A Stronger Univariate Benchmark for Foundational Time-Series Model Evaluation

Peter Cotton
peter.cotton@microprediction.com

October 8, 2026

Abstract. Reported progress for foundation time-series models rests on comparisons that leave out a strong classical reference. Across eleven benchmark papers the classical arms are per-window refits of ETS, ARIMA and Theta; no online forecaster appears, and no suite scores a full predictive density by log score. We supply the missing reference, `laplace`, an online distributional forecaster with no pretraining and no tunable parameters, and score it against TimesFM 3.0 one step ahead on held-out log density over an 18,815-series panel of FRED and M4-hourly series. The comparison is decided by a step the suites leave unstated: how the model's nine quantiles become a density. Under a tight reconstruction the reference leads by half a nat on the median series of every stratum; under a generous one the two are level at the median, and the reference keeps the better coverage. The reference is a `pip install`, with a Rust core and bit-exact parity.

Keywords: time series, foundation models, benchmarking, probabilistic forecasting, proper scoring rules, online learning.

1 Introduction

The comparisons that report progress for foundation time-series models leave out a strong classical reference.[1] A foundation model forecasts from a fixed window of recent history, with weights frozen at pretraining. An online forecaster carries state and revises it with every observation. The benchmarks for the first kind contain no example of the second.

Table 1 enumerates the classic arms of eleven studies, read from the papers rather than from their leaderboards: GIFT-Eval [Aksu et al., 2024], Chronos [Ansari et al., 2024], Chronos-2 [Ansari et al., 2025], TimesFM [Das et al., 2024], TiRex [Auer et al., 2025], Moirai [Woo et al., 2024], Moirai-MoE [Liu et al., 2024], fev-bench [Shchur et al., 2025], Lag-Llama [Rasul et al., 2023], TimeGPT-1 [Garza and Mergenthaler-Canseco, 2023], and the Monash archive [Godahewa et al., 2021a], which supplies several of the others' baseline tables. Eighteen distinct methods appear across the eleven, and nine of those appear in exactly one paper. The working set is much smaller.

Three methods dominate. Automated ETS, ARIMA and Theta each appear in ten of the eleven. With seasonal naive they form a quartet that is the complete classic roster of six studies. No method appears in all eleven, though Theta does if its dynamically optimized variant is treated as the same model, which Garza and Mergenthaler-Canseco [2023] declines to do by listing both.

Five of the eleven ran no classical model at all for at least one headline comparison. Ansari et al. [2025] takes its classic numbers from Shchur et al. [2025] and the Aksu et al. [2024] leaderboard; Auer et al. [2025] takes its from the same leaderboard and from Ansari et al. [2024]; and Das et al. [2024], Woo et al. [2024] and Liu et al. [2024] inherit a Godahewa et al. [2021a] table, the last at two removes.

[1] All results reproduced at `https://skaters.microprediction.org`.

Method	GIFT-Eval	Chronos	Chronos-2	TimesFM	TiRex	Moirai	Moirai-MoE	fev-bench	Lag-Llama	TimeGPT-1	Monash	n
(Auto)ETS	•	•	○	○		○	○	•	•	•	•	10
(Auto)ARIMA	•	•	○	○	○	⊙	○	•	•		•	10
(Auto)Theta	•	•	○	○	○	○	○	•		•	•	10
Seasonal naive	•	•	○		○	•	•	•		•		8
Naive	•	•	○	•		•		•				6
SES				○		○	○				•	4
TBATS				○		○					•	3
DHR-ARIMA				○		○					•	3
SCUM ensemble		•	○					•				3
DynOptTheta									•	•		2
Croston									•	•		2
AutoCES										•		1
ADIDA										•		1
IMAPA										•		1
Drift								•				1
HistoricAverage										•		1
ZeroModel										•		1
NPTS									•			1
`laplace`												0

Table 1: Every classic arm in eleven studies. • the authors ran it; ○ the number is inherited from another paper's table or from a leaderboard; ⊙ both, in different experiments. n counts the studies carrying each method. Tables inherited from Godahewa et al. [2021a] print ARIMA and its harmonic-regression variant as one column applied to disjoint frequency groups; both rows are marked. The two rows below the rule are the methods that issue a density without a per-window fit.

That matters because of what a Monash number is. Those models were run once, at library defaults, and the archive says so. Its ETS is forced non-seasonal above period 24 because the implementation cannot take longer cycles, and its harmonic regression is pinned to a single Fourier term. The table also reports point error only, so an inherited ETS column is a 2021 point-MAE number sitting beside a modern probabilistic score.

