fev-bench: A Realistic Benchmark for Time Series Forecasting

Oleksandr Shchur [1 *] **Abdul Fatir Ansari** [1 *] **Caner Turkmen** [1] **Lorenzo Stella** [1] **Nick Erickson** [1]
Pablo Guerron [2 3] **Michael Bohlke-Schneider** [1] **Yuyang Wang** [1]

arXiv:2509.26468v1 [cs.LG] 30 Sep 2025

Abstract

Benchmark quality is critical for meaningful evaluation and sustained progress in time series forecasting, particularly given the recent rise of pretrained models. Existing benchmarks often have narrow domain coverage or overlook important real-world settings, such as tasks with covariates. Additionally, their aggregation procedures often lack statistical rigor, making it unclear whether observed performance differences reflect true improvements or random variation. Many benchmarks also fail to provide infrastructure for consistent evaluation or are too rigid to integrate into existing pipelines. To address these gaps, we propose `fev-bench`, a benchmark comprising 100 forecasting tasks across seven domains, including 46 tasks with covariates. Supporting the benchmark, we introduce `fev`, a lightweight Python library for benchmarking forecasting models that emphasizes reproducibility and seamless integration with existing workflows. Using `fev`, `fev-bench` employs principled aggregation methods with bootstrapped confidence intervals to report model performance along two complementary dimensions: win rates and skill scores. We report results on `fev-bench` for various pretrained, statistical and baseline models, and identify promising directions for future research.

1. Introduction

Pretrained time series forecasting models are transforming forecasting practice. They often deliver more accurate forecasts than traditional methods (Aksu et al., 2024), while enabling zero-shot inference that simplifies production use and lowers barrier to entry for practitioners (Cohen et al., 2025).

Advances in pretrained forecasting models are largely assessed empirically on benchmarks, making benchmark quality essential for continued progress. Oversights in forecasting benchmarks propagate directly to the model development. For example, most general forecasting benchmarks completely ignore covariates despite their prevalence in real-world applications (Bojer & Meldgaard, 2021; Arango et al., 2025). As a result, the majority of pretrained models lack covariate support, limiting their effectiveness in domains like retail where promotional data and pricing information are essential for accurate demand forecasting (Fildes et al., 2022).

Beyond task coverage, existing benchmarks often lack statistical rigor. Most studies report single-number summaries, leaving it unclear whether improvements reflect true advances or random variation. Small gains may vanish or even reverse with minor benchmark changes (Roque et al., 2025). This undermines the reliability of conclusions and can mislead researchers and practitioners about which models perform better.

Finally, benchmark infrastructure presents additional barriers to progress and reproducibility. Many benchmarks provide only standalone datasets without evaluation code, leading to inconsistent implementations across studies that make results incomparable (Hewamalage et al., 2023). When infrastructure exists, it often consists of monolithic systems bundling models, datasets, and evaluation logic with extensive dependencies that become unmaintainable over time. These rigid systems prevent researchers from extending benchmarks to new domains or integrating evaluation into existing workflows, limiting their practical utility and lifespan.

To address these challenges, we make the following three contributions:

- **New benchmark.** We introduce `fev-bench`, a forecast evaluation benchmark containing 100 tasks spanning 7 real-world application domains. Our benchmark addresses a key gap in existing work by including 46 tasks with covariates alongside both univariate and multivariate forecasting scenarios, better reflecting real-world forecasting use cases.
- **Aggregation methods.** In our benchmark, we employ principled aggregation strategies including bootstrap-

[*]Equal contribution [1]AWS [2]Amazon [3]Pablo Guerron holds a concurrent appointment at Amazon and Boston College, and this paper describes work performed at Amazon. Correspondence to: Oleksandr Shchur <shchuro@amazon.com>.

Preprint. Work in progress.

based confidence intervals that quantify whether performance differences are statistically meaningful. This approach enables more reliable model comparisons and assesses the robustness of conclusions to variations in benchmark composition.

- **Evaluation package.** We introduce `fev`[1], a lightweight Python package for forecasting evaluation that introduces minimal dependencies while remaining compatible with popular forecasting libraries. The package focuses on reproducibility and extensibility, enabling researchers to easily build and share new benchmarks and the corresponding results.

2. Preliminaries

Problem definition. The multivariate time series forecasting problem can be formally stated as follows. We are given a collection $\{\boldsymbol{y}_{n,1:T}\}_{n=1}^{N}$ of N multivariate time series. For $n = 1, \ldots, N$ and $t = 1, \ldots, T$, let $\boldsymbol{y}_{n,t} = (y_{n,d,t})_{d=1}^{D} \in \mathbb{R}^D$ denote the D-dimensional observation vector for series n at time t, where the special case $D = 1$ corresponds to univariate forecasting. The goal of multivariate forecasting is to predict the future H values $\boldsymbol{y}_{n,T+1:T+H}$ for each series n, where H is the forecast horizon. Each time series may be accompanied by covariates $\mathbf{X}_{n,1:T+H}$, which include (i) *static covariates* that do not vary over time (e.g., item or location identifiers), (ii) *past-only dynamic covariates* observed up to time T (e.g., past measurements of related variables), and (iii) *known dynamic covariates* available for all time steps $1, \ldots, T + H$ (e.g., holiday indicators or planned interventions).

In the most general form, the aim is to model the conditional distribution

$$p\big(\boldsymbol{y}_{n,T+1:T+H} \mid \boldsymbol{y}_{n,1:T}, \mathbf{X}_{n,1:T+H}\big). \tag{1}$$

While this full distributional modeling provides the richest information, it is common in practice to produce *point forecasts* like the conditional mean or median of each $y_{n,d,t}$. Alternatively, it is often sufficient to estimate *predictive quantiles* of the conditional distributions $p(y_{n,d,t} \mid \boldsymbol{y}_{n,1:T}, \mathbf{X}_{n,1:T+H})$. We denote by $\mathcal{Q} \subset (0, 1)$ the set of quantile levels of interest and aim to produce forecasts $\hat{y}^{(q)}_{n,d,t}$ such that $\Pr(y_{n,d,t} \leq \hat{y}^{(q)}_{n,d,t}) = q$ for all $q \in \mathcal{Q}$.

Benchmarks. To evaluate forecasting methods systematically, the general problem must be instantiated as concrete tasks and combined with an evaluation protocol. We refer to such a collection of tasks together with their evaluation and aggregation procedure as a *benchmark*.

Tasks. A task specifies a concrete forecasting problem. It consists of a dataset together with all parameters that define how forecasts are produced and evaluated, including the forecast horizon H, the evaluation cutoff dates, which columns serve as targets or covariates, and the evaluation metric. A single dataset can yield multiple distinct tasks by varying these parameters. The choice of evaluation metric is integral to task definition because different metrics can correspond to different optimal forecasts; combining conflicting metrics within a single task creates ambiguity about the intended goal (Kolassa, 2020).

Rolling evaluation. Each task is evaluated using a *rolling-origin evaluation protocol* with W windows (Hyndman & Athanasopoulos, 2018). Let $\tau_1 < \tau_2 < \cdots < \tau_W$ denote the evaluation cutoffs. At each window $w \in \{1, \ldots, W\}$, the model receives all observations up to τ_w as input and is asked to produce H-step forecasts. Advancing the cutoff creates a sequence of forecast–target pairs, so that a single task produces W evaluation windows. This setup mimics real-world deployment and yields a more robust estimate of model performance over time, especially for datasets consisting of few series.

Aggregation. While tasks define individual evaluation problems, benchmarks also require a method to aggregate results across tasks, enabling us to answer questions like "Is model A more accurate than model B overall?". The aggregation method directly affects the reliability and interpretability of benchmark results.

We discuss task construction in Section 3, aggregation procedures in Section 4, and the software package supporting the benchmark in Section 5.

3. Task definitions

We introduce `fev-bench` (**F**orecast **EV**aluation Benchmark), a comprehensive benchmark designed to address the limitations of existing forecasting evaluation frameworks. Unlike previous benchmarks that focus on narrow domains or do not provide the evaluation infrastructure, `fev-bench` provides broad coverage across real-world applications while ensuring reproducible and statistically sound comparisons.

`fev-bench` comprises 100 forecasting tasks with complete specifications provided in Section A. This section explains our design choices that guided its construction.

3.1. Datasets and tasks

Our goal is to construct a *general* benchmark that is representative of various real-world forecasting applications. To achieve this, the benchmark must cover different domains, frequencies, horizons, and time series characteristics. We consider both univariate and multivariate forecasting problems, with covariates covering dynamic (both past-only and

[1] github.com/autogluon/fev

Benchmark	# datasets	# tasks	# tasks with covariates	# multivariate tasks	Forecast type
Monash (Godahewa et al., 2021)	42	42	0	0	point
LTSF (Zeng et al., 2023)	9	36	0	9	point
TFB (Qiu et al., 2024)	41	116	0	25	point
BasicTS+ (Shao et al., 2024)	20	40	0	20	point
ProbTS (Zhang et al., 2024)	18	18	0	14	point & quantile
Chronos BM2 (Ansari et al., 2024)	27	27	0	0	point & quantile
GIFT-Eval (Aksu et al., 2024)	55	97	0	43	point & quantile
fev-bench (this work)	96	100	46	35	point & quantile

Table 1. Overview of general time series forecasting benchmarks. The fev benchmark contains more unique datasets than existing benchmarks and includes 46 tasks with covariates, addressing a gap in current evaluation frameworks.

Benchmark	Energy	Nature	Cloud	Mobility	Econ	Health	Retail
GIFT-Eval	16	9	8	7	6	5	4
fev-bench	26	5	20	7	10	8	20

Table 2. Number of datasets from different domains in GIFT-Eval (Aksu et al., 2024) and fev-bench (this work).

Benchmark	10S	T	5T	10T	15T	30T	H	D	W	M	Q	Y
GIFT-Eval	2	0	4	2	4	0	13	15	8	5	1	1
fev-bench	0	6	7	2	5	4	22	19	16	7	4	4

Table 3. Number of datasets with different frequencies in GIFT-Eval (Aksu et al., 2024) and fev-bench (this work).

known) and static variables. We evaluate both point and probabilistic forecasting performance.

Datasets. We begin by sourcing time series datasets from established collections including Monash repository (Godahewa et al., 2021), GIFT-Eval (Aksu et al., 2024), and BOOM (Cohen et al., 2025). However, these collections lack datasets with covariates. To address this critical gap, we extend our benchmark by incorporating public datasets from Kaggle (Bojer & Meldgaard, 2021) and domain-specific repositories (Wang et al., 2023; Lago et al., 2021; Arango et al., 2025).

This curation process yields 96 unique time series datasets.[2] We construct 100 forecasting tasks from these datasets through different target column and covariate selections. Among these tasks, 30 include known dynamic covariates, 24 include past dynamic covariates, and 19 include static covariates. These categories are non-exclusive (a task may include multiple types of covariates) and both univariate and multivariate tasks can have associated covariates. Tables 2 and 3 compare our benchmark's coverage against GIFT-Eval, the most comprehensive existing general forecasting benchmark. fev-bench provides substantially more datasets across various domains and frequencies.

Forecast horizons. Many existing benchmarks reuse identical datasets with different horizons (Zeng et al., 2023; Aksu et al., 2024), which creates correlated tasks that provide limited additional insights into model performance. While fev-bench includes both short and long-horizon tasks, we deliberately avoid horizon duplication within datasets. Instead, we select horizons that reflect domain-appropriate forecasting needs, such as 168 steps for hourly energy demand or 30 steps for daily retail sales.

