE values that are slightly yet consistently above one. The errors of BVAR, DeepVAR, and XGBoost are consistently higher. The evaluation, based on only a modest number of observations, indicates that spikes at particular forecast



(a) rMSE



(b) rMAE

Fig. 3 Relative errors to the naive forecast depicted for each individual forecast horizon. The errors are calculated over all rolling origins and variables. Results for exchange rates, NASDAQ100 and FRED-MD are shown in for rMSE and rMAE in Figs. 3a), and b) respectively

horizons are primarily driven by outliers. Generally, there is no observed trend of increasing rMSE or rMAE with longer forecast horizons, a typical characteristic of unpredictable time series.

Given that rMSE in Fig. 3a represents an average across all variables, we utilize the Diebold-Mariano test to assess the predictability of individual time series at forecast steps 1, 4, 8, 12, 16, and 20. We conduct the test for each variable at forecast steps 1, 4, 8, 12, 16, and 20. Table 3 summarizes these results, with detailed findings available in Appendix E. The findings indicate that none of the model forecasts significantly outperform the naive baseline. Although certain variables exhibit improved forecasts with some benchmark models compared to the naive forecast, these improvements appear coincidental and lack statistical significance for all but two variables.

4.1.2 NASDAQ100

Analysis of rMSE and rMAE across individual forecast horizons in Figure 3 reveals a similar behavior for many models. The ARIMA, DeepVAR, ETS, HCNN, TFT, and LSTM models display errors very close to one, indicating performance comparable to the naive forecast. For certain models and specific forecast horizons, rMSE values marginally drop below one, potentially due to random variability. BVAR and XGBoost exhibit higher errors for almost all forecast steps.

Table 3 presents the results of the Diebold-Mariano test applied to the NASDAQ100 dataset. Across various forecast horizons and variables, no model consistently outperforms the naive forecast. Contrary to expectations for predictable models, the number of variables showing improved forecasts does not decrease with longer forecast horizons. Overall, only four out of 960 possibilities (calculated from 8 $models \times 6\ forecast\ horizons \times 20\ variables$) show a forecast significantly better than the naive forecast.

4.1.3 FRED-MD

In the third dataset of Fig. 3 BVAR and XGBoost consistently exhibit the highest relative errors. At the first forecast step, ARIMA, DeepVAR, ETS, TFT, and HCNN display similar errors. As the forecast horizon extends, both LSTM and HCNN show similar but higher errors compared to ARIMA, ETS, DeepVAR, and TFT. The latter maintain the lowest errors throughout all forecast horizons. Along with LSTM they consistently outperform the naive forecast. HCNN outperforms it on rMAE and additionally on rMSE for the forecast steps 1, 2, 5, and 6.

Table 3 Summary of Diebold-Mariano test for exchange rates, NASDAQ100 and FRED-MD dataset

Dataset	Forecast Horizon	Model							
		ARIMA	BVAR	DeepVar	ETS	HCNN	LSTM	TFT	XGBoost
ExchangeRate	1	2	1	1	1	1	5	3	0
	4	0	0	3	2	0	3	4	1
	8	3	1	2	3	2	2	4	1
	12	4	4	2	3	2	2	7 (2)	1
	16	1	2	2	2	2	3	6	0
	20	1	1	2	2	2	5	3	2
NASDAQ100	1	6 (1)	3	8 (1)	7	0	5	6	2
	2	2	0	4	5	0	3	5	3
	5	1	1	3	2	0	5	5	2
	10	5	0	5	2	2	8	6	5
	20	2	0	8	2	3	5 (1)	6 (1)	0
	30	3	1	11	3	2	4	6	1
FRED-MD	1	**17** (11)	2 (1)	**17** (9)	**17** (10)	16 (10)	9 (4)	12 (10)	5 (2)
	2	**15** (8)	2	14 (8)	14 (8)	9 (5)	9 (4)	11 (9)	3 (1)
	3	13 (8)	2 (1)	**14** (9)	13 (7)	7 (4)	9 (4)	11 (9)	3 (1)
	4	**15** (7)	3 (1)	13 (8)	**15** (9)	7 (3)	9 (4)	12 (10)	4 (1)
	5	**17** (8)	2 (1)	14 (8)	16 (9)	10 (4)	9 (5)	13 (10)	4 (1)
	6	13 (8)	2	**15** (9)	**15** (8)	9 (5)	9 (4)	13 (10)	4 (1)

The table shows the number of variables for which the respective model has smaller MSE than the naive forecast. Bold numbers indicate the benchmark models that outperform the naive forecast for the most variables on a forecast horizon

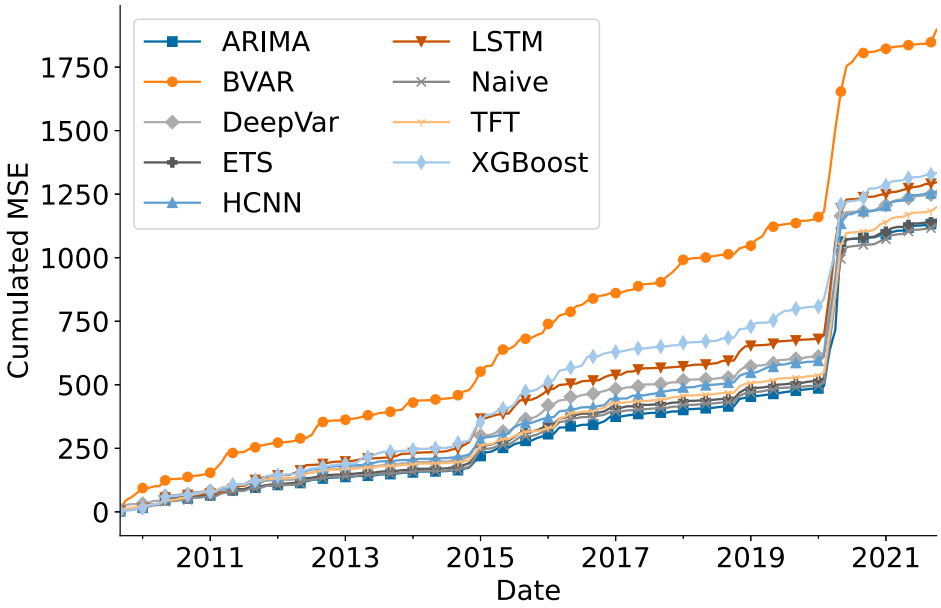


Fig. 4 Cumulative mean squared errors of the one-step forecast of the FRED-MD dataset for all models along the rolling origins

Results from the Diebold-Mariano test in Table 3 indicate that numerous variables exhibit a smaller MSE than the naive forecast across all forecast horizons. Specifically, ARIMA outperforms the naive forecast for 17 out of 20 variables, with 11 achieving statistical significance at the 5% level during the first forecast horizon.

Are there any specific episodes responsible for the forecast performance of the models? To investigate, we average the squared errors over all variables in the first forecast step for each rolling origin. Afterwards, we look at the cumulative sum of these errors over the test period in Fig. 4. The analysis reveals a faster increase in errors during the periods 2014-2016 and 2020. Notably, these errors are predominantly driven by the OILPRICEx variable, which exhibits significantly higher values compared to others. Between 2014 and 2016, the price of oil dropped rapidly, and began to stabilize in 2016. In 2020, the substantial price increases during March, April, May, and June were attributable to COVID-19 impacts. This led to higher volatility and correspondingly higher forecast errors. No method could accurately predict these periods and the naive forecast resulted in the smallest errors there.

Because the "consumer price index for all items" (variable "CPIAUCSL" in the FRED-MD dataset) is the most widely recognized price index and very relevant for the conduct of monetary policy, we take a closer look at the models' forecast performance for this particular time series. Figure 5 shows periods where the models have high forecast errors. However,

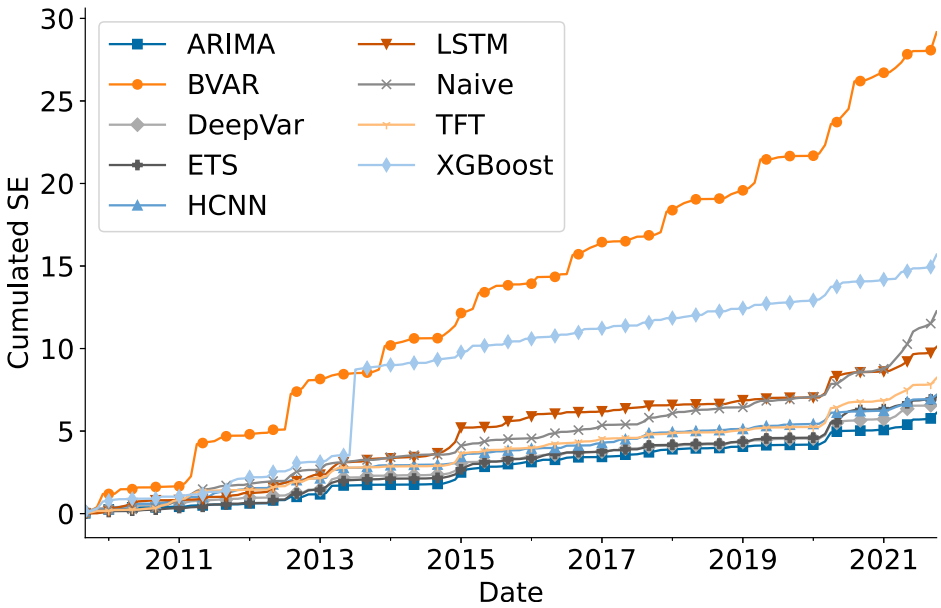


Fig. 5 Cumulative squared one-step "consumer price index for all items" forecast errors from the FRED-MD dataset for all models along the rolling origins

there are also periods that are predictable for some methods. In particular ARIMA, DeepVAR, HCNN, ETS, TFT, and LSTM are clearly able to outperform the naive forecast. This is important, for instance, for the conduct of forward looking monetary policy that aims at stabalizing consumer price inflation in the medium run. Furthermore, investors could use such informative price forecasts to generate trading signals in the foreign exchange market.