The arms that are run fare little better. Ansari et al. [2024] used library defaults with no tuning. Aksu et al. [2024] truncates statistical arms to 1000 observations and replaces any that exceeds a one-day limit with seasonal naive. Shchur et al. [2025] truncates to the same 1000 and caps the seasonal period at 200, and reports that its ensemble and automated ARIMA each failed four of a hundred tasks.

Statefulness and shape are separate matters, and neither is present. No arm in any of the eleven carries state between forecasts: the *statsforecast* predict path is handed an array, fits, forecasts, and keeps nothing. Seasonal naive, meanwhile, is a point forecaster serving as the probabilistic normalizer in seven of the eleven, and as the substituted score when another arm times out in two of them.

The last two rows of Table 1 are the methods in this literature that issue a density without a per-window fit. *NPTS* is training-free and is a sampling distribution by construction, and it ships in both *GluonTS* and *AutoGluon*, which these studies import for their datasets, their metrics and their deep baselines. It is carried by one of the eleven

[Rasul et al., 2023].

We scored *NPTS* through the same protocol as every other arm here. It loses to `laplace` on the median series in all four strata, by 0.05 nats on daily prices and 0.39 on daily economics, over 1,500 series per stratum where the corpus allows. The arm the field already ships and does not run is beatable. That sets the floor this paper's reference has to clear.

This paper supplies such an arm and reports what it does to the comparison. The contribution is deliberately small: a reference, a protocol, and four tables.

2 `laplace`: an online distributional forecaster

`laplace` imports nothing, requires no GPU, performs no fitting step, and exposes no tunable hyperparameters. One parameter-free call returns a predictive density.

The predictive object is a weighted mixture of Gaussians. That choice buys three things a reference arm needs: the mixture is closed under affine maps and under mixing, it admits a closed-form CRPS, and it approximates heavy tails as a scale mixture. A density can therefore be pushed through an invertible map and scored against a proper rule [Gneiting and Raftery, 2007] without sampling.

A *transform* is a stateful causal map that consumes one observation and emits one transformed value, paired with an inverse that rebuilds predictive distributions for future observations from the transformed coordinate, given the state it has accumulated. Differencing, standardisation, online autoregression, *GARCH*-style scaling, seasonal differencing and the Yeo–Johnson coordinates are all transforms. *Conjugation* composes one with an inner forecaster: the observation is transformed, the inner model issues a density in the new coordinate, and the inverse carries that density back. Multi-step uncertainty accumulates through the inverse rather than by simulation.

`laplace` is one fixed composition of these parts, with no knob exposed. Candidate models enter it only as mean forecasters, weighted online by predictive likelihood, so that a shaped leaf governs the tail rather than the dispersion of candidate means. A lattice projection places atoms on the exact values a grid-quoted series revisits, and vanishes identically on continuous data. Beyond one step the predictive mixes over decimated clocks, since the fine clock wins short horizons and a coarse one wins long. A conditional tail fit splices the extreme quantiles, which a bulk fit leaves too thin.

Cotton [2026] derives all of this. We restate only what a reader needs to judge the comparison.

Two properties matter for a reference arm, and neither is accuracy. The first is cost. Adding `laplace` to a leaderboard is a `pip install` and one function call, with no weights to download, no device to allocate, and no context length to choose. The pure-Python implementation, 288 KB of source across 28 files, is the reference; a Rust core runs the same arithmetic about twenty times faster.

Table 2 gives the runtime, and the two regimes disagree. Cold is one forecast for a series not seen before, which is what a benchmark harness asks for. Warm is one new observation on a series already tracked, which is what a deployment asks for.

Warm is the regime that separates the arms. One online update costs 0.012 milliseconds, against 7.44 for the cheapest foundation model here and 329.83 for the largest.

That per-forecast figure compounds, because a benchmark asks for a forecast at every step of every series. Table 3 multiplies it out over the 1,204,288 one-step forecasts this panel contains.