Rolling evaluation. We use rolling window evaluation to balance computational cost with statistical reliability. The number of windows W is determined by dataset size: up to 20 windows for datasets with fewer than 10 series, up to 10 windows for datasets with 10–2000 series, and 1 window for larger datasets. The actual value of W is constrained by available data length, ensuring at least $(2 \times H + 1)$ past observations before the first evaluation window. We aggregate the results across rolling windows with arithmetic mean.

3.2. Evaluation metrics

Each of the 100 tasks in our benchmark evaluates both point and probabilistic forecast accuracy using complementary metrics.

We evaluate point forecast accuracy using Mean Absolute Scaled Error (MASE), following most existing benchmarks (Aksu et al., 2024; Godahewa et al., 2021).

$$\text{MASE} = \frac{1}{NDH} \sum_{n=1}^{N} \sum_{d=1}^{D} \frac{1}{a_{n,d}} \sum_{t=T+1}^{T+H} |y_{n,d,t} - \hat{y}_{n,d,t}|, \tag{2}$$

where $\hat{y}_{n,d,t}$ is the point forecast and $a_{n,d}$ is the historical seasonal error of series n along dimension d, defined as

$$a_{n,d} = \frac{1}{T-m} \sum_{t=m+1}^{T} |y_{n,d,t} - y_{n,d,t-m}|. \tag{3}$$

Here, m is the seasonal period determined by the data frequency (e.g., m=12 for monthly data with yearly seasonality). MASE offers several advantages: it is scale-free, balances contributions across series with different magnitudes,

[2] huggingface.co/datasets/autogluon/fev_datasets

handles trends well, and remains robust when the forecast horizon contains zeros (Hyndman & Koehler, 2006).

We evaluate the probabilistic forecast accuracy using the Scaled Quantile Loss (SQL) computed on quantile levels $\mathcal{Q} = \{0.1, 0.2, ..., 0.9\}$:

$$\text{SQL} = \frac{1}{NDH} \sum_{n=1}^{N} \sum_{d=1}^{D} \frac{1}{a_{n,d}} \sum_{t=T+1}^{T+H} \sum_{q \in \mathcal{Q}} \rho_q(y_{n,d,t}, \hat{y}^{(q)}_{n,d,t}) \tag{4}$$

where $\hat{y}^{(q)}_{n,d,t}$ is the quantile forecast at level q, $a_{n,d}$ is the historical seasonal error (Equation (3)), and $\rho_q(\cdot, \cdot)$ is the quantile loss

$$\rho_q(y, \hat{y}^{(q)}) = \begin{cases} 2 \cdot (1-q) \cdot (\hat{y}^{(q)} - y), & \text{if } y < \hat{y}^{(q)} \\ 2 \cdot q \cdot (y - \hat{y}^{(q)}), & \text{if } y \geq \hat{y}^{(q)}. \end{cases} \tag{5}$$

We adopt the Scaled Quantile Loss (SQL) as our primary probabilistic metric, since it is the natural extension of MASE and inherits its desirable scale-independence properties. Still, SQL remains underutilized in forecasting benchmarks, where the scale-dependent Weighted Quantile Loss (WQL) is more common (Ansari et al., 2024; Aksu et al., 2024). Both SQL and WQL are related to the Continuous Ranked Probability Score (CRPS) (Gneiting & Raftery, 2007), Winkler Score and Weighted Interval Score (Tibshirani, 2023), but they differ in how they weight individual series: SQL normalizes by scale, while WQL aggregates in a scale-dependent manner, analogous to the distinction between MASE and Weighted Absolute Percentage Error (WAPE). Our choice follows the same reasoning as the M4 and M5 competitions, which also relied on SQL-equivalent metrics (Makridakis et al., 2020; 2022).

Both MASE and SQL can encounter numerical issues with intermittent time series where the seasonal error $a_{n,d}$ approaches zero (Hewamalage et al., 2023). We verified that this problem does not occur in our benchmark tasks. For completeness, we also report WQL and WAPE scores alongside our primary metrics to enable additional analysis.

3.3. Representative subset of the tasks

In addition to the full `fev-bench` benchmark consisting of 100 tasks, we provide `fev-bench-mini`, a curated subset of 20 tasks. These tasks are chosen to capture the diversity of covariates, dimensionalities, domains, and horizons present in the full benchmark, while being small enough to enable rapid iteration and reduced computational cost. `fev-bench-mini` approximates the relative performance ranking observed on the full benchmark, making it suitable for model development, ablation studies, and resource-constrained evaluations. More details on `fev-bench-mini` are provided in Section E.

4. Aggregating the results

After evaluating M models on R tasks, we obtain the error matrix $E \in \mathbb{R}^{R \times M}_{\geq 0}$, where E_{rj} denotes the error (e.g., MASE) of model j averaged over all evaluation windows of task r. Lower values correspond to more accurate forecasts. For models that fail to produce forecasts (e.g., due to timeouts or crashes), we substitute E_{rj} with $E_{r\beta}$, where β denotes a predefined baseline model (Seasonal Naive).

4.1. Marginal performance

The primary goal of any benchmark is to rank models by their average performance. We employ two complementary aggregation methods that capture different aspects of model quality.

Average win rate W_j represents the probability that model j achieves lower error than another randomly chosen model $k \neq j$ on a randomly chosen task:

$$W_j = \frac{1}{R(M-1)} \sum_{r=1}^{R} \sum_{\substack{k=1 \\ k \neq j}}^{M} \Big(\mathbb{1}(E_{rj} < E_{rk}) + 0.5 \cdot \mathbb{1}(E_{rj} = E_{rk}) \Big). \tag{6}$$

Here $\mathbb{1}(\cdot)$ is the binary indicator function. Ties ($E_{rj} = E_{rk}$) are treated as half-wins for each model. The win rate ranges from 0 (worst) to 1 (best) and provides an intuitive measure of relative model performance. However, win rate has two limitations: it is insensitive to the magnitude of performance differences and changes as new models are added to the benchmark, motivating our second aggregation method.

Skill score (Hyndman & Athanasopoulos, 2018) S_j quantifies how much model j reduces forecasting error compared to the fixed baseline model β on average:

$$S_j = 1 - \sqrt[R]{\prod_{r=1}^{R} \text{clip}\left(\frac{E_{rj}}{E_{r\beta}}; \ell, u\right)}, \tag{7}$$

where $\text{clip}(x; \ell, u) = \max(\ell, \min(x, u))$ clips x to the interval $[\ell, u]$. We aggregate relative errors across tasks using geometric mean, clipping values between $\ell = 10^{-2}$ and $u = 100$ to avoid excessive influence from extreme values.

Geometric mean aggregation is less sensitive to outliers than the arithmetic mean and ensures that the final ranking remains invariant to the choice of baseline model (Fleming & Wallace, 1986). The geometric mean appropriately handles the multiplicative nature of relative performance comparisons, averaging ratios in a meaningful way where opposing relative errors like $\frac{1}{2}$ and 2 cancel out.

The skill score ranges from 1 (perfect forecasts) to $-\infty$ (arbitrarily poor performance). Positive values indicate that the model outperforms the baseline on average, while negative values indicate underperformance.

4.2. Pairwise comparison

While marginal performance provides overall rankings, pairwise comparisons reveal specific model relationships that may be obscured in aggregate statistics. The above aggregation methods can be easily generalized to comparing any two models j and k.

Pairwise win rate W_{jk} represents the fraction of tasks where model j outperforms model k

$$W_{jk} = \frac{1}{R}\sum_{r=1}^{R}\left(\mathbb{1}(E_{rj} < E_{rk}) + 0.5 \cdot \mathbb{1}(E_{rj} = E_{rk})\right). \tag{8}$$

Pairwise skill score S_{jk} quantifies how much model j reduces error compared to model k on average

$$S_{jk} = 1 - \sqrt[R]{\prod_{r=1}^{R} \operatorname{clip}\left(\frac{E_{rj}}{E_{rk}}; \ell, u\right)}. \tag{9}$$

4.3. Significance of performance differences

A critical concern in benchmarking is the reliability of reported performance differences. State-of-the-art claims often rest on minor improvements that may vanish under small changes to benchmark composition, casting doubt on whether they reflect genuine advances (Roque et al., 2025).

To address this concern, we compute 95% confidence intervals using paired bootstrap over tasks (Efron, 1992). We generate $B = 1000$ bootstrap samples by drawing rows with replacement from E, where each $\tilde{E}^{(b)} \in \mathbb{R}_{\geq 0}^{R \times M}$ contains R tasks sampled from the original benchmark. For each $\tilde{E}^{(b)}$, we compute the aggregate statistics to obtain bootstrap distributions such as $\{\tilde{W}_{jk}^{(b)}\}_b$. The $(1-\alpha)$ confidence interval for the pairwise win rate W_{jk} is then

$$\left[Q_{\alpha/2}\left(\{\tilde{W}_{jk}^{(b)}\}_b\right),\ Q_{1-\alpha/2}\left(\{\tilde{W}_{jk}^{(b)}\}_b\right)\right], \tag{10}$$

where $Q_p(\cdot)$ denotes the empirical p-th quantile of the bootstrap distribution. Analogous intervals are constructed for the pairwise skill scores S_{jk}. These intervals quantify how conclusions about model comparisons vary under alternative benchmark compositions.

We report bootstrap confidence intervals only for the pairwise statistics (W_{jk}, S_{jk}), as these directly answer the question of interest: "Does model j consistently outperform model k under different benchmark compositions?". Confidence intervals for the marginal statistics (W_j, S_j) instead describe the variability of each model's average score in isolation, ignoring the correlations between models.

Summary. Benchmark interpretation proceeds in two steps. First, the marginal statistics (W_j, S_j) provide an overall model ranking. Second, the pairwise statistics with confidence intervals (W_{jk}, S_{jk}) refine this picture by showing which performance differences are robust to changes in benchmark composition. For example, if a model j ranks highest by marginal win rate W_j and all of its pairwise win rates W_{jk} against other models $k \neq j$ have lower bounds above 50%, then model j can be regarded as outperforming every competitor with high confidence.

5. Infrastructure

Comprehensive task coverage and principled evaluation are important, but standardized infrastructure is equally vital for the relevance and longevity of a benchmark. This includes code to define tasks, run evaluations, and aggregate results in a consistent manner.

5.1. Motivation

Existing forecasting benchmarks usually fall into one of two categories: standalone datasets without supporting infrastructure (Godahewa et al., 2021; Zeng et al., 2023) and end-to-end systems which bundle models, datasets and forecasting tasks (Qiu et al., 2024; Aksu et al., 2024).

Benchmarks with standalone datasets provide no guarantees that the results obtained by different users are comparable. Two users may have evaluated on distinct forecasting tasks even when referring to the same dataset. This could be due to differences in the forecast horizon, the forecast start date, or the evaluation metric implementation. Even with identical task specifications, differences in the aggregation strategy can dramatically changing the final conclusion.