4.2 Generalization and influence of volatility on performance

This section examines the generalizability of results across rolling origins and evaluates the impact of volatility on the differential forecast performance of statistical versus machine learning methods.

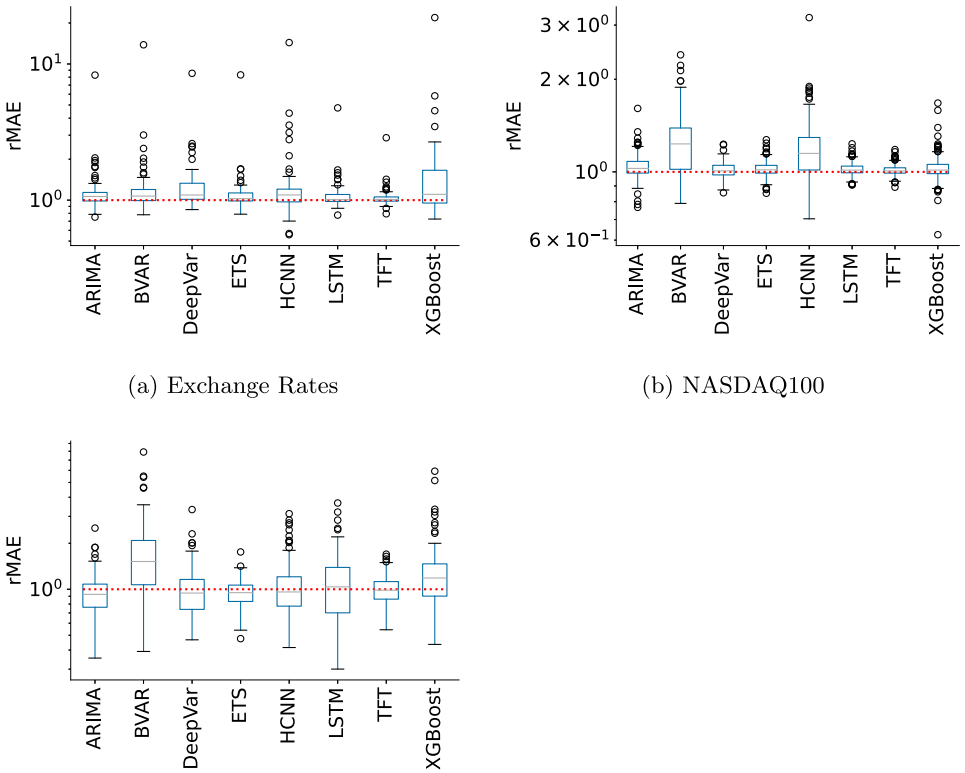
4.2.1 Error distribution over rolling origins

We aim to analyze the proportion of rolling origins where the naive forecast outperforms the benchmark models to determine if the results are mainly driven by a few unpredictable rolling origins. Accordingly, the rMAE is calculated as described in Section 4.1, but instead of aggregating the errors of the rolling origins, we aggregate the errors of the variables in the numerator and denominator of the fraction. We exclude rolling origins where the naive forecast error is zero from the analysis to avoid division by zero.

Figure 6 presents the rMAE distribution for the first forecast step across rolling origins for each model and specifically for the exchange rates, NASDAQ100, and FRED-MD datasets. In the first both datasets, the median rMAE consistently exceeds one, suggesting that the models mostly perform worse than guessing. Additionally, the 25% quantile reaches the threshold of one for most models. This indicates that, in the majority of cases, the variables are forecast less accurately than the naive forecast, and any improvements are small. This observation is underscored by the symmetry of the boxes on the logarithmic scale. These findings illustrate the challenge of consistently outperforming the naive forecast across various time series regimes. In the case of the FRED-MD dataset, the ARIMA, DeepVAR, ETS, TFT, and HCNN models outperform the naive forecast in the median, which demonstrates the predictability of the major part of regimes occurring in the dataset.

4.2.2 Relation of volatility to forecast performance

We aim to analyze the forecast performance difference between statistical models and machine learning models in relation to volatility. Specifically, at the first forecast step, we average the MAE for each variable and rolling origin across



(a) Exchange Rates

(b) NASDAQ100

(c) FRED-MD

Fig. 6 The distribution of rMAE with logarithmic scale for the rolling origins in the test set. Figure 6a) shows the exchange rates, Figure b) the NASDAQ100, and Figure c) the FRED-MD dataset

the statistical forecasts ($\hat{\mathbf{y}}_{stat} = \{\hat{\mathbf{y}}_{ETS}, \hat{\mathbf{y}}_{ARIMA}, \hat{\mathbf{y}}_{BVAR}\}$) and machine learning forecasts ($\hat{\mathbf{y}}_{ML} = \{\hat{\mathbf{y}}_{HCNN}, \hat{\mathbf{y}}_{LSTM}, \hat{\mathbf{y}}_{DeepVAR}, \hat{\mathbf{y}}_{XGBoost}, \hat{\mathbf{y}}_{TFT}\}$), where $\hat{\mathbf{y}}_{model}$ denotes the predictions from the respective models. The MAE for statistical models MAE^{stat} and machine learning models MAE^{ML} are computed respectively as

$$MAE^{stat}_{i,r} = \frac{1}{3} \sum_{\hat{\mathbf{y}} \in \hat{\mathbf{y}}_{stat}} |e_{i,r,1}(\mathbf{y}, \hat{\mathbf{y}})|,$$

$$MAE^{ML}_{i,r} = \frac{1}{5} \sum_{\hat{\mathbf{y}} \in \hat{\mathbf{y}}_{ML}} |e_{i,r,1}(\mathbf{y}, \hat{\mathbf{y}})|.$$

Afterwards, the differences between both errors are calculated as $\Delta MAE_{i,r} = MAE^{ML}_{i,r} - MAE^{stat}_{i,r}$.

The current volatility for each variable is calculated as the standard deviation of logarithmic returns over the preceding 20 periods for each rolling origin r.

We then look at the Pearson correlation between $\Delta MAE_{i,r}$ and the volatility for each variable, testing if this correlation is significantly greater than zero.

The *pearsonr* function in SciPy [26] assumes that the probability function is distributed like a Student's t-distribution with zero population correlation r

$$f(r) = \frac{\left(1 - r^2\right)^{n/2-2}}{\mathcal{B}\left(\frac{1}{2}, \frac{n}{2} - 1\right)},$$

with the Beta distribution $\mathcal{B}$. To fulfill the test assumptions of normal distributions the volatility is transformed with the logarithm beforehand.

In the analysis, 37.5% of variables in the exchange rates dataset and 65% in the NASDAQ100 dataset show a significant positive correlation at the 5% level. This suggests that in highly volatile datasets, such as exchange rates and NASDAQ100, the forecast performance of machine learning models deteriorates more in response to increased volatility. Given the minimal dependencies between variables in these datasets, machine learning models tend to overfit to apparent patterns, leading to higher errors on the test set. Conversely, in the FRED-MD dataset, only 5% of the variables exhibit a significant correlation. This dataset features stronger interdependencies among variables, which allow machine learning methods to better leverage their capacity to consider many variables.

5 Discussion and conclusion

Our paper demonstrates that complex forecasting methods fail to outperform the naive forecast in the case of two datasets that are commonly used to evaluate the performance of new machine learning approaches. This questions the practical value of other studies that compare new machine learning methods solely with complex models, which may overestimate the effectiveness of new machine learning techniques by not benchmarking them to simpler, cost-effective alternatives. In contrast, economic forecasting literature traditionally considers naive forecasts as a fundamental benchmark, underscoring a critical flaw in recent time series machine learning studies.

We assess the forecast performance of popular time series forecasting methods—ETS, ARIMA, BVAR, LSTM, HCNN, DeepVAR, TFT, and XGBoost—against the naive forecast using two frequently used datasets: exchange rates and NASDAQ100 prices. None of the models outperformed the naive forecast in either application, as confirmed by significance testing with Diebold-Mariano tests. This strengthens previous findings, highlighting the difficulty of improving upon the naive forecast in highly volatile datasets. In financial markets, certain time series are inherently unpredictable, causing complex methods to overfit the training data and misconstrue noise as signal. During forecasting, these models rely on incorrectly learned relationships and perform worse than random guessing. Simpler models, being less prone to overfitting, generally perform better on these time series. Our correlation analysis of a time series' volatility to the error differences of machine learning and statistical methods supports this observation. The closer a time series approximates a random walk, the more appropriate it is to consider simpler models for forecasting, thus diminishing the usefulness of complex machine learning models.

The datasets used in this paper are, of course, only exemplary, but many similar datasets exist where this problem arises. Frequently, new models are proposed and benchmarked on these datasets without comparison to the naive forecast, leading to potentially incorrect conclusions about their effectiveness.

We propose assessing model performance using more predictable datasets, such as the macroeconomic price indices from the FRED-MD database. Our analysis reveals that this dataset contains a mix of predictable and challenging variables. Overall, complex forecasting models outperform the naive forecast, demonstrating their practical value. The HCNN model performs comparably to the best forecasting models within the FRED-MD dataset.