The spread runs from fifteen seconds to four and a half days. It also accounts for the

	cold (ms)		warm (ms)
Model	batch 1	batch 64	
`laplace` (Rust)	3.39	—	0.012
`laplace` (Python)	67.08	—	0.251
AutoETS (refit)	1.93	—	1.93
Prophet (refit)	29.97	—	29.97
AutoARIMA (refit)	62.13	—	62.13
Chronos-Bolt-small	7.44	0.72	7.44
TiRex	96.47	76.34	95.68
TimesFM-2.5-200M	323.44	5.10	329.83

Table 2: Milliseconds per forecast, context 256, horizon 12, 64 series, one thread, Apple silicon, *skaters* 0.16.0. The TimesFM timed here is TimesFM-2.5-200M; Tables 5 and 7 score TimesFM 3.0. Foundation models batch in the cold regime, so the batched column is reported alongside the per-series one. In the warm regime `laplace` performs one state update; a stateless model re-runs its forward pass over a shifted window and a refitting baseline re-estimates from scratch, so for those the warm cost equals the cold cost.

Model	ms/step	batch 1	batch 64
`laplace` (Rust)	0.012	15 s	—
`laplace` (Python)	0.251	5 min	—
AutoETS (refit)	1.93	39 min	—
Chronos-Bolt-small	7.44	2.5 h	14 min
Prophet (refit)	29.97	10.0 h	—
AutoARIMA (refit)	62.13	20.8 h	—
TiRex	95.68	32.0 h	25.5 h
TimesFM-2.5-200M	329.83	4.6 d	1.7 h

Table 3: Wall clock to score the whole panel one step at a time, single threaded. A stateless model re-runs its forward pass at every step, so its per-step cost is its cold cost, and the batch-64 column is the fair figure for an offline sweep that can run many series in lockstep.

shape of the suites in Table 1: scoring a stateless model at every step of a large panel is expensive enough that one forecast origin per series is the affordable design, and four of the eleven use exactly that.

The second is reproducibility across machines. The Rust core uses portable mathematics (*libm*) and produces identical bits on x86 and ARM, and a shared parity suite checks roughly 100,000 probe values to 10^{-6} per implementation. A number one person publishes therefore recomputes on another person's hardware, which is the property a denominator needs and a pretrained checkpoint behind an inference stack does not have.

Sections 4 and 5 measure how strong it is. `laplace` takes the per-series log-likelihood against every classical arm in Table 1. Against TimesFM 3.0 it is level with or ahead of the model at the median in every stratum, by a margin that Section 6 shows is set by how the model's quantiles are turned into a density. Cotton [2026] reports the same ordering against neural and foundation-model baselines across the non-price FRED universe; *GARCH-t* on asset prices is the one arm that escapes, and that split is reported rather than averaged away.

	LL win		CRPS win		
Arm	raw	family	raw	family	N
AutoARIMA (*statsforecast*)	87 %	87 %	58 %	63 %	1688
AutoETS (*statsforecast*)	95 %	94 %	77 %	78 %	1688
Theta (R)	92 %	95 %	74 %	79 %	1917
auto.arima (R)	84 %	85 %	51 %	60 %	1917

Table 4: Share of series on which `laplace` scores better, over 5,397 continuous non-price FRED series in 1,213 correlated families. The family figure averages within a family first, so that a cluster of related series counts once. N is the subset each arm covers under its compute budget. Every arm forecasts one step ahead at every point, conditioned on all data up to that point, with coefficients re-estimated every 25 steps.

3 Protocol

Every arm sees the same series, the same windows and the same scoring rule. The score is held-out log predictive density, differenced per series so that each series' intrinsic difficulty cancels. Two conditioning protocols appear. The foundation-model comparison (Tables 5, 7 and 6) fixes the context at 128 observations, rolls the test window one step ahead, and fits no arm to the evaluation data. The classic-arm comparison (Table 4) is the companion paper's protocol: each arm conditions on all data up to the forecast point, with coefficients re-estimated every 25 steps. The two tables are not differenced against each other.

We report the median of that difference and the per-series win, draw and loss counts from a Diebold–Mariano test with HAC standard errors and a draw band, following Diebold and Mariano [1995]: two-sided at the five percent level, with a difference inside ± 1.96 standard errors counted as a draw. The long-run variance is Newey–West with a Bartlett kernel and lag $\lfloor n^{1/3} \rceil$ for a series of n scored steps; series with fewer than eight finite differences are counted as draws. Log densities are floored at -20 before differencing. We do not report a pooled mean. Averaging a log density across heterogeneous series estimates nothing, because a few near-constant series dominate the sum; Koning et al. [2005] made the corresponding argument about the M3 competition, and Hewamalage et al. [2023] prescribe per-series testing.