While standalone datasets are problematic due to ambiguity, end-to-end systems with models, datasets and tasks are often impractical due to their rigidity. These systems usually come with lots of dependencies and assumptions, which makes extending or integrating them into existing workflows difficult. The model implementations become stale over time as the maintenance overhead is often too high. Restrictions on commonly-used libraries such as `torch`, `numpy`, and `pandas` lead to "dependency hell". Differences in library versions also make the evaluation non-transparent: For example, a different `pandas` version may change how time series frequencies are inferred, which affects the seasonal period m, which in turn alters the computation of MASE.

To address these limitations, we introduce `fev`, a lightweight library that provides essential benchmarking functionality without unnecessary constraints or bloated dependencies. Its core features include task definition, data loading and splitting, prediction scoring, and result aggregation.

`fev` only depends on Hugging Face `datasets` (Lhoest et al., 2021) and `pydantic` (Colvin et al., 2025) libraries for input validation and does not fix versions of commonly-used packages like `torch` or `numpy`, allowing hassle-free integration in existing model pipelines. Using `datasets` enables effortless data loading, avoiding custom file formats or complex data processing pipelines. Moreover, it makes the library future-proof and it can support use cases such as multimodal forecasting (with text and image features) out of the box.

`fev` does not include model implementations, which have the potential of becoming outdated. Instead, it provides adapters which transform data into formats expected by popular forecasting packages like GluonTS (Alexandrov et al., 2020), darts (Herzen et al., 2022), Nixtla libraries (Garza et al., 2022; Olivares et al., 2022), AutoGluon (Shchur et al., 2023), and sktime (Löning et al., 2019).

5.2. `fev` API

The `fev` library is built around three main constructs:

- `EvaluationWindow` — a single train–test split of time series data, defined by a cutoff index or date. This is the smallest unit on which metrics can be computed.
- `Task` — a complete specification of a forecasting problem. It includes the dataset, forecast horizon, initial cutoff, covariates (past-only, future-known, and static), target columns used for evaluation, evaluation metric(s), and any metric-specific parameters such as the seasonal period used for MASE. A `Task` may include one or more `EvaluationWindow`s.
- `Benchmark` — a collection of `Task`s.

Benchmarks in `fev` can be defined using YAML files, allowing users to easily construct custom benchmarks. Each task produces an evaluation summary containing not only the metric values but also the full task specification, ensuring that the results are unambiguous. These summaries make it straightforward to compare results and immediately identify any differences in setup. Additionally, `fev` provides utilities for aggregating results across tasks in a benchmark, as described in Section 4.

6. Related work

6.1. Existing benchmarks

Early works on deep learning approaches for time series forecasting (Lai et al., 2018; Salinas et al., 2020; Rangapuram et al., 2018) did not rely on standardized benchmarks. Instead, they typically evaluated on 4–6 datasets from different domains, which often varied across studies. The M4 (Makridakis et al., 2020) and M5 (Makridakis et al., 2022) competition tasks gained popularity for evaluating forecasting models. The LTSF benchmark (Zhou et al., 2021; Zeng et al., 2023) introduced new datasets with an emphasis on long-horizon forecasting. However, these benchmarks were limited in scope, focusing on narrow domains without the comprehensiveness needed to assess general forecasting models. In addition, the LTSF benchmark has been criticized for its over-representation of similar datasets and impractical evaluation tasks (Hewamalage et al., 2023).

A subsequent wave of benchmarks, including the Monash forecasting repository (Godahewa et al., 2021), BasicTS+ (Shao et al., 2024), TFB (Qiu et al., 2024), ProbTS (Zhang et al., 2024), and Chronos Benchmarks (Ansari et al., 2024), broadened evaluation by covering datasets from diverse domains and supporting both univariate and multivariate forecasting. More recently, GIFT-Eval (Aksu et al., 2024) greatly expanded the range of forecasting tasks with varied domains and frequencies, quickly becoming the standard benchmark for pretrained time series models. Domain-specific benchmarks, such as BOOM (Cohen et al., 2025), have also been introduced. Yet, none of these benchmarks include forecasting tasks with covariates, despite their immense practical relevance.

In contrast, `fev-bench` provides broader dataset and domain coverage, including a substantial proportion of tasks with covariates, along with domain-appropriate forecasting setups. Complementing this, the `fev` library offers standardized infrastructure for evaluation, incorporating principled aggregation strategies and confidence intervals, leading to more robust conclusions.

6.2. Aggregation strategies

Several aggregation methods have been proposed in the literature, each with specific trade-offs for benchmarking.

Average rank has been widely used in the benchmarking literature (Aksu et al., 2024; Ansari et al., 2024). As we show in Section C.1, average rank is mathematically equivalent to the average win rate W_j. The two induce exactly the same ordering of models, differing only by an affine transformation. Since ranks scale with the number of models and lack a natural extension to pairwise comparisons, we prefer reporting win rates, which are bounded between 0 and 1 and extend directly to pairwise evaluations.

Bradley–Terry (Elo) scores (Bradley & Terry, 1952) have been applied in diverse domains, from large language models (Chiang et al., 2024) to tabular benchmarks (Erickson et al., 2025). As we show in Section C.2, when all models

Model	Avg. win rate (%)	Skill score (%)	Median runtime (s)	Leakage (%)	# failures
TiRex	86.7	42.6	1.4	1	0
TimesFM-2.5	82.1	42.3	117.6	8	0
Toto-1.0	73.8	40.7	90.7	8	0
Moirai-2.0	68.8	39.3	2.5	28	0
Chronos-Bolt	68.8	38.9	1.0	0	0
TabPFN-TS	66.9	39.6	305.5	0	2
Sundial-Base	49.2	33.4	35.6	1	0
Stat. Ensemble	48.7	20.2	690.6	0	11
Seasonal Naive	21.7	0.0	2.3	0	0

Table 4. Marginal probabilistic forecasting performance of select models (according to the SQL metric) on the full fev-bench benchmark. Results for all models and other metrics are available in Section D.

Model	Avg. win rate (%)	Skill score (%)	Median runtime (s)	Leakage (%)	# failures
TiRex	80.5	30.0	1.4	1	0
TimesFM-2.5	79.9	30.3	117.6	8	0
Toto-1.0	69.9	28.2	90.7	8	0
Moirai-2.0	65.2	27.3	2.5	28	0
Chronos-Bolt	64.8	26.5	1.0	0	0
TabPFN-TS	62.0	27.6	305.5	0	2
Sundial-Base	56.7	24.7	35.6	1	0
Stat. Ensemble	51.0	15.7	690.6	0	11
Seasonal Naive	22.3	0.0	2.3	0	0

Table 5. Marginal point forecasting performance of select models (according to the MASE metric) on the full fev-bench benchmark. Results for all models and other metrics are available in Section D.

are compared on all tasks, Bradley–Terry scores induce the same model ranking as average win rates. In other words, average rank, average win rate, and Bradley–Terry scores are equivalent in terms of model ordering given our setup in Section 4. We choose to report win rates for simplicity.

Nemenyi post-hoc test with critical difference diagrams (Demšar, 2006) is another rank-based method, since it relies on average ranks as input. It controls family-wise error rates but is often overly conservative, frequently failing to detect meaningful differences (Garcia & Herrera, 2008). Moreover, it yields only binary significance decisions without quantifying effect sizes. Confidence intervals on win rates provide a more informative alternative: they preserve the intuitive interpretation of win rates and display the magnitude and uncertainty of performance differences directly.

Geometric mean relative error (GMRE) (Ansari et al., 2024; Aksu et al., 2024) yields rankings identical to skill scores since $\mathrm{GMRE}_j = 1 - S_j$. We adopt this established approach with two modifications: extreme values are clipped to limit the contribution of outliers, and we report its complement (the skill score) to maintain a consistent “higher-is-better” interpretation across all aggregation methods.

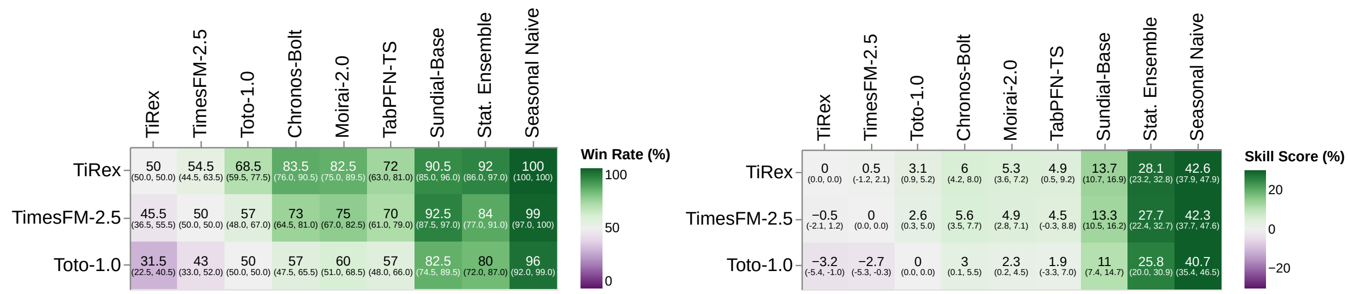
7. Results

In this section we present the experimental evaluation of various forecasting models on the fev-bench benchmark.

7.1. Setup

Models. Our evaluation focuses primarily on pretrained forecasting models that represent the current frontier in time series forecasting research. We select models based on three criteria: strong performance on existing benchmarks such as GIFT-Eval, publicly available implementations, and computational feasibility on consumer hardware (single NVIDIA A10G GPU with 24GB RAM). More details about the model configuration are provided in Section B.

We evaluate seven pretrained models: TimesFM-2.5 (Das et al., 2024), TiRex (Auer et al., 2025), Chronos-Bolt (Base) (Ansari et al., 2024), Toto-1.0 (Cohen et al., 2025), Moirai-2.0 (Woo et al., 2024), TabPFN-TS (Hoo et al., 2025), and Sundial (Liu et al., 2025). Among these models only Toto-1.0 natively supports multivariate forecasting ($D > 1$). For the remaining tasks, we first convert each multivariate time series in the original dataset into D separate univariate series, one for each dimension. TabPFN-TS is the only model that supports known covariates, the remaining models ignore all



(a) (b)

Figure 1. Pairwise win rates (a) and skill scores (b) of the top-3 models against other models under the SQL metric on `fev-bench`, with 95% confidence intervals obtained via bootstrapping. Higher values are better. The confidence intervals show heavy overlap between TiRex and TimesFM-2.5, suggesting no clear winner between the two, while both outperform the remaining models. Full pairwise results are available in Section D. Best viewed on screen.

covariates.

We include statistical baselines representing different modeling approaches: AutoETS, AutoARIMA, and AutoTheta (Garza et al., 2022), plus SCUM ensemble that combines the aforementioned univariate models with AutoCES (Petropoulos & Svetunkov, 2020). We also evaluate simple baselines Seasonal Naive, Naive, and Drift (Hyndman & Athanasopoulos, 2018).

These results represent an initial analysis, and we welcome future submissions from authors of any forecasting approach—pretrained, statistical, or task-specific models.

Evaluation metrics. We follow the evaluation protocol described in Sections 3 and 4. Each task is evaluated using SQL for probabilistic forecasting and MASE for point forecasting, with results aggregated across all 100 tasks using average win rates and skill scores for both marginal and pairwise model comparisons. In addition, we report the following metrics to provide broader context for model performance.