Utilizing a metric that incorporates a naive benchmark proves beneficial. It enables the comparison of complex models' relative forecast performance and allows for the assessment of the absolute value each machine learning approach offers in practice.

Forecasting tasks differs widely in terms of the properties of the data to be forecast. It is no surprise that no method performs best for all data. Out study shows that in situations with time series that closely resemble random walks (or white noise in first differences), such as the exchange

rates and NASDAQ100 datasets, it is impossible for linear statistical models and even for machine learning models to outperform naive no-change predictions. This is especially true when limited training data is available because then training methods lack power to detect weak systematic patterns in the data. In that sense, our main finding applies primarily to highly volatile data without strong intertemporal dependencies. Linear statistical models tend to perform better for autocorrelated time series and in cases where non-linear dependencies are a minor feature of time series. In contrast, machine learning models can perform well if they are able to find complex dependencies without overfitting to the training dataset. This requires time series with longer histories, especially if these dependencies are rather weak. In multivariate time series use cases with better data availability and stronger dependencies between variables, machine learning methods can leverage their capabilities, notably, for instance, in areas such as demand forecasting.

Appendix A Model parameters

Table 4 Detailed description of the search spaces for the hyperparameter optimization

Model	Parameter	Hyperparameter optimization Search Space	Test Exchange rates	NASDAQ 100	FRED-MD
BVAR	lag	Exchange rate: choice(5, 10) NASDAQ100: choice(10, 20) FRED-MD: choice(12)	5	10	12
HCNN	lag	$range(ph_{min}, ph_{max}, step = 6)$	122	180	108
	epochs	$range(5,000, 40,000, step = 100)$	26,500	40,000	40,000
	num cells	$range(40, 120, step = 5)$	40	110	45
	decrease teacher forcing	$interval(0, 2.0e^{-5})$	$2.0e^{-5}$	$1.6e^{-5}$	$1.0e^{-5}$
	sparsity	$interval(0, 0.8)$	0.17	0.08	0.57
DeepVAR	lag	$range(ph_{min}, ph_{max}, step = 6)$	92	174	12
	epochs	$range(25, 500, step = 5)$	430	410	495
	num batches per epoch	$range(25, 75, step = 5)$	40	60	55
	num layers	$range(1, 2, step = 1)$	1	1	1
	num cells	$range(40, 140, step = 5)$	110	70	40
	dropout	$interval(0, 0.5)$	0.13	0.27	0.48
	batch size	$range(8, 64, step = 8)$	24	32	56
XGBoost	lag	$range(ph_{min}, ph_{max}, step = 6)$	20	30	66
	min bin size	$range(10, 100, step = 10)$	70	100	20
	max n datapts	$range(5e^5, 1.5e^6, step = 1e^5)$	$1.4e^6$	$1.4e^6$	$1.1e^6$
	use feat dynamic real	choice(True, False)	False	False	False
TFT	lag	$range(ph_{min}, ph_{max}, step = 6)$	104	90	30
	epochs	$range(25, 500, step = 5)$	130	110	150
	num batches per epoch	$range(25, 75, step = 5)$	55	30	40
	num heads	$range(1, 6, step = 1)$	2	5	5
	num cells	$range(40, 140, step = 5)$	140	105	120
	dropout	$interval(0, 0.5)$	0.36	0.27	0.41
	batch size	$range(8, 64, step = 8)$	32	40	8
LSTM	lag	$range(ph_{min}, ph_{max}, step = 6)$	80	96	102
	epochs	$range(25, 500, step = 5)$	375	275	25
	num batches per epoch	$range(25, 75, step = 5)$	55	60	25
	num layers	$range(1, 2, step = 1)$	2	1	1
	num cells	$range(40, 140, step = 5)$	60	100	80
	dropout	$interval(0, 0.5)$	0.32	0.43	0.08
	batch size	$range(8, 64, step = 8)$	32	40	16

The parameters ph_{min} are dataset specific and equal to the forecast horizon. Also, ph_{max} depends on the dataset and is 140, 180, 140 for exchange rates, NASDAQ100 and FRED-MD respectively. For hyperparameter optimization, *choice* indicates that the algorithm can select between the given values. The option $range(start, end, step)$ indicates that all integers $start + i \cdot step < end$ for $i = 0, 1, \ldots$ are valid choices and $interval(start, end)$ allows all values between *start* and *end*

Appendix B Further error metrics

As an additional error measure we employ the normalized root mean squared error (NRMSE). For each variable the root mean squared error of a model is normalized by the standard deviation of the data points used to evaluate the respective forecast horizon. Precisely, for forecast horizon τ we include the τ-th forecast step of each rolling origin. We then, average the NRMSE over the variables

$$NRMSE_\tau(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\sqrt{\frac{1}{R}\sum_r (e_{i,r,\tau}(\mathbf{y}, \hat{\mathbf{y}}))^2}}{std(\hat{\mathbf{y}}^i_{r_1+\tau}, \ldots, \hat{\mathbf{y}}^i_{r_R+\tau})}.$$

Similar to Section 4.1, Fig. 7 shows that the naive forecast can not be outperformed by statistical or machine learning models for exchange rates and NASDAQ100 stocks. For the FRED-MD dataset the naive forecast can only be beat by ARIMA and ETS.

Appendix C Model execution efficiency

To evaluate the model execution efficiency, [30] propose to represent the error of a model as the angle in a pie chart and the execution time as the radius of the corresponding wedge. The area of the wedge then represents the model execution efficiency. If the area is small, the model either has good performance, is fast to compute, or both. As the error, we choose the MAE as described in Section 4.1 of the first forecast horizon. The angle is the proportion of the error in the aggregated error of all models times 360 degrees.

The radius of a wedge associated with the model is the execution time of the model to be fitted and for the prediction of one rolling origin for all forecast horizons 20, 30, and 6 respectively for exchange rates, NASDAQ100, and FRED-MD. We measure the execution time for fitting and predicting one rolling origin in seconds. The radius is then calculated by adding one and taking the logarithm, as we have very short durations (naive forecast) and very long durations (HCNN). Note that the execution times are highly dependent on the model hyperparameters determined as described in Section 3.

For DeepVAR, LSTM, XGBoost, and TFT, most of the time is used to fit the models. When they are fitted, inference for other rolling origins is much faster (less than a second). The results show that the naive forecast is the most efficient method since almost no computation is necessary. For the three datasets additional computation time shows no improvements. In particular, for exchange rates and NASDAQ100, the execution time exerts a dominant influence on the execution efficiency, since the errors are similar. For FRED-MD, ARIMA and ETS have the best execution efficiency, since they have the smallest errors and are fast to compute. Overall, the statistical models exhibit a better execution efficiency, compared to the machine learning models.

Appendix D Parameter sensitivity

For those models where we perform a hyperparameter optimization, we analyze how the performance changes when the parameters are changed. Therefore, we look at the hyperparameter optimization of the FRED-MD dataset and analyze the MSE of the first forecast horizon on the validation set.

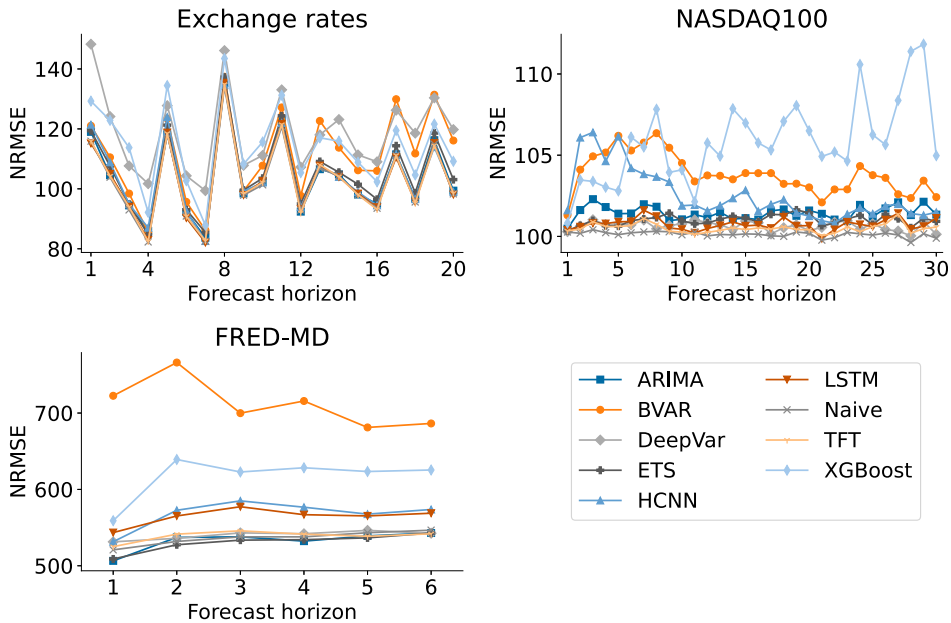


Fig. 7 NRMSE depicted for each individual forecast horizon with the errors calculated over all rolling origins and variables. Results for exchange rates, NASDAQ100 and FRED-MD are shown