The context is fixed rather than growing. Several foundation models degrade beyond the window they were pretrained on, and most of them silently truncate over-long input rather than raising, so feeding four thousand points to a model configured for two thousand scores a two-thousand-context model without saying so. A protocol that lets context grow needs a declared per-model cap, set at the documented architectural maximum, and recorded per arm.

Recording it matters for a second reason. The Woo et al. [2024] benchmark numbers come from a per-dataset search over context lengths from one thousand to five thousand, jointly with patch size, which the arms it is compared against do not receive.

4 Against the classic arms

The arms of Table 1 are the denominator every published skill score divides by, so their strength sets what a skill score means. Table 4 scores `laplace` against them on the non-price FRED universe.

On log-likelihood the margin is wide: the closest arm loses on 84 % of series and the

Stratum	N	Win / draw / loss	Median ΔLL	Loss rate
Daily, economic	2,179	66/923/1,190	−0.854	54.6 %
Daily, price/returns	7,633	15/4,593/3,025	−0.616	39.6 %
Weekly, economic	3,013	517/1,481/1,015	−0.520	33.7 %
Monthly, economic	5,576	193/3,592/1,791	−0.647	32.1 %
M4-hourly, seasonal	414	8/241/165	−0.543	39.9 %

Table 5: TimesFM 3.0 against `laplace`, one step ahead on held-out log density, 128-length context. ΔLL is the per-series difference, model minus reference, reported as a median. The loss rate is the share of series on which the model loses.

widest on 95 %. On CRPS the arms do better, which is the expected pattern, since CRPS is forgiving of a mis-shaped tail that a log score punishes.

These arms are stronger than the ones the studies of Table 1 run.[2] Their harnesses re-estimate at each window, but the forecast for step j of a window conditions only on data up to the window origin, so inside a window of length h the baseline never sees the observations it is forecasting past. Four of the eleven use a single origin per series.

This is the sense in which the incumbent denominators are soft. A skill score computed against them credits a model for clearing a bar that a dependency-free online method clears by a wide margin on the same data.

5 Against TimesFM 3.0

Table 5 scores TimesFM 3.0 [Das et al., 2024] zero-shot against the reference on an 18,815-series panel: four FRED strata and the 414 M4-hourly series. The model's nine deciles are turned into a density by the local-bandwidth mixture (`quantile_dist`, the `local` method of Section 6). Negative favours the reference.

Under the local reconstruction the model trails the reference on the median series in every stratum, by roughly half a nat, and loses on more series than it wins.

Table 6 rescores the same forecasts under the widest reconstruction of Section 6, three times the local bandwidth. The median gap closes to between −0.08 and +0.05 nats; the model wins more series than it loses on the daily price, weekly and monthly strata; and its central 90% coverage rises from the 0.81 to 0.85 of Table 5 to 0.86 to 0.89, against the reference's 0.89 to 0.92. The half-nat lead belongs to the reconstruction, not to the reference. What survives the choice is that the reference is never behind at the median by more than 0.05 nats and is better calibrated in every stratum.

Table 7 repeats the comparison with the model wrapped in a post-hoc portfolio: a long-only online log-score blend of the reference with the model's issued density after PIT recalibration against the reference, the weights started two log units toward the reference, and the mixture rebuilt on the standard quantile grid. The wrap requires no retraining and no access to model internals. N is smaller than in Table 5 because the recalibration needs 128 further warm-up observations, and series that cannot supply them are dropped.

[2] The two differ in which half of the model goes stale. *statsforecast*'s `refit` argument freezes the fitted coefficients between refits but hands `forward` the current history, so the arms here are coefficient-stale and conditioning-current. The frozen parameters include the innovation variance, so the interval inherits a scale up to 25 steps old, which is the one setting that could flatter Table 4. Re-estimating at every step instead, on twenty paired series, moves AutoARIMA by a median of 0.005 nats and AutoETS by −0.000, against margins of 0.31 to 0.59 nats. The published harnesses are the mirror image: they re-estimate at every window and freeze the conditioning.

Stratum	N	Win / draw / loss	Median ΔLL	Loss rate
Daily, economic	2,179	178/1,382/619	−0.082	28.4 %
Daily, price/returns	7,633	506/6,925/202	−0.008	2.6 %
Weekly, economic	3,013	969/1,418/626	+0.001	20.8 %
Monthly, economic	5,576	1,429/3,537/610	+0.046	10.9 %
M4-hourly, seasonal	414	70/214/130	−0.014	31.4 %

Table 6: The forecasts of Table 5 rescored under the widest reconstruction (`wide` in `nozzles.py`, three times the local bandwidth). Same series, same test points, same reference; only the step from nine quantiles to a density differs.