Data leakage. The purpose of `fev-bench` is to assess the *zero-shot* capabilities of pretrained forecasting models. Two kinds of leakage can undermine this goal: (i) if a model has been trained on the benchmark training split, the evaluation is no longer zero-shot; and (ii) if the model has seen the test split, this constitutes direct test leakage. To prevent both, model contributors must indicate, for each model–task pair, whether *any* part of the dataset at the same frequency was used during training. Resampled variants at different frequencies are not considered leakage. For tasks where overlap is reported, we discard the submitted results and impute performance with the 100% zero-shot model that had the highest average win rate at the time of the benchmark release, namely Chronos-Bolt (Base). This eliminates leakage for the affected tasks without heavily penalizing the submitted model.

This leakage indicator is required only for pretrained models. Developers may also submit task-specific models, in which case training on the task's training split is permitted and the zero-shot requirement does not apply. Overall, this policy offers a practical safeguard to separate genuine zero-shot performance from overfitting to benchmark datasets, while recognizing that more subtle forms of leakage may remain difficult to detect.

Runtime. In addition to accuracy indicators, we report the median end-to-end runtime (training plus prediction across all evaluation windows) for each model. While this metric has limitations due to varying hardware, batch sizes, and implementation details, it still offers useful insights into computational efficiency. In practice, it helps distinguish between different modeling paradigms (e.g., sample-based autoregressive models versus direct multi-step forecasters) and between zero-shot and fine-tuned approaches, while also providing an incentive for contributors to develop and submit more efficient implementations.

Model failures. If a model fails to produce a forecast on certain tasks (e.g., due to exceeding the 6 hour runtime limit or encountering an internal error), we replace its performance with the score of the Seasonal Naive baseline.

7.2. Results

Tables 4 & 5 summarize the marginal performance of the top-performing models on `fev-bench`. Complete marginal and pairwise results for all models are provided in Section D. Live results for the leaderboard are available on Hugging Face.[3]

The overall ranking aligns with findings from other benchmarks (Aksu et al., 2024; Ansari et al., 2024). TiRex and TimesFM-2.5 emerge as the top two models, leading the

[3] `huggingface.co/spaces/autogluon/fev-leaderboard`

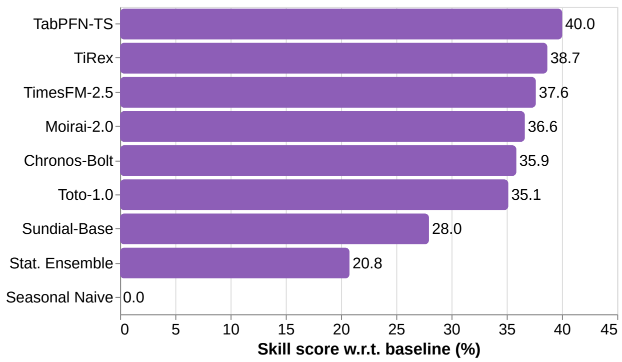


Figure 2. Average skill scores on the 42 fev-bench tasks with dynamic covariates (based on SQL). TabPFN-TS, the only model that uses known covariates, outperforms all others, indicating that pretrained models miss valuable predictive signal from covariates.

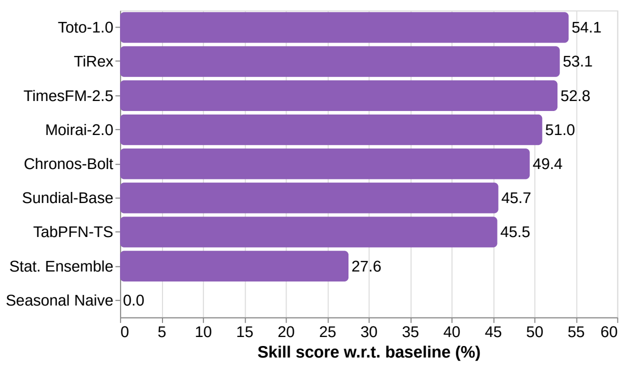


Figure 3. Average skill scores relative to the baseline on 35 multivariate tasks of fev-bench (based on SQL). Toto-1.0, the only multivariate model, outperforms others despite ranking third overall in Table 4, indicating room for improvement on multivariate forecasting.

benchmark in both point and probabilistic forecast accuracy. Older pretrained models such as Toto-1.0, Chronos-Bolt and Sundial rank lower. All pretrained models are substantially more accurate and faster than the best statistical approach (SCUM Ensemble).

To assess whether the observed differences among the leading models reflect genuine improvements rather than evaluation noise, we examine pairwise comparisons under the SQL metric in Figures 1a and 1b. The confidence intervals show no statistically significant gap between TiRex and TimesFM-2.5, despite the difference in their marginal win rates and skill scores in Table 4. This indicates that under different benchmark compositions or task weightings, either of the two could emerge as the top performer. In contrast, both TiRex and TimesFM-2.5 demonstrate clear advantages over all other models, with confidence intervals that separate them from the rest of the field.

7.3. Directions for improvement

Our evaluation highlights key limitations of current pretrained forecasting models, especially on tasks with covariates and on multivariate forecasting.

Forecasting with covariates. On the 42 tasks in fev-bench with dynamic covariates, we compared the same set of pretrained models as in the main evaluation. As shown in Figure 2, TabPFN-TS achieves the highest skill score on covariate tasks (40.0%), clearly ahead of the second-best model TiRex (37.8%). This marks a notable change compared to the overall results (Table 4), where TabPFN-TS ranked fourth by skill score. This demonstrates that current pretrained models leave substantial performance untapped by ignoring covariates.

Multivariate forecasting. Restricting attention to the 35 multivariate tasks in fev-bench, Toto-1.0 outperforms TiRex, reversing the overall ranking (Figure 3). Toto-1.0 is the only model that natively supports multivariate forecasting; all others predict each dimension independently. This advantage underscores the need for pretrained models that handle multivariate series directly, though doing so requires advances in both architecture design and access to multivariate training data.

8. Conclusion

We introduced fev-bench, a benchmark that incorporates covariates, multivariate tasks, and principled aggregation methods. By enabling statistically sound comparisons across a broad range of domains, fev-bench provides a reliable foundation for advancing pretrained forecasting models and evaluating their ability to handle real-world requirements such as covariates and multivariate structure.

Complementing this, we presented fev, a lightweight evaluation package designed for reproducibility and extensibility. fev makes it easy to define specialized benchmarks, integrate them into existing workflows, and share results in a consistent way, lowering barriers for future community-driven progress in time series forecasting.

References

Walmart recruiting - store sales forecasting. Kaggle competition, 2014. URL https://www.kaggle.com/competitions/walmart-recruiting-store-sales-forecasting.

Rossmann store sales. Kaggle competition, 2015. URL https://www.kaggle.com/competitions/rossmann-store-sales.

Recruit restaurant visitor forecasting. Kaggle competition, 2017. URL https://www.kaggle.com/c/recruit-restaurant-visitor-forecasting.

Store sales - time series forecasting. Kaggle competition, 2020. URL https://www.kaggle.com/competitions/store-sales-time-series-forecasting.

CO2 emissions by country. Kaggle dataset, 2025a. URL https://www.kaggle.com/datasets/ulrikthygepedersen/co2-emissions-by-country. Country-level CO_2 emissions from 1960 to present.

Global life expectancy data (1950–2023). Kaggle dataset, 2025b. URL https://www.kaggle.com/datasets/nafayunnoor/global-life-expectancy-data-1950-2023. Life expectancy data by country from 1950 to 2023.

Renewable energy and weather conditions. Kaggle dataset, 2025c. URL https://www.kaggle.com/datasets/samanemami/renewable-energy-and-weather-conditions. Hourly weather and renewable energy generation data.

Tourism and economic impact. Kaggle dataset, 2025d. URL https://www.kaggle.com/datasets/bushraqurban/tourism-and-economic-impact. Key tourism and economic indicators for 200+ countries, 1999–2023.

UK COVID-19 dashboard data. Kaggle dataset, 2025e. URL https://www.kaggle.com/datasets/happyadam73/uk-covid19-dashboard-data-sqlite-compressed. UK COVID-19 data from official UK government sources.

Rohlik sales forecasting challenge V2. Kaggle competition, 2025. URL https://www.kaggle.com/competitions/rohlik-sales-forecasting-challenge-v2/data.

Aksu, T., Woo, G., Liu, J., Liu, X., Liu, C., Savarese, S., Xiong, C., and Sahoo, D. Gift-Eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., et al. GluonTS: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116): 1–6, 2020.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Pineda Arango, S., Kapoor, S., Zschiegner, J., Maddix, D. C., Mahoney, M. W., Torkkola, K., Gordon Wilson, A., Bohlke-Schneider, M., and Wang, Y. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR.

Arango, S. P., Mercado, P., Kapoor, S., Ansari, A. F., Stella, L., Shen, H., Senetaire, H., Turkmen, C., Shchur, O., Maddix, D. C., et al. ChronosX: Adapting pretrained time series models with exogenous variables. *arXiv preprint arXiv:2503.12107*, 2025.

Athanasopoulos, G., Ahmed, R. A., and Hyndman, R. J. Hierarchical forecasts for Australian domestic tourism. *International Journal of Forecasting*, 25(1):146–166, 2009.

Auer, A., Podest, P., Klotz, D., Böck, S., Klambauer, G., and Hochreiter, S. TiRex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. *arXiv preprint arXiv:2505.23719*, 2025.

Bojer, C. S. and Meldgaard, J. P. Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, 37(2):587–603, 2021.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhu, B., Zhang, H., Jordan, M., Gonzalez, J. E., et al. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*, 2024.

Christiano, L. J., Eichenbaum, M., and Evans, C. L. Monetary policy shocks: What have we learned and to what end? *Handbook of macroeconomics*, 1:65–148, 1999.

Cohen, B., Khwaja, E., Doubli, Y., Lemaachi, S., Lettieri, C., Masson, C., Miccinilli, H., Ramé, E., Ren, Q., Rostamizadeh, A., du Terrail, J. O., Toon, A.-M., Wang, K., Xie, S., Xu, Z., Zhukova, V., Asker, D., Talwalkar, A., and Abou-Amal, O. This time is different: An observability perspective on time series foundation models, 2025. URL https://arxiv.org/abs/2505.14766.

Colvin, S., Jolibois, E., Ramezani, H., Garcia Badaracco, A., Dorsey, T., Montague, D., Matveenko, S., Trylesinski, M., Runkle, S., Hewitt, D., Hall, A., and Plot, V. Pydantic Validation, July 2025. URL https://github.com/pydantic/pydantic.

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

David, E., Bellot, J., and Corff, S. L. HERMES: Hybrid error-corrector model with inclusion of external signals for nonstationary fashion time series. *arXiv preprint arXiv:2202.03224*, 2022.

Demšar, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7 (Jan):1–30, 2006.

ECDC. Respiratory viruses weekly data. https://github.com/EU-ECDC/Respiratory_viruses_weekly_data/tree/main, 2025. Open data repository; weekly respiratory virus surveillance in the EU/EEA.

Efron, B. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pp. 569–593. Springer, 1992.

Erickson, N., Purucker, L., Tschalzev, A., Holzmüller, D., Desai, P. M., Salinas, D., and Hutter, F. TabArena: A living benchmark for machine learning on tabular data. *arXiv preprint arXiv:2506.16791*, 2025.

Fildes, R., Ma, S., and Kolassa, S. Retail forecasting: Research and practice. *International Journal of Forecasting*, 38(4):1283–1318, 2022.