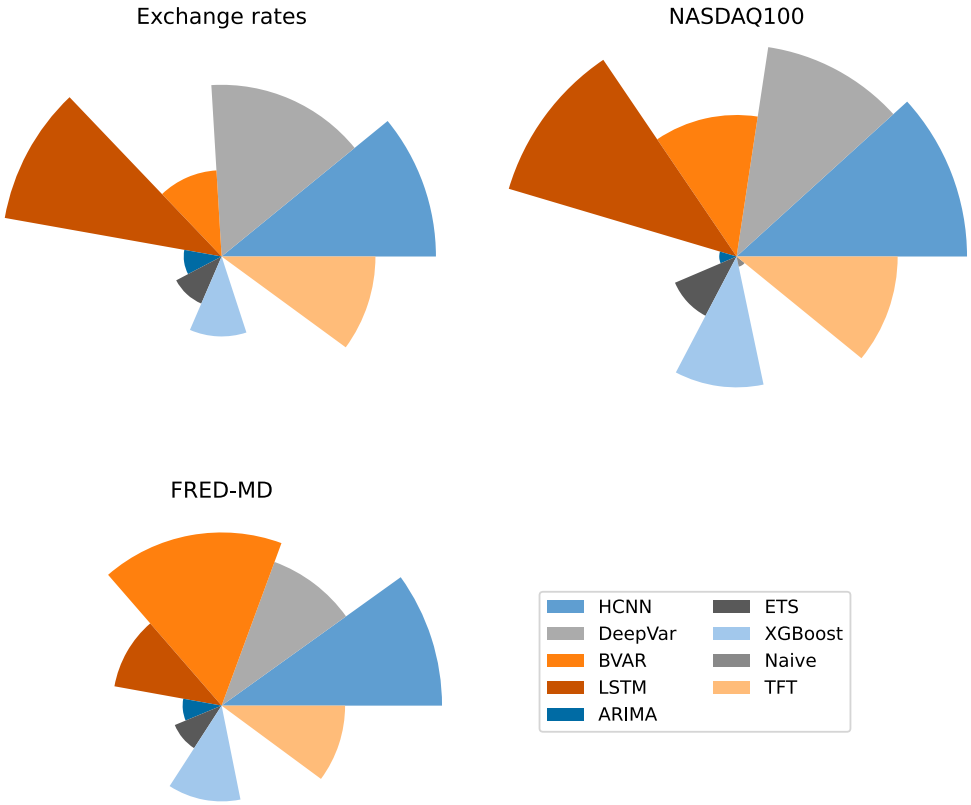


Fig. 8 Comparison of the model execution efficiency of the benchmark models for exchange rates, NASDAQ100, and FRED-MD. The angle of a model represents the error and the radius of the corresponding wedge represents the execution time

In addition, we plot a linear least squares regression of the errors over the trials as a trend line.

In Fig. 9 we see that all models are sensitive to their hyperparameters. However, the range in which the models have errors is quite different. HCNN and TFT have the highest divergence between the best and worst parameter set with 4.4 and 4.3 respectively, while XGBoost has the clearly the lowest divergence with 1.4.

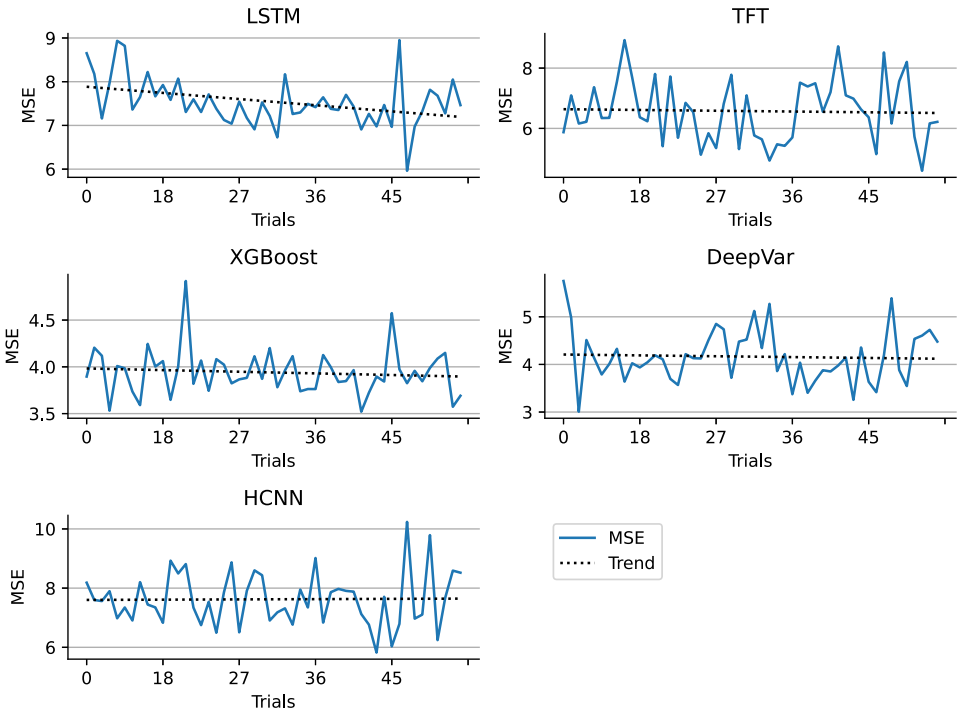


Fig. 9 Analysis of the MSE of the first forecast horizon during the hyperparameter optimization for the FRED-MD dataset. Shown are the models for which we have optimized hyperparameters. The trend shows whether better parameters are found for more trials

In addition, by analyzing the trend, we can see that TPE performs differently in finding parameter patterns with better performance. On the one hand, TFT, DeepVAR, and HCNN have almost no visible trend. On the other hand, XGBoost has a slightly negative trend and LSTM has a strong negative trend. However, LSTM performs worst in the beginning, so there is room for improvement.

Appendix E Diebold Mariano test

Table 5 Diebold-Mariano test for exchange rates

Model	Variable	Forecast Horizon					
		1	4	8	12	16	20
ARIMA	0	1.14**	1.04	1.04	1.0	1.0	1.04
	1	1.01	1.13*	0.98	1.04	1.04	1.01
	2	1.04	1.05	0.97	0.96	1.04	1.0
	3	1.12	1.15	1.0	1.02	1.01	0.99
	4	0.99	1.0	1.08*	0.97	1.08	1.04
	5	1.0	1.11**	1.06**	1.0	1.1*	1.05
	6	1.13*	1.08	1.02	0.99	0.97	1.01
	7	0.98	1.18	1.03	0.95	1.04	1.04
BVAR	0	1.15**	1.06	1.0	0.95	0.95	1.01
	1	1.06	1.08	1.03	0.99	0.99	1.02
	2	1.01	1.04	1.04	1.06	1.1	1.09*
	3	0.99	1.02	1.04	1.1	1.05	1.04
	4	1.44*	1.08	1.43***	1.7***	3.47***	5.48***
	5	1.07	1.1	1.01	0.96	1.15**	0.98
	6	1.09	1.11	1.0	1.05	1.01	1.06
	7	1.07	1.16**	1.06	0.96	1.18	1.16
DeepVar	0	1.06	0.97	1.02*	1.02	1.02	1.01
	1	1.03	0.9	0.94	1.15*	1.04	1.02
	2	1.01	1.04	1.06	1.07	0.98	1.17***
	3	0.99	1.05	1.05	1.17*	1.03	0.98
	4	16.66***	6.53***	5.58***	3.42***	4.78***	7.29***
	5	1.02	0.98	1.14**	1.01	1.18***	1.0
	6	1.02	1.03	1.03	0.99	1.02	1.09**
	7	1.0	1.11***	1.0	0.97	0.99	1.03
ETS	0	0.99	1.0	0.99	1.0	1.03	0.99
	1	1.05	1.04	1.04	0.99	1.03	1.04
	2	1.03	1.0	1.03	1.07**	0.91*	1.04
	3	1.01	1.05	1.06***	1.06	0.98	1.0
	4	1.42***	1.27***	1.67***	1.23**	1.47***	1.63***
	5	1.02	1.02	0.98	1.03	1.08**	1.07
	6	1.05	1.03	0.99	0.97	1.02	1.08**
	7	1.05	1.05	1.01	1.0	1.0	1.05
HCNN	0	1.21**	1.1	1.02	1.1	1.04	1.03
	1	1.09	1.1	1.01	1.06	1.06	1.04
	2	1.06	1.25*	0.97	1.09	1.06	1.0
	3	1.24**	1.05	1.04	1.02	0.99	1.02

Table 5 continued

Model	Variable	Forecast Horizon					
		1	4	8	12	16	20
	4	0.85	1.01	1.01	0.95	1.02	1.01
	5	1.08	1.17***	0.99	0.91	1.02	1.03
	6	1.12	1.16*	1.01	1.05	1.02	0.99
	7	1.05	1.18	1.01	1.03	0.95	0.99
LSTM	0	1.0	0.95	1.02	1.03	1.03	1.01
	1	0.97	1.06	1.02	1.01	1.02	1.02
	2	0.96*	0.99	1.02	1.02	1.02	0.99
	3	0.97	1.04	0.99	1.06	0.97	0.99
	4	1.05	1.01	1.03	1.01	1.04	0.96
	5	1.0	1.06**	0.99	0.95	1.0	1.02
	6	0.99	1.01	1.04***	1.01	1.0	0.98
	7	0.98	1.0	1.04**	0.97	0.98	1.0
TFT	0	0.98	1.0	1.01	1.01	0.99	1.02
	1	0.98	1.05	1.03	1.0	1.0	1.01
	2	0.99	1.01	1.03	0.98	0.99	0.99
	3	1.01	0.99	1.0	0.96	0.99	1.0
	4	1.01	1.0	1.0	0.98	1.02	0.99
	5	1.01	1.01	1.0	0.94**	1.0	1.02
	6	1.01	1.0	1.02	1.0	0.97	0.99
	7	1.02	0.97	1.0	0.95**	0.99	1.01
XGBoost	0	1.36**	1.69**	0.79	1.1	1.28*	1.27**
	1	1.16	1.5**	1.28*	1.25	1.13	0.93
	2	1.03	1.21	1.24**	1.48**	1.41***	1.25
	3	1.34**	1.35**	1.74	1.65**	1.21	0.87
	4	1.1	0.98	1.17	0.96	1.03	1.09
	5	1.46**	1.11	1.09	2.13*	1.08	1.31**
	6	1.31	1.03	1.02	1.13	1.46**	1.52***
	7	1.15	1.25**	1.04	1.11	1.01	1.74