Stratum	N	Win / draw / loss	Median ΔLL	Loss rate
Daily, economic	2,179	76/1,383/720	−0.094	33.0 %
Daily, price/returns	7,633	152/7,136/345	−0.008	4.5 %
Weekly, economic	3,011	467/2,045/499	−0.049	16.6 %
Monthly, economic	5,195	350/4,480/365	−0.056	7.0 %
M4-hourly, seasonal	414	58/352/4	−0.007	1.0 %

Table 7: The same model wrapped in a never-worse portfolio with the reference. The loss rate falls by as much as thirty-nine points; the median stays negative in every stratum.

The wrap cuts the loss rate by as much as thirty-nine points. The median stays negative in every stratum, so the wrapped model still trails the reference.

Because the portfolio contains the reference, the narrowing of the gap mostly measures the reference. What the wrap shows is narrower: the model's density can be combined with the reference without retraining, and the combination loses on fewer series than the raw model.

6 The reconstruction moves the answer more than the model does

A quantile-only model does not emit a density, and a density must exist before a log score does. A reconstruction step builds one from the quantiles, and the choice of step is independent of the model being scored.

Fix TimesFM 3.0 and fix 3,200 held-out test points, then vary only the reconstruction across four methods. The mean log density moves from −0.373 to +2.104. The mean per-point spread is 3.13 nats, which exceeds the entire measured gap between the model and the reference in every stratum of Table 5. Two methods already in independent use elsewhere differ by 0.62 nats.

The reference, scored on the same 3,200 points, has a mean log density of +2.206, above every reconstruction. Paired per series, the reference minus the widest reconstruction has a median of −0.008, with the reference ahead on 48 of the 100 series; against the local reconstruction used in Tables 5 and 7 the median is +0.558, with the reference ahead on 75. At full scale the same choice moves the median gap of Table 5 by 0.5 to 0.8 nats in every stratum (Table 6).

The choice therefore belongs in the protocol, stated rather than assumed. It is not peculiar to this study: the Aksu et al. [2024] submission interface requires a mean and nine deciles, so a model emitting a genuine density must reduce it before scoring and a model emitting samples must convert.

That interface's interval score asks for the 2.5th and 97.5th percentiles, and a sample forecast returns a nearest order statistic without interpolation. At the twenty samples the reference *Chronos* notebook uses, those two quantiles are the minimum and the maximum of twenty draws.

7 Horizon, grid, and statefulness

The results above do not contradict any published leaderboard, because no published leaderboard occupies this regime. Three settings separate them.

The horizon never reaches one step. The shortest prediction length in Aksu et al. [2024] is six, and Shchur et al. [2025] use domain horizons such as thirty and one hundred sixty-eight.

The scoring rule is a nine-point grid. What these suites report as CRPS is a mean of weighted quantile losses at the deciles, so nothing outside the tenth and ninetieth percentile enters any published score. Ansari et al. [2025] call that grid a defect and move to twenty-one levels including the first and ninety-ninth.

And no baseline carries state, which is the hole of Section 1. Each setting is defensible for the task those suites set themselves. Together they define a regime in which an online distributional method is never scored.

8 Scope

Everything measured here is univariate and one step ahead. A single pretrained model amortized across many related series can exploit cross-series and panel structure that no univariate method reaches by construction, and that is where pretraining should pay for itself most directly. This paper does not make that comparison, and it is the more decisive one.

Leakage is controlled in this literature by dataset name rather than by time. Ansari et al. [2024] state the temporal standard, that evaluation data should begin after the last pretraining observation, and then decline it as a minimal risk. Meyer et al. [2025] and Oreshkin et al. [2026] report that a corpus used in TimesFM pretraining reappears in its Monash evaluation; their pretraining table lists Traffic at 862 series and 15,122,928 observations, and the Monash archive file [Godahewa et al., 2021b] holds 862 series of exactly 17,544 steps.

The finding that a simple method matches an elaborate one is not new: Zeng et al. [2023] report it for linear models against transformers on point forecasts, and Tan et al. [2024] for ablated language models. This paper differs in what is scored (a density, by log score), in the reference carrying state, and in the horizon being one step.