Fleming, P. J. and Wallace, J. J. How not to lie with statistics: The correct way to summarize benchmark results. *Communications of the ACM*, 29(3):218–221, 1986.

Garcia, S. and Herrera, F. An extension on" statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of machine learning research*, 9(12), 2008.

Garza, A., Canseco, M. M., Challú, C., and Olivares, K. G. StatsForecast: Lightning fast forecasting with statistical and econometric models. PyCon Salt Lake City, Utah, US 2022, 2022. URL https://github.com/Nixtla/statsforecast.

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

Godahewa, R., Bergmeir, C., Webb, G. I., Hyndman, R. J., and Montero-Manso, P. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*, 2021.

Herzen, J., Lässig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., Pottelbergh, T. V., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., Kościsz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., and Grosch, G. Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, 23(124):1–6, 2022. URL http://jmlr.org/papers/v23/21-1177.html.

Hewamalage, H., Ackermann, K., and Bergmeir, C. Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, 37(2): 788–832, 2023.

Hong, T., Pinson, P., and Fan, S. Global energy forecasting competition 2012, 2014.

Hoo, S. B., Müller, S., Salinas, D., and Hutter, F. From tables to time: How TabPFN-v2 outperforms specialized time series forecasting models. *arXiv preprint arXiv:2501.02945*, 2025.

Hyndman, R. J. and Athanasopoulos, G. *Forecasting: Principles and Practice*. OTexts, 2018.

Hyndman, R. J. and Koehler, A. B. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

Jiang, J., Han, C., Jiang, W., Zhao, W. X., and Wang, J. LibCity: A unified library towards efficient and comprehensive urban spatial-temporal prediction. *arXiv preprint arXiv:2304.14343*, 2023.

Kolassa, S. Why the "best" point forecast depends on the error or accuracy measure. *International Journal of Forecasting*, 36(1):208–211, 2020.

Lago, J., Marcjasz, G., De Schutter, B., and Weron, R. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293:116983, 2021.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104, 2018.

Lhoest, Q., Villanova del Moral, A., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Le Scao, T., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A., and Wolf, T. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-demo.21.

Liu, Y., Qin, G., Shi, Z., Chen, Z., Yang, C., Huang, X., Wang, J., and Long, M. Sundial: A family of highly capable time series foundation models. *arXiv preprint arXiv:2502.00816*, 2025.

Löning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J., and Király, F. J. sktime: A unified interface for machine learning with time series. *arXiv preprint arXiv:1909.07872*, 2019.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1): 54–74, 2020.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4):1346–1364, 2022.

Mancuso, P., Piccialli, V., and Sudoso, A. M. A machine learning approach for forecasting hierarchical time series. *Expert Systems with Applications*, 182:115102, 2021.

McCracken, M. and Ng, S. FRED-QD: A quarterly database for macroeconomic research. Technical report, National Bureau of Economic Research, 2020.

McCracken, M. W. and Ng, S. FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589, 2016.

Mohaddes, K. and Raissi, M. Compilation, revision and updating of the Global VAR (GVAR) Database, 1979Q2-2023Q3. 2024.

of Health Affairs, G. D. and Ministry of Health, S. A. Riyadh hospital admissions dataset (2020–2024). Kaggle dataset, 2025. URL https://www.kaggle.com/datasets/datasetengineer/riyadh-hospital-admissions-dataset-20202024. Hospital admissions in Riyadh over 2020–2024.

Olivares, K. G., Challú, C., Garza, A., Canseco, M. M., and Dubrawski, A. NeuralForecast: User-friendly state-of-the-art neural forecasting models. PyCon Salt Lake City, Utah, US 2022, 2022. URL https://github.com/Nixtla/neuralforecast.

Open Power System Data. Data package time series. https://doi.org/10.25832/time_series/2020-10-06, 2020. (Primary data from various sources, for a complete list see URL).

Palaskar, S., Ekambaram, V., Jati, A., Gantayat, N., Saha, A., Nagar, S., Nguyen, N. H., Dayama, P., Sindhgatta, R., Mohapatra, P., et al. Automixer for improved multivariate time-series forecasting on business and IT observability data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 22962–22968, 2024.

Petropoulos, F. and Svetunkov, I. A simple combination of univariate models. *International journal of forecasting*, 36(1):110–115, 2020.

Qiu, X., Hu, J., Zhou, L., Wu, X., Du, J., Zhang, B., Guo, C., Zhou, A., Jensen, C. S., Sheng, Z., et al. TFB: Towards comprehensive and fair benchmarking of time series forecasting methods. *arXiv preprint arXiv:2403.20150*, 2024.

Rangapuram, S. S., Seeger, M. W., Gasthaus, J., Stella, L., Wang, Y., and Januschowski, T. Deep state space models for time series forecasting. *Advances in Neural Information Processing Systems*, 31, 2018.

Roque, L., Cerqueira, V., Soares, C., and Torgo, L. Cherry-picking in time series forecasting: How to select datasets to make your model shine. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20192–20199, 2025.

Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

Shao, Z., Wang, F., Xu, Y., Wei, W., Yu, C., Zhang, Z., Yao, D., Sun, T., Jin, G., Cao, X., et al. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Shchur, O., Turkmen, A. C., Erickson, N., Shen, H., Shirkov, A., Hu, T., and Wang, B. AutoGluon–TimeSeries: Automl for probabilistic time series forecasting. In *International Conference on Automated Machine Learning*, pp. 9–1. PMLR, 2023.

Staffell, I., Pfenninger, S., and Johnson, N. A global model of hourly space heating and cooling demand at multiple spatial scales. *Nature Energy*, 8(12):1328–1344, 2023.

Tibshirani, R. Forecast scoring and calibration, 2023. URL https://www.stat.berkeley.edu/~ryantibs/statlearn-s23/lectures/calibration.pdf. Advanced Topics in Statistical Learning, Spring 2023.

van Renen, A., Horn, D., Pfeil, P., Vaidya, K. E., Dong, W., Narayanaswamy, M., Liu, Z., Saxena, G., Kipf, A., and Kraska, T. Why TPC is not enough: An analysis of the Amazon Redshift fleet. In *VLDB 2024*, 2024. URL https://www.amazon.science/publications/why-tpc-is-not-enough-an-analysis-of-the-amazon-redshift-f

Vito, S. D. D., Massera, E., Piga, M., Martinotto, L., and Francia, G. D. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2): 750–757, 2008. doi: 10.1016/j.snb.2007.09.060. UCI Machine Learning Repository - Air Quality dataset.

Wang, Z., Wen, Q., Zhang, C., Sun, L., Von Krannichfeldt, L., Pan, S., and Wang, Y. Benchmarks and custom package for energy forecasting. *arXiv preprint arXiv:2307.07191*, 2023.

Wilms, I. and Croux, C. Forecasting using sparse cointegration. *International Journal of Forecasting*, 32(4): 1256–1267, 2016.

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

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023.

Zhang, J., Wen, X., Zhang, Z., Zheng, S., Li, J., and Bian, J. ProbTS: Benchmarking point and distributional forecasting across diverse prediction horizons. *Advances in Neural Information Processing Systems*, 37:48045–48082, 2024.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021.

Zhou, J., Lu, X., Xiao, Y., Su, J., Lyu, J., Ma, Y., and Dou, D. SDWPF: A dataset for spatial dynamic wind power forecasting challenge at KDD Cup 2022. *arXiv preprint arXiv:2208.04360*, 2022.

A. Tasks

In total, fev-bench contains 100 time series forecasting tasks. In this section we provide the main statistics of these tasks together with citations to the sources of the datasets. For competition datasets we use their fixed forecast horizon H; for all others, H is set by a frequency–horizon mapping, except for a subset of hourly datasets where we use $H = 168$ to enable long-range forecasting. The number of evaluation windows W is then chosen to evenly split the series while ensuring that sufficient historical data is available for each forecast of length H. Dataset frequencies are reported using pandas frequency aliases (minu**T**ely, **H**ourly, **D**aily, **W**eekly, **M**onthly, **Q**uarterly, **Y**early).

The precise task definitions in YAML format are available under https://github.com/autogluon/fev. The datasets used for evaluation are hosted on Hugging Face https://huggingface.co/datasets/autogluon/fev_datasets.

A.1. GIFT-Eval

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
BizITObs - L2C	cloud	5T	288	20	31,968	1	7	0	0	0
BizITObs - L2C	cloud	H	24	20	2,664	1	7	0	0	0
ETT	energy	15T	96	20	69,680	2	7	0	0	0
ETT	energy	H	168	20	17,420	2	7	0	0	0
ETT	energy	D	28	20	724	2	7	0	0	0
ETT	energy	W	13	5	103	2	7	0	0	0
Hierarchical Sales	retail	D	28	10	1,825	118	1	0	0	0
Hierarchical Sales	retail	W	13	10	260	118	1	0	0	0
Hospital	healthcare	M	12	4	84	767	1	0	0	0
Jena Weather	nature	10T	144	20	52,704	1	21	0	0	0
Jena Weather	nature	D	28	11	366	1	21	0	0	0
Jena Weather	nature	H	24	20	8,784	1	21	0	0	0
Loop Seattle	mobility	D	28	10	365	323	1	0	0	0
Loop Seattle	mobility	5T	288	10	105,120	323	1	0	0	0
Loop Seattle	mobility	H	168	10	8,760	323	1	0	0	0
M-DENSE	mobility	D	28	10	730	30	1	0	0	0
M-DENSE	mobility	H	168	10	17,520	30	1	0	0	0
SZ Taxi	mobility	15T	96	10	2,976	156	1	0	0	0
SZ Taxi	mobility	H	168	2	744	156	1	0	0	0
Solar	energy	W	13	1	52	137	1	0	0	0
Solar	energy	D	28	10	365	137	1	0	0	0

Table 6. Tasks based on datasets coming from the GIFT-Eval corpus (Aksu et al., 2024).

The GIFT-Eval corpus (Aksu et al., 2024) contains various univariate and multivariate datasets, none of which provide covariates. The original datasets have been collected from sources such as Godahewa et al. (2021); Jiang et al. (2023); Mancuso et al. (2021); Wu et al. (2021); Palaskar et al. (2024).

A.2. Macroeconomic datasets

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
Australian Tourism	econ	Q	8	2	36	89	1	0	0	0
FRED-MD - CEE	econ	M	12	20	798	1	3	4	0	0
FRED-MD - Macro	econ	M	12	20	798	1	51	0	0	0
FRED-QD - CEE	econ	Q	8	20	266	1	3	4	0	0
FRED-QD - Macro	econ	Q	8	20	266	1	51	0	0	0
GVAR	econ	Q	8	10	178	33	6	3	0	0
US Consumption	econ	M	12	10	792	31	1	0	0	0
US Consumption	econ	Q	8	10	262	31	1	0	0	0
US Consumption	econ	Y	5	10	64	31	1	0	0	0
World CO2 Emissions	econ	Y	5	9	60	191	1	0	0	0
World Life Expectancy	econ	Y	5	10	74	237	1	0	0	0
World Tourism	econ	Y	5	2	21	178	1	0	0	0

Table 7. Tasks based on various macroeconomic datasets.