Values indicate MSE relative to the naive forecast. The stars mark different significance levels: * 10%, ** 5% and *** 1%

Table 6 Diebold-Mariano test for NASDAQ100

Model	Variable	Forecast Horizon					
		1	2	5	10	20	30
ARIMA	AAL	1.01*	1.02	1.03**	1.04	1.0	1.02
	AAPL	1.0	1.05***	1.03	0.98	1.03	1.0
	ADBE	1.01**	1.01	1.01	1.03	1.01	1.0
	ADI	1.01	1.06***	1.02	1.05*	1.06*	1.04
	ADP	1.01**	1.01	1.04*	1.03	1.04	0.99
	ADSK	1.04	1.0	1.03	1.03	1.05	1.05
	AKAM	1.01***	1.03*	1.02	1.05	1.02	0.98
	ALXN	1.0*	1.09	1.08**	1.02	1.07**	1.09**
	AMAT	1.0	1.12***	1.0	1.0	1.01	1.07**
	AMGN	1.01	1.02	1.03***	0.99	1.03	1.02
	AMZN	1.01	0.96	1.01	1.03	1.03	1.0
	ATVI	1.0	1.03*	1.02	1.03*	1.0	1.03*
	AVGO	1.0**	1.03	1.0	1.01	1.03	1.03
	BBBY	1.0	1.03**	1.01	0.97	1.03*	1.05
	BIDU	1.01	1.03	1.06**	1.03	0.98	1.03*
	BIIB	0.99	1.03	1.02	1.05*	1.02	1.02
	CA	1.0	1.0	1.04*	1.03	1.07	1.07***
	CELG	0.99	1.02	1.02	0.99	1.01	1.08***
	CERN	1.01*	1.03	1.01	1.01	1.03	1.02
	CMCSA	0.97**	1.01	1.04	1.02	1.03*	1.01
BVAR	AAL	1.0	1.09*	1.18***	1.06**	1.08**	1.04
	AAPL	1.06**	1.1**	1.23***	1.08**	1.01	1.0
	ADBE	1.03*	1.05	1.04	1.08**	1.01	1.03*
	ADI	1.05**	1.22***	1.48***	1.06	1.17***	1.11***
	ADP	0.99	1.05*	1.07**	1.0	1.01	1.06***
	ADSK	1.06**	1.01	1.01	1.05	1.06*	1.08***
	AKAM	1.02***	1.05**	1.0	1.04	1.01	1.03
	ALXN	1.01	1.14**	1.09**	1.09***	1.14***	1.16***
	AMAT	1.0	1.07*	1.05	1.07**	1.02	1.01
	AMGN	1.03***	1.05*	1.06*	1.07*	1.02	1.04*
	AMZN	1.05**	1.08**	1.17***	1.07	1.07**	1.05
	ATVI	1.01	1.14**	1.18***	1.09***	1.08***	1.03
	AVGO	1.01	1.05	1.09**	1.09***	1.03	1.01
	BBBY	1.01	1.04*	1.25***	1.12***	1.04	1.05
	BIDU	1.03	1.12**	1.09**	1.21***	1.05*	1.03
	BIIB	1.04**	1.07*	1.28***	1.33***	1.15***	1.17***
	CA	1.0	1.11***	1.08**	1.03	1.05**	1.0
	CELG	1.01	1.04**	1.08**	1.08***	1.01	1.01
	CERN	1.0	1.02	1.04*	1.09**	1.1***	1.11***
	CMCSA	1.02	1.09***	1.09*	1.12***	1.05**	1.01
DeepVar	AAL	1.0*	0.99	1.01	0.99	1.01	1.02
	AAPL	1.01*	1.01	1.02	1.03**	1.03**	1.01
	ADBE	1.0	1.01	1.03***	1.01	0.99	1.03**

Table 6 continued

Model	Variable	Forecast Horizon					
		1	2	5	10	20	30
	ADI	1.01	1.02	0.99	1.02	0.99	0.99
	ADP	1.01	1.0	1.02***	1.02*	1.0	1.0
	ADSK	1.01	1.01	1.01	1.02	0.98	1.01
	AKAM	0.99	1.01	1.0	1.01	1.0	1.02
	ALXN	1.0	1.02	1.0	0.99	0.99	1.0
	AMAT	1.01**	1.0	1.0	1.01	1.01	0.99
	AMGN	1.0	1.03	1.01	1.01	1.02	0.99
	AMZN	1.0	0.99	1.0	1.03*	0.99	1.02
	ATVI	0.99**	1.03**	0.99	1.03*	1.01	1.02
	AVGO	1.0	1.0	1.04**	1.0	1.0	1.03
	BBBY	1.0	1.02**	1.05*	1.0	0.99	0.99
	BIDU	1.0	1.01	1.01	1.03*	0.99	1.0
	BIIB	1.0	1.02	1.02	1.04*	1.01	0.99
	CA	1.0	1.0	1.01	1.0	1.0	0.99
	CELG	1.0	1.0	1.0	1.01	1.02	1.0
	CERN	1.0	1.0	1.01	1.01	1.01	0.99
	CMCSA	0.99	1.0	1.01	1.01	1.01	1.01
ETS	AAL	1.01	0.99	0.99	1.04***	1.0	1.07***
	AAPL	1.01	1.02	1.02	1.02	1.06***	0.99
	ADBE	0.99	1.01	1.01	1.03**	1.03**	1.05***
	ADI	1.01*	1.0	1.01	0.99	1.02	1.02
	ADP	1.01	1.0	1.0	1.01	1.01	1.02
	ADSK	1.0	1.01	1.01	0.99	1.02	1.0
	AKAM	1.01	1.02*	1.05***	1.0	1.04	1.02
	ALXN	1.0	1.01	1.02	1.03*	0.99	0.99
	AMAT	1.0	1.0	1.02*	1.02*	1.02	1.03*
	AMGN	1.0	1.02	1.01	1.02	1.07***	1.05**
	AMZN	1.0	1.02	1.0	1.01	1.03**	1.02
	ATVI	0.99	1.02	0.98	1.01	1.03**	1.0
	AVGO	1.0	1.01	1.01	1.01	1.01	1.02
	BBBY	1.0	0.99	1.02	1.0	1.0	1.03**
	BIDU	1.0	1.02**	1.02	1.01	1.04**	1.01
	BIIB	1.0	1.02**	1.01	1.03*	1.05***	1.01
	CA	1.0	1.02*	1.0	1.01	1.02*	1.02*
	CELG	1.01	1.01	1.0	1.03*	1.02	1.01
	CERN	1.0	1.0	1.0	1.04**	1.03	1.03
	CMCSA	1.0	1.01	1.01	1.04**	1.04**	1.02**
HCNN	AAL	1.02**	1.11**	1.04	1.09***	1.01	1.03
	AAPL	1.06***	1.11***	1.12***	1.04	1.01	1.01
	ADBE	1.02*	1.09**	1.13***	1.0	1.01	0.97
	ADI	1.04*	1.18***	1.17***	1.04	1.02	1.03
	ADP	1.05***	1.06**	1.13**	1.01	1.06*	0.99
	ADSK	1.02	1.1***	1.22***	1.0	1.02	1.03

Table 6 continued

Model	Variable	Forecast Horizon					
		1	2	5	10	20	30
	AKAM	1.03**	1.07***	1.15***	1.05	1.03	1.01
	ALXN	1.01**	1.24**	1.16***	1.04	1.02	1.03
	AMAT	1.01	1.18***	1.1*	1.02	0.97	1.02
	AMGN	1.03**	1.12**	1.09***	1.03	1.04**	1.08***
	AMZN	1.02	1.11*	1.18***	1.05	1.01	1.05***
	ATVI	1.0	1.23***	1.17***	1.02	1.02	1.02
	AVGO	1.0	1.17***	1.11***	0.99	1.0	1.09**
	BBBY	1.02**	1.12***	1.12***	1.01	0.99	1.04
	BIDU	1.03**	1.11***	1.14***	1.02	1.06***	1.02
	BIIB	1.03***	1.1	1.18***	1.13**	1.01	1.06*
	CA	1.06***	1.07*	1.07**	1.04	1.02	1.01
	CELG	1.02*	1.12***	1.06	1.05**	1.04**	1.08**
	CERN	1.03***	1.11**	1.08***	1.05*	1.09**	1.07**
	CMCSA	1.02	1.04	1.08**	1.04	1.0	1.02
LSTM	AAL	1.0	1.03**	0.99	1.01	0.97	1.05***
	AAPL	1.01**	1.02**	1.03**	1.0	1.01	0.99
	ADBE	1.0	1.01	1.02	0.99	0.97**	1.01
	ADI	1.0	1.02	0.98	1.01	1.02	1.01
	ADP	1.0	1.0	1.04**	1.02	1.03	0.99
	ADSK	1.01	0.99	1.04**	0.98	1.0	1.04
	AKAM	1.0	1.0	1.02	1.03	1.01	1.01
	ALXN	1.0	0.97	1.02	1.03	1.04	1.09*
	AMAT	1.0	1.05***	1.0	1.0	1.0	1.04
	AMGN	1.0	1.0	1.02	0.99	1.0	1.04**
	AMZN	1.01*	1.01	1.04**	1.01	1.01	1.0
	ATVI	1.0	1.03***	1.02	0.98	1.01	1.03
	AVGO	1.0*	1.02	0.99	1.01	1.01	1.03
	BBBY	1.0	1.01	1.04	1.03	1.01	1.04
	BIDU	1.0*	1.0	1.03*	1.0	1.02*	1.01
	BIIB	1.0	1.01	1.02	1.04	0.98	1.02
	CA	1.0	0.98	1.02	1.0	1.02	1.02
	CELG	1.01**	1.0	0.99	1.0	1.0	1.04*
	CERN	1.0*	1.0	1.01	1.02	1.05	1.04
	CMCSA	1.01**	1.0	1.03**	0.99	1.01	1.0
TFT	AAL	1.0***	1.01*	0.99	1.0	0.99	1.02*
	AAPL	1.0	1.02**	1.02*	1.0	1.01	0.99
	ADBE	1.0	1.01*	1.02	1.0	0.97**	1.01
	ADI	1.0	1.01	0.99	1.04	0.99	1.0
	ADP	1.0	1.01	1.01**	1.01	1.01	1.0
	ADSK	1.0	0.99*	1.03*	0.97	0.99	1.03
	AKAM	1.0	1.0	1.01	1.0	1.01	1.0
	ALXN	1.0	0.98	1.0	1.01	1.01	1.04*
	AMAT	1.0	1.03*	1.01	0.99	1.01	1.04*