A calibration framing would have to engage Adler et al. [2026], who find these models better calibrated than automated ARIMA and N-BEATS on standard datasets. Nothing here contradicts that. Calibration on a decile grid and sharpness of a full density are different measurements.

Finally, the panel is FRED, and FRED is the universe these models' pretraining corpora almost entirely omit: the corpus of Woo et al. [2024] is 0.09 % economic by observation, and the entire economics domain of Aksu et al. [2024] is the M4 collection. Whether the gap survives on the energy and transport data these models were built for is an open question, and Wang et al. [2026] suggest it will not.

9 Reproduction

Everything is in the repository https://github.com/microprediction/skaters, directory `papers/stronger-benchmark`. `constants.py` derives every number in Sections 5 and 6 from a version-controlled store. `verify_paper.py` re-derives each one, compares it against the tables above, and exits non-zero on drift. `REPRODUCE.md` gives three tiers: the figures check in seconds from committed files, the summaries rebuild from per-step records in minutes, and the records regenerate from raw data in days. The first tier is

```
git clone https://github.com/microprediction/skaters
cd skaters && pip install numpy
python papers/stronger-benchmark/verify_paper.py
```

which prints one line reporting the table rows and prose figures that match the store.

The data vintage is published as a release asset in the repository `skaters-benchmarking-data`. FRED revises series, so a fresh pull returns different vintages and will not reproduce these numbers. The archive carries 210,574 public-domain series; commercial index series are excluded and listed, which costs the price stratum and nothing else.

References

Coen Adler, Tiffany Chang, Fabian Draxler, Samar Abdi, and Padhraic Smyth. Beyond accuracy: Are time series foundation models well-calibrated? *International Conference on Learning Representations*, 2026. arXiv:2510.16060.

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-Eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. arXiv:2403.07815.

Abdul Fatir Ansari, Oleksandr Shchur, Caner Turkmen, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Andreas Auer et al. TiRex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. *arXiv preprint arXiv:2505.23719*, 2025.

Peter Cotton. Transforms all the way down: Automatic online distributional forecasting by conjugation. Technical Report 7220478, SSRN, 2026. https://ssrn.com/abstract=7220478, doi:10.2139/ssrn.7220478.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *PMLR*, pages 10148–10167, 2024. arXiv:2310.10688.

Francis X. Diebold and Roberto S. Mariano. Comparing predictive accuracy. *Journal of Business and Economic Statistics*, 13(3):253–263, 1995.

Azul Garza and Max Mergenthaler-Canseco. TimeGPT-1. *arXiv preprint arXiv:2310.03589*, 2023.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. *NeurIPS Datasets and Benchmarks Track*, 2021a. arXiv:2105.06643.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Traffic hourly dataset. Zenodo record 4656132, 2021b.

Hansika Hewamalage, Klaus Ackermann, and Christoph Bergmeir. Forecast evaluation for data scientists: Common pitfalls and best practices. *Data Mining and Knowledge Discovery*, 37:788–832, 2023.

Alex J. Koning, Philip Hans Franses, Michèle Hibon, and H. O. Stekler. The M3 competition: Statistical tests of the results. *International Journal of Forecasting*, 21(3): 397–409, 2005.

Xu Liu, Juncheng Liu, Gerald Woo, et al. Moirai-MoE: Empowering time series foundation models with sparse mixture of experts. *arXiv preprint arXiv:2410.10469*, 2024.

Marcel Meyer, Sascha Kaltenpoth, Kevin Zalipski, and Oliver Müller. Rethinking evaluation in the era of time series foundation models: (un)known information leakage challenges. *arXiv preprint arXiv:2510.13654*, 2025.

Boris N. Oreshkin, Aakash Jauhari, Rohan Selvam, et al. Zero-shot forecasting by simulation alone. *arXiv preprint arXiv:2601.00970*, 2026.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, et al. Lag-llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.

Oleksandr Shchur, Abdul Fatir Ansari, Caner Turkmen, Lorenzo Stella, Nick Erickson, et al. fev-bench: A realistic benchmark for time series forecasting. *arXiv preprint arXiv:2509.26468*, 2025.

Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? In *Advances in Neural Information Processing Systems*, volume 37, 2024. arXiv:2406.16964.

Y. Wang, X. Quan, Z. Yang, and A. Srivastava. Spectral predictability and the limits of zero-shot time series forecasting. *arXiv preprint arXiv:2511.08884*, 2026.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. arXiv:2402.02592.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. arXiv:2205.13504.