We consider various macroeconomic datasets such as GVAR (Mohaddes & Raissi, 2024), US Consumption (Wilms & Croux, 2016), Australian Tourism (Athanasopoulos et al., 2009), FRED-MD (McCracken & Ng, 2016), FRED-QD (McCracken & Ng, 2020), world CO2 emmissions (Kag, 2025a), life expectancy (Kag, 2025b) and global tourism (Kag, 2025d).

For each of FRED-MD and FRED-QD, we create two forecasting tasks. The first follows the CEE model (Christiano et al., 1999) and focuses on forecasting employment, inflation, and federal funds rate indicators. The second task involves jointly forecasting 51 core macroeconomic indicators. Note that we use the snapshot of FRED-MD corresponding to August 2025, which is different from FRED-MD snapshot used in Godahewa et al. (2021).

A.3. Energy datasets

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
ENTSO-e Load	energy	15T	96	20	175,292	6	1	0	3	0
ENTSO-e Load	energy	30T	96	20	87,645	6	1	0	3	0
ENTSO-e Load	energy	H	168	20	43,822	6	1	0	3	0
EPF-BE	energy	H	24	20	52,416	1	1	0	2	0
EPF-DE	energy	H	24	20	52,416	1	1	0	2	0
EPF-FR	energy	H	24	20	52,416	1	1	0	2	0
EPF-NP	energy	H	24	20	52,416	1	1	0	2	0
EPF-PJM	energy	H	24	20	52,416	1	1	0	2	0
ERCOT	energy	D	28	20	6,452	8	1	0	0	0
ERCOT	energy	H	168	20	154,872	8	1	0	0	0
ERCOT	energy	M	12	15	211	8	1	0	0	0
ERCOT	energy	W	13	20	921	8	1	0	0	0
GFC12	energy	H	168	10	39,414	11	1	0	1	0
GFC14	energy	H	168	20	17,520	1	1	0	1	0
GFC17	energy	H	168	20	17,544	8	1	0	1	0
Solar with Weather	energy	15T	96	20	198,600	1	1	2	7	0
Solar with Weather	energy	H	24	20	49,648	1	1	2	7	0

Table 8. Tasks based on datasets related to energy generation and consumption.

These datasets include electricity price forecasting (EPF) benchmark (Lago et al., 2021), ERCOT generation data (Ansari et al., 2024), ENTSO-e load data (Open Power System Data, 2020) with weather originating from `Renewables.ninja` (Staffell et al., 2023), and solar generation with weather (Kag, 2025c).

A.4. BOOMLET

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
BOOMLET - 1062	cloud	5T	288	20	16,384	1	21	0	0	0
BOOMLET - 1209	cloud	5T	288	20	16,384	1	53	0	0	0
BOOMLET - 1225	cloud	T	60	20	16,384	1	49	0	0	0
BOOMLET - 1230	cloud	5T	288	20	16,384	1	23	0	0	0
BOOMLET - 1282	cloud	T	60	20	16,384	1	35	0	0	0
BOOMLET - 1487	cloud	5T	288	20	16,384	1	54	0	0	0
BOOMLET - 1631	cloud	30T	96	20	10,463	1	40	0	0	0
BOOMLET - 1676	cloud	30T	96	20	10,463	1	100	0	0	0
BOOMLET - 1855	cloud	H	24	20	5,231	1	52	0	0	0
BOOMLET - 1975	cloud	H	24	20	5,231	1	75	0	0	0
BOOMLET - 2187	cloud	H	24	20	5,231	1	100	0	0	0
BOOMLET - 285	cloud	T	60	20	16,384	1	75	0	0	0
BOOMLET - 619	cloud	T	60	20	16,384	1	52	0	0	0
BOOMLET - 772	cloud	T	60	20	16,384	1	67	0	0	0
BOOMLET - 963	cloud	T	60	20	16,384	1	28	0	0	0

Table 9. Tasks based on datasets from BOOMLET (Cohen et al., 2025).

We include the multivariate observability datasets from the BOOMLET benchmark (Cohen et al., 2025). BOOMLET is a subset of the larger BOOM benchmark curated by the original authors. We additionally limit our attention to datasets with frequency of at least 1 minute to avoid including too many datasets from a single source to `fev-bench`.

A.5. Forecasting competitions

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
Favorita Store Sales	retail	M	12	2	54	1,579	1	1	1	6
Favorita Store Sales	retail	W	13	10	240	1,579	1	1	1	6
Favorita Store Sales	retail	D	28	10	1,688	1,579	1	1	2	6
Favorita Transactions	retail	M	12	2	54	51	1	1	0	5
Favorita Transactions	retail	W	13	10	240	51	1	1	0	5
Favorita Transactions	retail	D	28	10	1,688	51	1	1	1	5
KDD Cup 2022	energy	D	14	10	243	134	1	9	0	0
KDD Cup 2022	energy	10T	288	10	35,279	134	1	9	0	0
KDD Cup 2022	energy	30T	96	10	11,758	134	1	9	0	0
M5	retail	M	12	1	58	30,490	1	0	8	5
M5	retail	W	13	1	257	30,490	1	0	8	5
M5	retail	D	28	1	1,810	30,490	1	0	8	5
Restaurant	retail	D	28	8	296	817	1	0	0	4
Rohlik Orders	retail	W	8	5	170	7	1	9	4	0
Rohlik Orders	retail	D	61	5	1,197	7	1	9	4	0
Rohlik Sales	retail	W	8	1	150	5,243	1	1	13	7
Rohlik Sales	retail	D	14	1	1,046	5,390	1	1	13	7
Rossmann	retail	W	13	8	133	1,115	1	1	4	10
Rossmann	retail	D	48	10	942	1,115	1	1	5	10
Walmart	retail	W	39	1	143	2,936	1	0	10	4

Table 10. Tasks based on datasets coming from various forecasting competitions

We use datasets from forecasting competitions held on `kaggle.com` (Bojer & Meldgaard, 2021). These include Favorita store sales & transactions (Kag, 2020), the M5 competition (Makridakis et al., 2022), restaurant visitor & reservation (Kag, 2017), Rossmann (Kag, 2015), Walmart (Kag, 2014), and Rohlik (Roh, 2025) store sales forecasting competitions. We also consider the KDD Cup 2022 dataset where the goal is to predict wind power generation (Zhou et al., 2022), and the Global Energy Forecasting Competitions held in 2012, 2014 and 2017 (Hong et al., 2014).

A.6. Other sources

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
ECDC ILI	healthcare	W	13	10	201	25	1	0	0	0
Hermes	retail	W	52	1	261	10,000	1	0	1	2
Hospital Admissions	healthcare	D	28	20	1,731	8	1	0	0	0
Hospital Admissions	healthcare	W	13	16	246	8	1	0	0	0
Redset	cloud	5T	288	10	25,920	118	1	0	0	1
Redset	cloud	15T	96	10	8,640	126	1	0	0	1
Redset	cloud	H	24	10	2,160	138	1	0	0	1
UCI Air Quality	nature	H	168	20	9,357	1	4	0	3	0
UCI Air Quality	nature	D	28	11	389	1	4	0	3	0
UK COVID - Nation - Cumulative	healthcare	D	28	20	729	4	3	5	0	0
UK COVID - Nation - Cumulative	healthcare	W	8	4	105	4	3	5	0	0
UK COVID - Nation - New	healthcare	D	28	20	729	4	3	5	0	0
UK COVID - Nation - New	healthcare	W	8	4	105	4	3	5	0	0
UK COVID - UTLA - Cumulative	healthcare	W	13	5	104	214	1	0	0	0
UK COVID - UTLA - New	healthcare	D	28	10	721	214	1	0	0	0

Table 11. Tasks based on datasets collected from other sources.

We also include datasets from the following miscellaneous sources.

- Influenza-like-illness cases collected by the European Centre for Disease Prevention and Control (ECDC, 2025).
- Fashion trend data from Hermes (David et al., 2022).
- Hospital admissions data from Riyadh (of Health Affairs & Ministry of Health, 2025).
- Query counts for Amazon Redshift database servers (van Renen et al., 2024).

- Solar energy generation with corresponding weather covariates (Kag, 2025c).
- Air quality measurements in an Italian city with accompanying weather data (Vito et al., 2008).
- COVID-19 cases, hospital admissions, and deaths in the United Kingdom at different administrative levels (Kag, 2025e).

B. Models

We evaluate seven pretrained models on fev-bench, whose key properties are summarized in Table 12. Most are decoder-only transformers, except TiRex, a decoder-only xLSTM model, and Chronos-Bolt (Base), an encoder-decoder transformer. All models except TabPFN-TS process non-overlapping patches of time series rather than individual observations. Toto, TabPFN-TS, and Sundial produce sample forecasts, while the others generate quantiles on a fixed grid. Toto is the only model that natively supports multivariate targets. TabPFN-TS is the only model that accepts known covariates, though we did not provide past covariates since they consistently degraded its accuracy.

For all pretrained models, we keep hyperparameters at their default values unless specified in the table. For Toto, we reduced the samples per batch and batch size to avoid out-of-memory errors on large multivariate datasets. We run all pretrained models on a g5.2xlarge AWS instance with a single A10G GPU (24GB GPU RAM, 32GB RAM), using PyTorch 2.6 with CUDA 12.6 via AWS Deep Learning Containers. The version in the table below refers to either the PyPI package version, or the date on which the official repository was cloned if no PyPI package is provided by the authors.

Model Name	Hugging Face ID	Batch size	Version	Max Context	Hyperparameters
TiRex	`NX-AI/TiRex`	512	2025-09-01	2048	-
Toto	`Datadog/Toto-Open-Base-1.0`	24	2025-08-01	4096	{`samples_per_batch: 8`}
Moirai 2.0	`Salesforce/moirai-2.0-R-small`	128	2025-08-10	4000	-
Chronos-Bolt	`amazon/chronos-bolt-base`	256	1.5.3	2048	-
TimesFM 2.5	`google/timesfm-2.5-200m-pytorch`	256	2025-09-28	16000	-
TabPFN-TS	`Prior-Labs/TabPFN-v2-reg`	-	1.0.3	5000	{`checkpoint: '2noar4o2'`}
Sundial	`thuml/sundial-base-128m`	512	2025-09-01	2880	-

Table 12. Properties of different pretrained time series forecasting models.

We also include statistical baselines from the StatsForecast library (Garza et al., 2022), such as AutoETS, AutoARIMA, AutoTheta, as well as the SCUM ensemble (Petropoulos & Svetunkov, 2020). To avoid long runtimes, we truncate the context length of statistical models to 1000 steps and set the maximum season length to 200 for AutoETS, AutoTheta, AutoARIMA and SCUM Ensemble. For evaluation of statistical models we used StatsForecast `v2.0.1` and ran the experiments on the `m6i.4xlarge` AWS instances with 16 vCPU cores and 64GB RAM.

C. Extended discussion of aggregation methods

This appendix clarifies how average win rates W_j (Equation (6)) relate to other aggregation methods commonly used in benchmarking, such as average ranks (Aksu et al., 2024; Ansari et al., 2024) and Bradley–Terry ("Elo") scores (Erickson et al., 2025). We show that all three induce the same ordering of models.