Table 6 continued

Model	Variable	Forecast Horizon					
		1	2	5	10	20	30
	AMGN	1.0	1.0	1.01	0.99	1.01	1.02
	AMZN	1.0	1.0	1.02*	1.01	1.01	1.0
	ATVI	1.0	1.03***	1.02**	0.99	1.0	1.03
	AVGO	1.0	1.01	0.99	1.01	1.02	1.01
	BBBY	1.0	1.0	1.01	1.01	1.01	1.02
	BIDU	1.0	1.01	1.01	1.01	1.01	1.0
	BIIB	1.0	1.0	1.02**	1.01	0.99	0.99
	CA	1.0	0.99	1.01	1.01	0.99	1.02
	CELG	1.0*	1.01	1.0	1.01	1.02	1.02*
	CERN	1.0**	1.0	1.01*	1.02	1.02	1.03
	CMCSA	1.0	1.0	1.02*	0.99	1.02	1.0
XGBoost	AAL	1.0	1.18	0.99	1.04	1.08**	1.11**
	AAPL	1.01*	1.24	1.03	0.99	1.04	1.05
	ADBE	1.03	1.02	1.08	1.06	1.13	1.07
	ADI	1.01*	1.02*	1.15	1.07*	1.15	1.11**
	ADP	1.02	1.09	1.07	0.99	1.21	1.04
	ADSK	0.99	1.0	1.08*	0.95	1.08	1.26***
	AKAM	1.0	1.07	1.05	1.03	1.13**	1.06
	ALXN	1.01	1.09***	1.07*	1.06	1.11	1.09*
	AMAT	1.01	1.12	1.03	1.07**	1.05	1.04
	AMGN	1.0	1.06*	1.06*	1.04	1.09	1.03
	AMZN	1.07	1.02	1.07*	0.99	1.07	1.04
	ATVI	1.0	1.03	1.01	1.16***	1.07**	1.06
	AVGO	1.0	1.03	1.14**	1.12	1.05	1.51**
	BBBY	1.0	1.18	1.13	1.19***	1.11*	1.13*
	BIDU	1.01	1.14	1.04	1.0	1.51	1.03
	BIIB	1.03	1.04	0.99	1.63	1.2**	1.11
	CA	1.01	1.0	1.06	1.05	1.15	1.17
	CELG	1.01*	1.06*	1.01	1.07	1.06	1.14**
	CERN	1.01	0.99	1.0	1.1	1.16	1.08
	CMCSA	0.99	1.0	1.02	1.09***	1.17	1.0

The stars mark different significance levels: * 10%, ** 5% and *** 1%

Table 7 Diebold-Mariano test for FRED-MD

Model	Variable	Forecast Horizon					
		1	2	3	4	5	6
ARIMA	CPIAPPSL	1.03	1.03	1.02	1.03	0.99	1.03
	CPIAUCSL	0.5***	0.69***	0.73***	0.71***	0.69***	0.7***
	CPIMEDSL	0.37***	0.37***	0.38***	0.4***	0.39***	0.39***
	CPITRNSL	0.78**	1.0	1.02	1.01	1.05	1.04
	CPIULFSL	0.52***	0.7***	0.73***	0.73***	0.73***	0.73***
	CUSR0000SA0L2	0.61***	0.86*	0.84**	0.87*	0.85**	0.86**
	CUSR0000SA0L5	0.49***	0.7***	0.74***	0.71***	0.71***	0.72***
	CUSR0000SAC	0.78**	1.0	1.03	0.96	0.97	0.98
	CUSR0000SAD	0.63*	0.87	0.95	0.87	0.91	1.04
	CUSR0000SAS	0.18***	0.22***	0.22***	0.22***	0.22***	0.22***
	DDURRG3M086SBEA	0.8*	0.95	0.89	0.88	0.9	1.08
	DNDGRG3M086SBEA	0.87*	1.06	1.04	1.04	1.02	0.99
	DSERRG3M086SBEA	0.21***	0.2***	0.2***	0.21***	0.19***	0.21***
	OILPRICEx	1.02	1.09	1.03**	1.0	1.0	1.02
	PCEPI	0.46***	0.6***	0.61***	0.62***	0.58***	0.6***
	PPICMM	0.96	1.11**	1.06	0.98	1.0	1.04
	WPSFD49207	0.86	0.9*	0.94	0.94	0.93	0.94
	WPSFD49502	0.93	0.92	0.96	0.96	0.94	0.94
	WPSID61	0.57***	0.79**	0.9	0.95	0.96	0.97
	WPSID62	1.04	1.09	1.1	1.09*	1.12**	1.08
BVAR	CPIAPPSL	2.12***	1.72***	1.69***	1.55***	1.56***	1.54***
	CPIAUCSL	2.38***	2.54***	2.89***	3.13***	1.95***	2.02***
	CPIMEDSL	0.89	0.86	0.84	0.86	0.89	0.87
	CPITRNSL	2.79***	2.52***	2.96***	3.2***	1.96***	1.8***
	CPIULFSL	2.02***	2.18***	2.3***	2.24***	1.34**	1.49***
	CUSR0000SA0L2	3.0***	3.05***	3.54***	3.77***	2.71***	2.39***
	CUSR0000SA0L5	2.4***	2.78***	2.96***	3.11***	2.12***	2.1***
	CUSR0000SAC	3.07***	2.86***	3.41***	3.93***	2.58***	2.33***
	CUSR0000SAD	1.27**	1.26**	1.39***	1.47***	1.2**	1.36***
	CUSR0000SAS	1.43***	1.42***	1.44***	0.97	1.04	1.01
	DDURRG3M086SBEA	1.24*	1.27**	1.4***	1.34***	1.39***	1.41***
	DNDGRG3M086SBEA	4.07***	3.62***	4.52***	4.77***	3.17***	2.8***
	DSERRG3M086SBEA	0.8**	0.94	0.76***	0.75***	0.73***	0.96
	OILPRICEx	1.5***	1.63***	1.26***	1.37***	1.13**	1.13***
	PCEPI	2.13***	2.37***	2.63***	2.66***	1.9***	1.94***
	PPICMM	3.21***	3.39***	2.84***	2.8***	2.53***	2.49***
	WPSFD49207	2.01***	2.12***	2.01***	2.35***	1.99***	1.87***
	WPSFD49502	1.85***	1.84***	1.85***	2.11***	1.96***	1.74***
	WPSID61	2.78***	3.21***	3.01***	2.8***	2.04***	1.93***
	WPSID62	2.06***	2.28***	1.72***	1.78***	1.77***	1.82***
DeepVar	CPIAPPSL	0.96	1.02	1.02	1.04	1.06	1.05
	CPIAUCSL	0.57***	0.68***	0.68***	0.67***	0.64***	0.63***
	CPIMEDSL	0.49***	0.5***	0.51***	0.51***	0.51***	0.52***

Table 7 continued

Model	Variable	Forecast Horizon					
		1	2	3	4	5	6
	CPITRNSL	0.85*	1.05	1.04	1.09	1.07	1.05
	CPIULFSL	0.62***	0.76**	0.75**	0.76**	0.73***	0.71***
	CUSR0000SA0L2	0.71***	0.87	0.86	0.86	0.83*	0.8**
	CUSR0000SA0L5	0.58***	0.72***	0.71***	0.71***	0.68***	0.67***
	CUSR0000SAC	0.88	0.99	1.0	1.01	0.96	0.94
	CUSR0000SAD	0.64*	0.78*	0.82**	0.8**	0.79**	0.89
	CUSR0000SAS	0.22***	0.22***	0.23***	0.23***	0.23***	0.23***
	DDURRG3M086SBEA	0.75*	0.85	0.86	0.83*	0.85*	0.85*
	DNDGRG3M086SBEA	0.9	1.12	1.13	1.11	1.04	0.99
	DSERRG3M086SBEA	0.2***	0.22***	0.2***	0.21***	0.21***	0.21***
	OILPRICEx	1.12	1.06	1.12*	1.09	1.1**	1.07
	PCEPI	0.45***	0.56***	0.56***	0.55***	0.54***	0.53***
	PPICMM	1.28	1.18*	1.11	1.1	1.1	1.1
	WPSFD49207	0.87*	0.89*	0.93	0.91	0.9	0.85**
	WPSFD49502	0.91	0.97	0.99	0.98	0.96	0.9
	WPSID61	0.65***	0.8**	0.83**	0.86*	0.88	0.87*
	WPSID62	1.1	1.08*	1.04	1.06	1.05	1.03
ETS	CPIAPPSL	1.06	1.05	1.05	1.03	1.03	1.05
	CPIAUCSL	0.58***	0.67***	0.7***	0.64***	0.63***	0.64***
	CPIMEDSL	0.44***	0.48***	0.5***	0.49***	0.48***	0.51***
	CPITRNSL	0.98	1.04	1.0	1.01	1.0	1.02
	CPIULFSL	0.68***	0.85	0.85	0.78**	0.73***	0.77**
	CUSR0000SA0L2	0.75***	0.82**	0.85*	0.82**	0.8**	0.78**
	CUSR0000SA0L5	0.63***	0.71***	0.75**	0.72***	0.68***	0.67***
	CUSR0000SAC	0.86*	0.92	0.92	0.91	0.9	0.87*
	CUSR0000SAD	0.75	1.04	1.11	0.99	0.86	0.81
	CUSR0000SAS	0.2***	0.23***	0.24***	0.22***	0.22***	0.23***
	DDURRG3M086SBEA	0.71**	0.76**	0.74***	0.74***	0.78**	0.83*
	DNDGRG3M086SBEA	0.98	1.03	1.03	1.0	0.96	0.98
	DSERRG3M086SBEA	0.21***	0.21***	0.2***	0.2***	0.19***	0.2***
	OILPRICEx	1.03	1.03	1.01	1.02	1.02	1.03
	PCEPI	0.49***	0.58***	0.58***	0.57***	0.55***	0.56***
	PPICMM	1.08	1.29***	1.23**	1.23*	1.26*	1.25**
	WPSFD49207	0.9	0.93	0.98	0.96	0.91	0.92
	WPSFD49502	0.92	0.98	0.99	0.99	0.96	0.93
	WPSID61	0.6***	0.8	0.94	0.98	0.97	0.98
	WPSID62	1.0	0.98	1.01	1.01	0.99	1.01
HCNN	CPIAPPSL	0.98	1.03	1.05	1.04	1.0	1.01
	CPIAUCSL	0.59***	0.9	0.96	0.94	0.84*	0.78**
	CPIMEDSL	0.45***	0.47***	0.46***	0.47***	0.47***	0.49***
	CPITRNSL	0.85	1.3***	1.41***	1.4***	1.28***	1.22***
	CPIULFSL	0.65***	1.0	1.07	1.05	0.93	0.87
	CUSR0000SA0L2	0.75**	1.16	1.23**	1.21**	1.09	1.02

Table 7 continued

Model	Variable	Forecast Horizon					
		1	2	3	4	5	6
	CUSR0000SA0L5	0.62***	0.96	1.02	1.02	0.89	0.84*
	CUSR0000SAC	0.87	1.3***	1.35***	1.36***	1.24***	1.17**
	CUSR0000SAD	0.62*	0.8**	0.86**	0.88*	0.96	0.99
	CUSR0000SAS	0.21***	0.24***	0.27***	0.26***	0.26***	0.24***
	DDURRG3M086SBEA	0.81**	0.92	0.93	0.92	0.97	0.98
	DNDGRG3M086SBEA	1.03	1.45***	1.48***	1.49***	1.35***	1.28***
	DSERRG3M086SBEA	0.18***	0.2***	0.24***	0.24***	0.24***	0.23***
	OILPRICEx	1.13*	1.2***	1.19***	1.14***	1.1***	1.1***
	PCEPI	0.52***	0.76***	0.83*	0.84*	0.76***	0.72***
	PPICMM	1.16	1.28***	1.33***	1.3***	1.28***	1.24***
	WPSFD49207	0.93	1.15**	1.24**	1.18**	1.08	1.07
	WPSFD49502	0.95	1.19**	1.27***	1.22***	1.11	1.09
	WPSID61	0.68***	1.06	1.3***	1.31***	1.23***	1.24***
	WPSID62	1.12	1.22***	1.26***	1.23***	1.15***	1.21***
LSTM	CPIAPPSL	1.02	1.04	1.03	1.02	1.03	1.05*
	CPIAUCSL	0.82	0.82	0.82	0.82	0.81	0.79*
	CPIMEDSL	0.59***	0.58***	0.57***	0.57***	0.58***	0.57***
	CPITRNSL	1.14*	1.13*	1.15*	1.14*	1.14*	1.12
	CPIULFSL	0.9	0.9	0.88	0.88	0.87	0.83
	CUSR0000SA0L2	1.09	1.08	1.07	1.07	1.07	1.04
	CUSR0000SA0L5	0.88	0.87	0.87	0.85	0.84	0.82
	CUSR0000SAC	1.13	1.12	1.13	1.12	1.13	1.09
	CUSR0000SAD	0.97	0.97	0.96	0.96*	0.96**	0.96
	CUSR0000SAS	0.31***	0.31***	0.31***	0.3***	0.29***	0.27***
	DDURRG3M086SBEA	0.95	0.96	0.96	0.98	0.99	0.99
	DNDGRG3M086SBEA	1.26**	1.24**	1.27**	1.24**	1.23**	1.18*
	DSERRG3M086SBEA	0.25***	0.25***	0.25***	0.25***	0.25***	0.25***
	OILPRICEx	1.15***	1.21***	1.23***	1.19***	1.15**	1.13**
	PCEPI	0.68***	0.68***	0.69***	0.68***	0.67***	0.68***
	PPICMM	1.27**	1.33***	1.33***	1.26**	1.2**	1.31***
	WPSFD49207	1.04	1.04	1.04	1.02	1.01	1.0
	WPSFD49502	1.06	1.06	1.08	1.05	1.04	1.03
	WPSID61	1.07	1.08	1.11	1.06	1.06	1.07
	WPSID62	1.09	1.13**	1.17***	1.12*	1.12*	1.11*
TFT	CPIAPPSL	1.03	1.07	1.06	1.06*	1.05	1.06*
	CPIAUCSL	0.67***	0.7***	0.7***	0.68***	0.66***	0.66***
	CPIMEDSL	0.44***	0.46***	0.46***	0.46***	0.46***	0.46***
	CPITRNSL	1.07*	1.12***	1.11***	1.09**	1.06	1.06
	CPIULFSL	0.76***	0.8***	0.8***	0.78***	0.76***	0.75***
	CUSR0000SA0L2	0.84**	0.88*	0.88*	0.85**	0.83**	0.82**
	CUSR0000SA0L5	0.72***	0.76***	0.76***	0.74***	0.71***	0.71***
	CUSR0000SAC	0.96	1.0	1.0	0.98	0.95	0.95
	CUSR0000SAD	0.79**	0.87**	0.88**	0.84**	0.83**	0.85**

Table 7 continued

Model	Variable	Forecast Horizon 1	2	3	4	5	6
	CUSR0000SAS	0.22***	0.23***	0.24***	0.24***	0.24***	0.24***
	DDURRG3M086SBEA	0.79***	0.83***	0.83***	0.82***	0.84***	0.86**
	DNDGRG3M086SBEA	1.08	1.1*	1.1**	1.07	1.04	1.04
	DSERRG3M086SBEA	0.2***	0.21***	0.2***	0.2***	0.2***	0.21***
	OILPRICEx	1.06	1.09***	1.08***	1.04*	1.01	1.01
	PCEPI	0.57***	0.59***	0.59***	0.58***	0.57***	0.57***
	PPICMM	1.06	1.08	1.07	1.08	1.08	1.06
	WPSFD49207	1.0	1.03	1.04	1.01	0.98	0.97
	WPSFD49502	1.04	1.07**	1.08**	1.06	1.03	1.02
	WPSID61	0.93	0.98	1.01	1.0	0.99	0.99
	WPSID62	1.09**	1.1**	1.09**	1.1***	1.05	1.06*
XGBoost	CPIAPPSL	1.12	1.31***	1.21**	1.13	1.26**	1.09
	CPIAUCSL	1.28	1.15	1.27	1.31	1.01	1.05
	CPIMEDSL	0.64***	0.76*	0.73**	0.76*	0.83	0.9
	CPITRNSL	1.41**	1.49***	1.22**	1.27***	1.42***	1.51***
	CPIULFSL	1.2	1.11	1.39	1.17	1.14	1.23
	CUSR0000SA0L2	1.38	1.45**	1.55**	1.48**	1.52***	1.11
	CUSR0000SA0L5	1.42	1.27	1.24	1.11	1.0	1.13
	CUSR0000SAC	1.45	1.45**	1.73***	1.6***	1.59***	1.44***
	CUSR0000SAD	0.96	1.08	1.09	1.11	1.18**	1.13
	CUSR0000SAS	0.92	0.76**	0.78	0.7***	0.82	0.77**
	DDURRG3M086SBEA	1.02	1.02	1.24	0.97	1.01	1.47**
	DNDGRG3M086SBEA	1.51***	1.79***	1.87***	2.31***	1.65***	2.21**
	DSERRG3M086SBEA	0.72***	0.75	0.76*	0.78	0.6***	0.88
	OILPRICEx	1.18	1.37**	1.25*	1.37***	1.12	1.2***
	PCEPI	1.1	1.16	1.05	1.15	1.07	0.93
	PPICMM	1.24**	1.45***	1.61***	1.65***	1.58***	1.44***
	WPSFD49207	1.16	1.45***	1.33***	1.33**	1.89**	1.08
	WPSFD49502	1.17	1.25**	1.31***	1.26**	1.32***	1.62**
	WPSID61	0.99	1.24**	1.43***	1.64***	1.15	1.51***
	WPSID62	1.11	1.66***	1.44***	1.35***	1.53***	1.45***

Values indicate MSE relative to the naive forecast. The stars mark different significance levels: * 10%, ** 5% and *** 1%

Acknowledgements This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy through the Center for Analytics - Data - Applications (ADA-Center) within the framework of "BAYERN DIGITAL II" (20-3410-2-9-8).