C.1. Average win rate and average rank are equivalent

For a given task r and model j, let

$$M_{\text{lower}} = \sum_{\substack{k=1 \\ k \neq j}}^{M} \mathbb{1}(E_{rk} < E_{rj}), \qquad M_{\text{tied}} = \sum_{\substack{k=1 \\ k \neq j}}^{M} \mathbb{1}(E_{rk} = E_{rj}).$$

The midrank of j on task r is

$$\text{rank}_{rj} \;=\; 1 + M_{\text{lower}} + \tfrac{1}{2} M_{\text{tied}}.$$

Its contribution to W_j equals

$$\frac{1}{M-1}\sum_{k\neq j}\Big(\mathbb{1}(E_{rj}<E_{rk})+\tfrac{1}{2}\,\mathbb{1}(E_{rj}=E_{rk})\Big)=1-\frac{\text{rank}_{rj}-1}{M-1}.$$

Averaging over tasks gives

$$W_j=1-\frac{\overline{\text{rank}}_j-1}{M-1},\qquad \overline{\text{rank}}_j=\tfrac{1}{R}\sum_{r=1}^{R}\text{rank}_{rj}\,.$$

Thus, W_j and $\overline{\text{rank}}_j$ are affinely equivalent and induce the same ordering of models (with lower rank $\leftrightarrow$ higher win rate).

C.2. Average win rate and Bradley–Terry (Elo) scores result in the same ranking

The Bradley–Terry (BT) model, also known as Elo rating (Chiang et al., 2024), provides a parametric way to convert pairwise win rates into latent skill scores. In contrast to the nonparametric average win rate W_j, the BT model assumes that each model j has an underlying skill parameter $\theta_j\in\mathbb{R}$, and that the probability of j outperforming k follows a logistic link:

$$\Pr(E_{rj}<E_{rk})\;=\;\sigma\big(\lambda(\theta_j-\theta_k)\big),\qquad \sigma(x)=\tfrac{1}{1+e^{-x}},\;\;\lambda>0.$$

Here λ is a scaling constant (in Elo, $\lambda=\ln 10/400$). The parameters $\theta=(\theta_1,\ldots,\theta_M)$ are estimated by maximum likelihood:

$$\hat{\theta}\;\in\;\arg\max_{\theta\in\mathbb{R}^M}\sum_{j<m}\Big[W_{jm}\log\sigma(\lambda(\theta_j-\theta_m))+(1-W_{jm})\log\sigma(\lambda(\theta_m-\theta_j))\Big].$$

Typically some identifiability constraint is added, such as fixing $\theta_\beta=1000$ for a chosen baseline β.

Proposition C.1. *Suppose all M models are compared on the same R tasks, with pairwise win rates W_{jk} (Equation (8)) and average win rates $W_j=\frac{1}{M-1}\sum_{k\neq j}W_{jk}$ (Equation (6)). At the BT MLE with scale $\lambda>0$,*

$$\theta_j>\theta_k\iff W_j>W_k,\qquad \theta_j=\theta_k\iff W_j=W_k.$$

Proof. Differentiating the log-likelihood gives the score equations

$$\frac{\partial\ell}{\partial\theta_j}=\lambda\sum_{m\neq j}\Big(W_{jm}-\sigma(\lambda(\theta_j-\theta_m))\Big)=0.$$

Subtracting the equations for j and k yields

$$(M-1)(W_j-W_k)=\sum_{m\neq j,k}\big[\sigma(\lambda(\theta_j-\theta_m))-\sigma(\lambda(\theta_k-\theta_m))\big]+\big[\sigma(\lambda\Delta)-\sigma(-\lambda\Delta)\big],$$

where $\Delta=\theta_j-\theta_k$. Each term on the right is strictly increasing in Δ, so the whole expression has the same sign as Δ. Thus $\text{sign}(W_j-W_k)=\text{sign}(\theta_j-\theta_k)$, with equality iff $\Delta=0$. Strict concavity of the BT log-likelihood ensures uniqueness of the solution up to translation. ☐

Conclusion. Average win rate W_j and BT/Elo scores θ_j induce the same ordering of models, with higher win rate corresponding to higher Elo score.

D. Extended results

Tabular results for each dataset-task combination, updated marginal and pairwise results in an interactive foramt are available under https://huggingface.co/spaces/autogluon/fev-leaderboard.

The failures for Stat. Ensemble, AutoARIMA, and AutoETS models in the tables below correspond to the models exceeding the 6 hour time limit for a single task. TabPFN-TS failed on 2 tasks due to out of memory errors.

D.1. Marginal performance

Model	Avg. win rate (%)	Skill score (%)	Median runtime (s)	Leakage (%)	# failures
TiRex	86.7	42.6	1.4	1	0
TimesFM-2.5	82.1	42.3	117.6	8	0
Toto-1.0	73.8	40.7	90.7	8	0
Moirai-2.0	68.8	39.3	2.5	28	0
Chronos-Bolt	68.8	38.9	1.0	0	0
TabPFN-TS	66.9	39.6	305.5	0	2
Sundial-Base	49.2	33.4	35.6	1	0
Stat. Ensemble	48.7	20.2	690.6	0	11
AutoARIMA	43.5	20.6	186.8	0	10
AutoETS	35.8	-26.8	17.0	0	3
AutoTheta	29.2	5.5	9.3	0	0
Seasonal Naive	21.7	0.0	2.3	0	0
Naive	14.9	-45.4	2.2	0	0
Drift	9.9	-45.8	2.2	0	0

Table 13. Marginal probabilistic forecasting performance of all models (according to the SQL metric) on the full fev-bench benchmark. The reported metrics are defined in Sections 4.1 and 7.1.

Model	Avg. win rate (%)	Skill score (%)	Median runtime (s)	Leakage (%)	# failures
TiRex	80.5	30.0	1.4	1	0
TimesFM-2.5	79.9	30.3	117.6	8	0
Toto-1.0	69.9	28.2	90.7	8	0
Moirai-2.0	65.2	27.3	2.5	28	0
Chronos-Bolt	64.8	26.5	1.0	0	0
TabPFN-TS	62.0	27.6	305.5	0	2
Sundial-Base	56.7	24.7	35.6	1	0
Stat. Ensemble	51.0	15.7	690.6	0	11
AutoARIMA	39.0	11.2	186.8	0	10
AutoTheta	37.1	11.0	9.3	0	0
AutoETS	34.9	2.3	17.0	0	3
Seasonal Naive	22.3	0.0	2.3	0	0
Naive	20.6	-16.7	2.2	0	0
Drift	16.0	-18.1	2.2	0	0

Table 14. Marginal point forecasting performance of all models (according to the MASE metric) on the full fev-bench benchmark. The reported metrics are defined in Sections 4.1 and 7.1.

D.2. Pairwise comparison

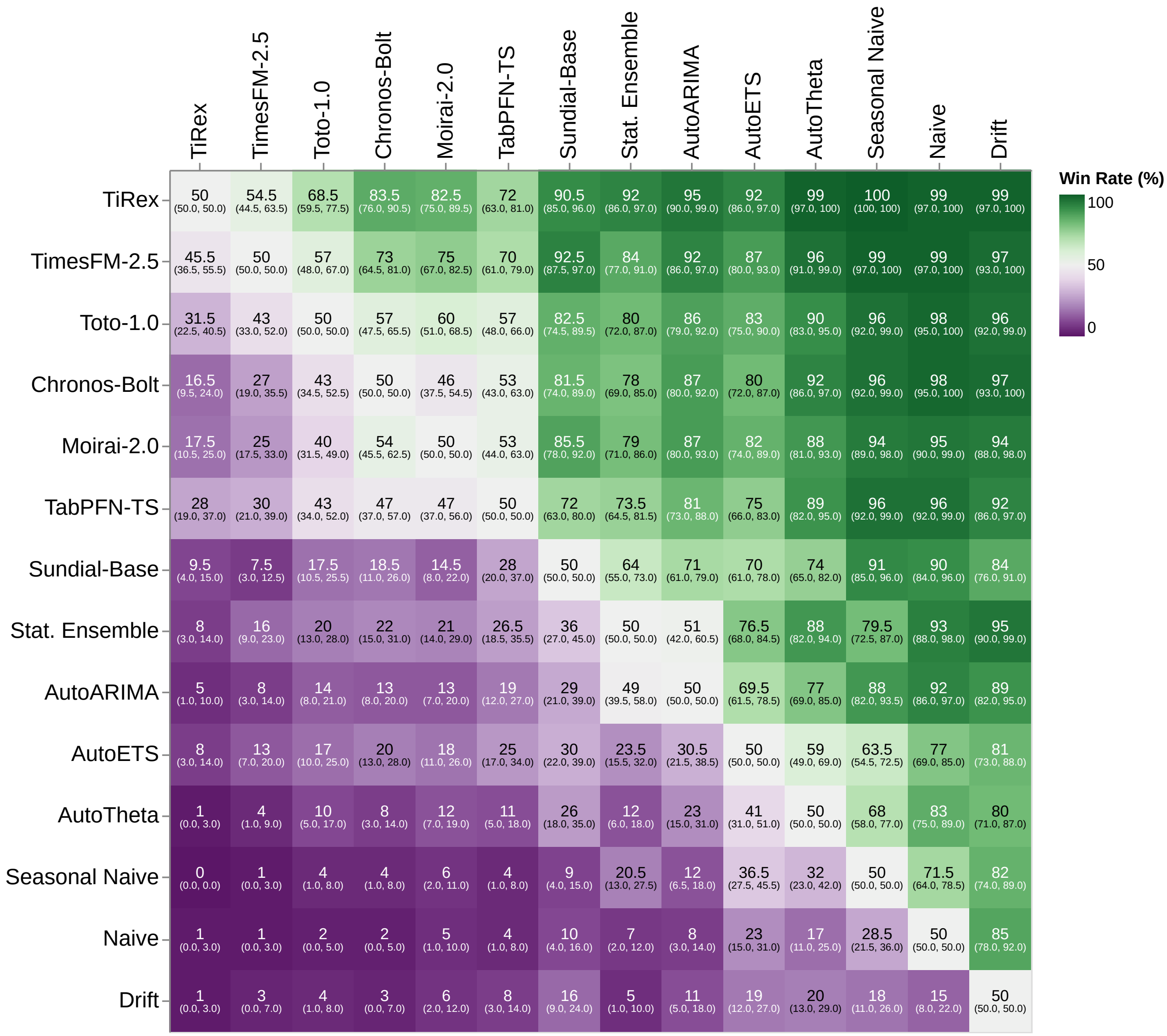


Figure 4. Pairwise win rates W_{jk} (Equation (8)) of all models against each other under the scaled quantile loss (SQL) metric on fev-bench, with 95% confidence intervals obtained via bootstrapping. Higher values are better.