Author Contributions Nico Beck: statistical analysis, validation of results and writing. Jonas Dovern: supervision, writing, review and editing. Stefanie Vogl: supervision, writing, review and editing.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability The FRED-MD dataset [21] is available at https://research.stlouisfed.org/econ/mccracken/fred-databases/, the exchange rate dataset [17] at https://github.com/laiguokun/multivariate-time-series-data/tree/master/exchange_rate, and the NASDAQ100 dataset [22] at https://cseweb.ucsd.edu/~yaq007/NASDAQ100_stock_data.html.

Declarations

Ethical and Informed Consent for Data Used This article does not contain any studies with human participants or animals performed by any of the authors.

Competing Interests The authors have no competing interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Aastveit KA, Cross JL, van Dijk HK (2023) Quantifying time-varying forecast uncertainty and risk for the real price of oil. J Bus Econ Stat 41(2):523–537. https://doi.org/10.1080/07350015.2022.2039159
2. Ahlers D, Lakonishok J (1983) A study of economists' consensus forecasts. Manag Sci 29(10):1113–1125. https://doi.org/10.1287/mnsc.29.10.1113
3. Alexandrov A, Benidis K, Bohlke-Schneider M et al (2020) Gluonts: Probabilistic and neural time series modeling in python. J Mach Learn Res 21(116):1–6. https://www.jmlr.org/papers/v21/19-820.html
4. Baumeister C, Kilian L (2015) Forecasting the real price of oil in a changing world: A forecast combination approach. J Bus Econ Stat 33(3):338–351. https://doi.org/10.1080/07350015.2014.949342
5. Beck, Nico; Schemm, Julia; Ehrig, Claudia; Sonnleitner, Benedikt; Neumann, Ursula; Zimmermann, Hans Georg (2024): Introducing ProsperNN—a Python package for forecasting with neural networks. In: PeerJ Comput. Sci. 10, e2481. DOI: https://doi.org/10.7717/peerj-cs.2481.
6. Bergstra J, Bardenet R, Bengio Y et al (2011) Algorithms for hyper-parameter optimization. Advances in Neural Information Processing Systems 24
7. Chen T, Guestrin C (2016) Xgboost. In: Krishnapuram B (ed) Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, NY, ACM Digital Library, pp 785–794. https://doi.org/10.1145/2939672.2939785
8. Dai X, Zou Q (2023) Satnet: Upgraded lstm network for mining time series correlation utilizing the self-attention mechanism. In: 2023 International joint conference on neural networks (IJCNN) pp 1–8. https://doi.org/10.1109/IJCNN54540.2023.10191490
9. Diebold F, Mariano R (1995) Comparing predictive accuracy. J Bus Econ Stat 13(3):253–263. https://econpapers.repec.org/article/besjnlbes/v_3a13_3ay_3a1995_3ai_3a3_3ap_3a253-63.htm
10. Drachal K (2021) Forecasting selected energy commodities prices with bayesian dynamic finite mixtures. Energy Economics 99:105283. https://doi.org/10.1016/j.eneco.2021.105283
11. Eicher TS, Kawai R (2023) Imf trade forecasts for crisis countries: Bias, inefficiency, and their origins. Int J Forecast 39(4):1615–1639. https://doi.org/10.1016/j.ijforecast.2022.07.006
12. Hewamalage H, Ackermann K, Bergmeir C (2023) Forecast evaluation for data scientists: Common pitfalls and best practices. Data Min Knowl Disc 37(2):788–832. https://doi.org/10.1007/s10618-022-00894-5
13. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
14. Hyndman R, Koehler AB, Ord JK et al (2008) Forecasting with exponential smoothing: the state space approach. SpringerLink Bücher, Springer. https://doi.org/10.1007/978-3-540-71918-2
15. Hyndman RJ, Athanasopoulos G (2018) Forecasting: Principles and Practice, 2nd edn. OTexts, Lexington, Ky
16. Jarque CM, Bera AK (1980) Efficient tests for normality, homoscedasticity and serial independence of regression residuals. Econ Lett 6(3):255–259. https://doi.org/10.1016/0165-1765(80)90024-5. https://www.sciencedirect.com/science/article/pii/0165176580900245
17. Lai G, Chang WC, Yang Y, et al (2018) Modeling long- and short-term temporal patterns with deep neural networks. In: Collins-Thompson K (ed) The 41st International ACM SIGIR conference on research & development in information retrieval. ACM, New York, NY, ACM Conferences, pp 95–104. https://doi.org/10.1145/3209978.3210006
18. Lara-Benítez P, Carranza-García M, Riquelme JC (2021) An experimental review on deep learning architectures for time series forecasting. Int J Neural Syst 31(3):2130001. https://doi.org/10.1142/S0129065721300011
19. Li H, Shao J, Liao K et al (2022) Do simpler statistical methods perform better in multivariate long sequence time-series forecasting? In: Al Hasan M (ed) Proceedings of the 31st ACM international conference on information & knowledge management. association for computing machinery, New York, NY, United States, ACM Digital Library, pp 4168–4172. https://doi.org/10.1145/3511808.3557585
20. Lim B, Arık SÖ, Loeff N et al (2021) Temporal fusion transformers for interpretable multi-horizon time series forecasting. Int J Forecast 37(4):1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012
21. McCracken MW, Ng S (2016) Fred-md: A monthly database for macroeconomic research. J Bus Econ Stat 34(4):574–589. https://doi.org/10.1080/07350015.2015.1086655
22. Qin Y, Song D, Chen H et al (2017) A dual-stage attention-based recurrent neural network for time series prediction. In: Sierra C (ed) international joint conferences on artificial intelligence (IJCAI

2017). Curran Associates Inc, Red Hook, NY, pp 2627–2633. https://doi.org/10.24963/ijcai.2017/366
23. Rubaszek M (2021) Forecasting crude oil prices with dsge models. Int J Forecast 37(2):531–546. https://doi.org/10.1016/j.ijforecast.2020.07.004
24. Salinas D, Bohlke-Schneider M, Callot L et al (2019) High-dimensional multivariate forecasting with low-rank gaussian copula processes. Advances in Neural Information Processing Systems 32
25. Salvatier J, Wiecki TV, Fonnesbeck C (2016) Probabilistic programming in python using pymc3. PeerJ Computer Science 2:e55. https://doi.org/10.7717/peerj-cs.55
26. Virtanen P, Gommers R, Oliphant TE et al (2020) Scipy 1.0: fundamental algorithms for scientific computing in python. Nature Methods 17(3):261–272. https://doi.org/10.1038/s41592-019-0686-2. https://www.nature.com/articles/s41592%e2%80%90019%e2%80%900686%e2%80%902
27. Wang Z, Kong F, Feng S et al (2024) Is mamba effective for time series forecasting?. http://arxiv.org/pdf/2403.11144
28. Wu H, Xu J, Wang J et al (2021) Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Adv Neural Inf Process Syst 34:22419–22430
29. Wu Z, Pan S, Long G et al (2020) Connecting the dots: Multivariate time series forecasting with graph neural networks. In: Gupta R (ed) Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. association for computing machinery, New York, NY, United States, ACM Digital Library, pp 753–763, https://doi.org/10.1145/3394486.3403118
30. Xu Z, Lv Z, Chu B et al (2024) A fast spatial-temporal information compression algorithm for online real-time forecasting of traffic flow with complex nonlinear patterns. Chaos, Solitons, and Fractals 182:114852. https://doi.org/10.1016/j.chaos.2024.114852
31. Yin X, Han Y, Sun H et al (2020) A multivariate time series prediction schema based on multi-attention in recurrent neural network. In: 2020 IEEE symposium on computers and communications (ISCC). IEEE, pp 1–7, https://doi.org/10.1109/iscc50000.2020.9219721
32. Zhou H, Zhang S, Peng J et al (2021) Informer: Beyond efficient transformer for long sequence time-series forecasting. Proceedings of the AAAI Conf Artif Intell 35(12):11106–11115. https://doi.org/10.1609/aaai.v35i12.17325
33. Zhou T, Ma Z, Wang X et al (2022a) Film: Frequency improved legendre memory model for long-term time series forecasting. Adv Neural Inf Process SysT 35:12677–12690. https://proceedings.neurips.cc/paper_files/paper/2022/file/524ef58c2bd075775861234266e5e020-Paper-Conference.pdf
34. Zhou T, Ma Z, Wen Q et al (2022b) Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. International Conference on Machine Learning pp 27268–27286. https://proceedings.mlr.press/v162/zhou22g.html
35. Zimmermann HG, Grothmann R, Tietz C et al (2011) Market modeling, forecasting and risk analysis with historical consistent neural networks. In: Hu B (ed) Operations research proceedings 2010. Operations Research Proceedings, Springer, pp 531–536. https://doi.org/10.1007/978-3-642-20009-0_84
36. Ouyang Z, Ravier P, Jabloun M (2022) Are deep learning models practically good as promised? a strategic comparison of deep learning models for time series forecasting. In: 30th European signal processing conference (EUSIPCO), pp 1477–1481. https://doi.org/10.23919/EUSIPCO55093.2022.9909926

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.