	TiRex	TimesFM-2.5	Toto-1.0	TabPFN-TS	Moirai-2.0	Chronos-Bolt	Sundial-Base	AutoARIMA	Stat. Ensemble	AutoTheta	Seasonal Naive	AutoETS	Naive	Drift
TiRex	0 (0.0, 0.0)	0.5 (-1.2, 2.1)	3.1 (0.9, 5.2)	4.9 (0.5, 9.2)	5.3 (3.6, 7.2)	6 (4.2, 8.0)	13.7 (10.7, 16.9)	27.7 (23.7, 32.0)	28.1 (23.2, 32.8)	39.3 (33.9, 44.5)	42.6 (37.9, 47.9)	52.4 (41.7, 62.6)	60.5 (54.7, 65.3)	60.6 (54.5, 65.5)
TimesFM-2.5	−0.5 (-2.1, 1.2)	0 (0.0, 0.0)	2.6 (0.3, 5.0)	4.5 (-0.3, 8.8)	4.9 (2.8, 7.1)	5.6 (3.5, 7.7)	13.3 (10.5, 16.2)	27.4 (23.3, 32.0)	27.7 (22.4, 32.7)	39 (33.2, 44.7)	42.3 (37.7, 47.6)	52.2 (41.5, 62.6)	60.3 (54.5, 65.1)	60.4 (54.4, 65.4)
Toto-1.0	−3.2 (-5.4, -1.0)	−2.7 (-5.3, -0.3)	0 (0.0, 0.0)	1.9 (-3.3, 7.0)	2.3 (0.2, 4.5)	3 (0.1, 5.5)	11 (7.4, 14.7)	25.4 (20.7, 30.5)	25.8 (20.0, 30.9)	37.3 (31.0, 43.1)	40.7 (35.4, 46.5)	51 (40.2, 61.4)	59.2 (53.2, 64.2)	59.4 (53.0, 64.5)
TabPFN-TS	−5.2 (-10.1, -0.5)	−4.7 (-9.7, 0.3)	−2 (-7.6, 3.2)	0 (0.0, 0.0)	0.4 (-4.4, 5.2)	1.1 (-3.2, 5.5)	9.3 (5.0, 13.5)	23.9 (18.9, 29.2)	24.3 (19.0, 29.7)	36.1 (30.2, 42.2)	39.6 (34.6, 44.7)	50.3 (38.7, 61.1)	58.4 (52.6, 63.6)	58.6 (52.1, 63.9)
Moirai-2.0	−5.6 (-7.7, -3.8)	−5.2 (-7.6, -2.9)	−2.4 (-4.7, -0.2)	−0.4 (-5.5, 4.2)	0 (0.0, 0.0)	0.7 (-1.6, 2.9)	8.9 (5.4, 12.3)	23.6 (18.9, 28.6)	24 (18.7, 29.2)	35.8 (29.9, 41.5)	39.3 (34.3, 44.8)	50 (38.2, 60.9)	58.3 (52.3, 63.4)	58.4 (51.9, 63.6)
Chronos-Bolt	−6.4 (-8.8, -4.4)	−5.9 (-8.3, -3.7)	−3.1 (-5.9, -0.1)	−1.1 (-5.9, 3.1)	−0.7 (-3.0, 1.6)	0 (0.0, 0.0)	8.2 (4.5, 11.6)	23.1 (18.8, 27.7)	23.5 (18.4, 28.4)	35.4 (30.0, 40.9)	38.9 (34.2, 44.1)	49.6 (37.9, 60.5)	58 (52.4, 62.8)	58.1 (52.0, 63.0)
Sundial-Base	−15.9 (-20.3, -11.9)	−15.4 (-19.3, -11.8)	−12.4 (-17.2, -7.9)	−10.2 (-15.5, -5.3)	−9.7 (-14.0, -5.7)	−9 (-13.2, -4.7)	0 (0.0, 0.0)	16.2 (10.2, 22.0)	16.6 (9.6, 23.0)	29.6 (22.6, 36.4)	33.4 (28.1, 39.2)	45.3 (32.3, 57.6)	54.2 (47.3, 59.8)	54.3 (47.0, 60.2)
AutoARIMA	−38.3 (-47.0, -31.1)	−37.7 (-47.1, -30.3)	−34.1 (-43.9, -26.1)	−31.5 (-41.2, -23.3)	−30.9 (-40.1, -23.3)	−30 (-38.3, -23.2)	−19.3 (-28.2, -11.4)	0 (0.0, 0.0)	0.5 (-4.3, 4.7)	16 (9.9, 21.6)	20.6 (15.4, 26.3)	35.9 (21.1, 49.6)	45.4 (38.8, 51.2)	45.5 (38.4, 51.6)
Stat. Ensemble	−39 (-48.8, -30.2)	−38.4 (-48.7, -28.8)	−34.7 (-44.8, -25.0)	−32.2 (-42.3, -23.4)	−31.6 (-41.2, -23.0)	−30.7 (-39.6, -22.6)	−19.9 (-29.8, -10.7)	−0.5 (-4.9, 4.1)	0 (0.0, 0.0)	15.6 (10.8, 20.8)	20.2 (14.8, 25.7)	36.5 (22.0, 50.1)	45.1 (39.5, 50.3)	45.2 (39.4, 50.6)
AutoTheta	−64.6 (-80.3, -51.3)	−63.9 (-80.9, -49.7)	−59.6 (-75.9, -44.9)	−56.5 (-73.1, -43.3)	−55.8 (-71.1, -42.7)	−54.7 (-69.1, -42.9)	−42 (-57.2, -29.2)	−19 (-27.6, -11.0)	−18.4 (-26.3, -12.1)	0 (0.0, 0.0)	5.5 (-3.8, 13.5)	25.7 (7.5, 41.5)	35 (29.3, 40.3)	35.1 (28.8, 40.8)
Seasonal Naive	−74.1 (-91.9, -61.1)	−73.3 (-90.8, -60.5)	−68.8 (-86.7, -54.8)	−65.5 (-80.9, -52.9)	−64.8 (-81.3, -52.2)	−63.6 (-78.8, -52.0)	−50.2 (-64.5, -39.0)	−25.9 (-35.6, -18.2)	−25.3 (-34.7, -17.4)	−5.8 (-15.6, 3.7)	0 (0.0, 0.0)	21.1 (1.3, 39.2)	31.2 (22.9, 38.6)	31.4 (22.2, 39.6)
AutoETS	−110.2 (-167, -71.4)	−109.1 (-167, -70.8)	−104.1 (-159, -67.2)	−101.1 (-157, -63.2)	−99.9 (-156, -61.9)	−98.3 (-153, -60.9)	−82.8 (-136, -47.8)	−56 (-98.3, -26.8)	−57.5 (-100, -28.2)	−34.5 (-70.9, -8.1)	−26.8 (-64.4, -1.3)	0 (0.0, 0.0)	10.4 (-17.3, 29.7)	10.3 (-16.1, 29.7)
Naive	−153.2 (-188, -121)	−152 (-187, -120)	−145.4 (-179, -114)	−140.7 (-175, -111)	−139.7 (-173, -110)	−137.9 (-169, -110)	−118.4 (-149, -89.8)	−83 (-105, -63.5)	−82.1 (-101, -65.3)	−53.8 (-67.6, -41.4)	−45.4 (-62.8, -29.8)	−11.6 (-42.2, 14.8)	0 (0.0, 0.0)	0.3 (-2.0, 2.2)
Drift	−153.9 (-190, -120)	−152.7 (-189, -119)	−146 (-182, -113)	−141.3 (-177, -109)	−140.3 (-175, -108)	−138.6 (-170, -108)	−119 (-151, -88.6)	−83.5 (-106, -62.3)	−82.6 (-102, -65.0)	−54.2 (-69.0, -40.4)	−45.8 (-65.6, -28.5)	−11.5 (-42.3, 13.9)	−0.3 (-2.3, 2.0)	0 (0.0, 0.0)

Skill Score (%): 20, 0, −20

Figure 5. Pairwise skill scores S_{jk} (Equation (9)) of all models against each other under the scaled quantile loss (SQL) metric on `fev-bench`, with 95% confidence intervals obtained via bootstrapping. Higher values are better. Note that pairwise skill score is not symmetric, $S_{jk} \neq S_{kj}$.

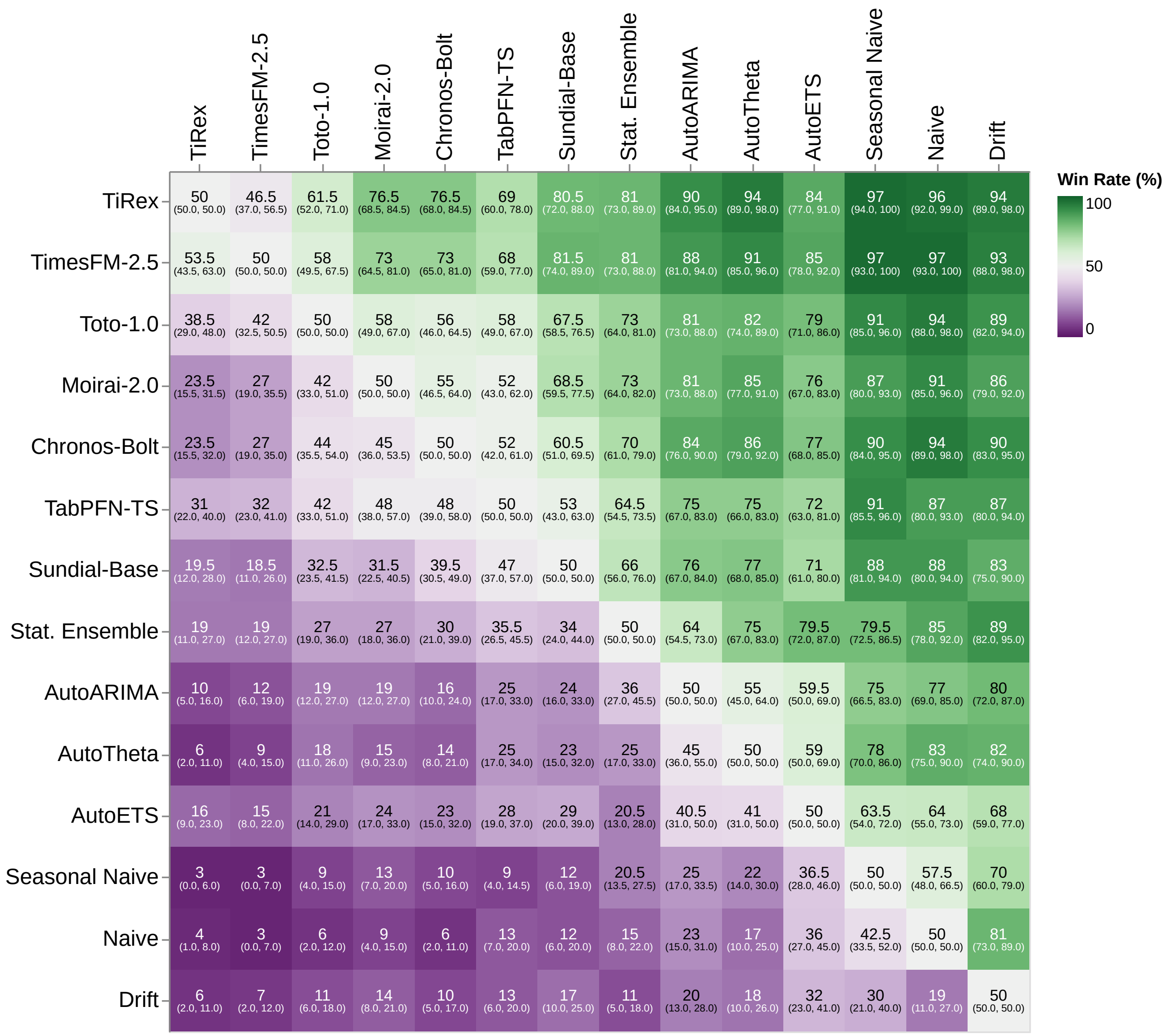


Figure 6. Pairwise win rates W_{jk} (Equation (8)) of all models against each other under the mean absolute scaled error (MASE) metric on fev-bench, with 95% confidence intervals obtained via bootstrapping. Higher values are better.

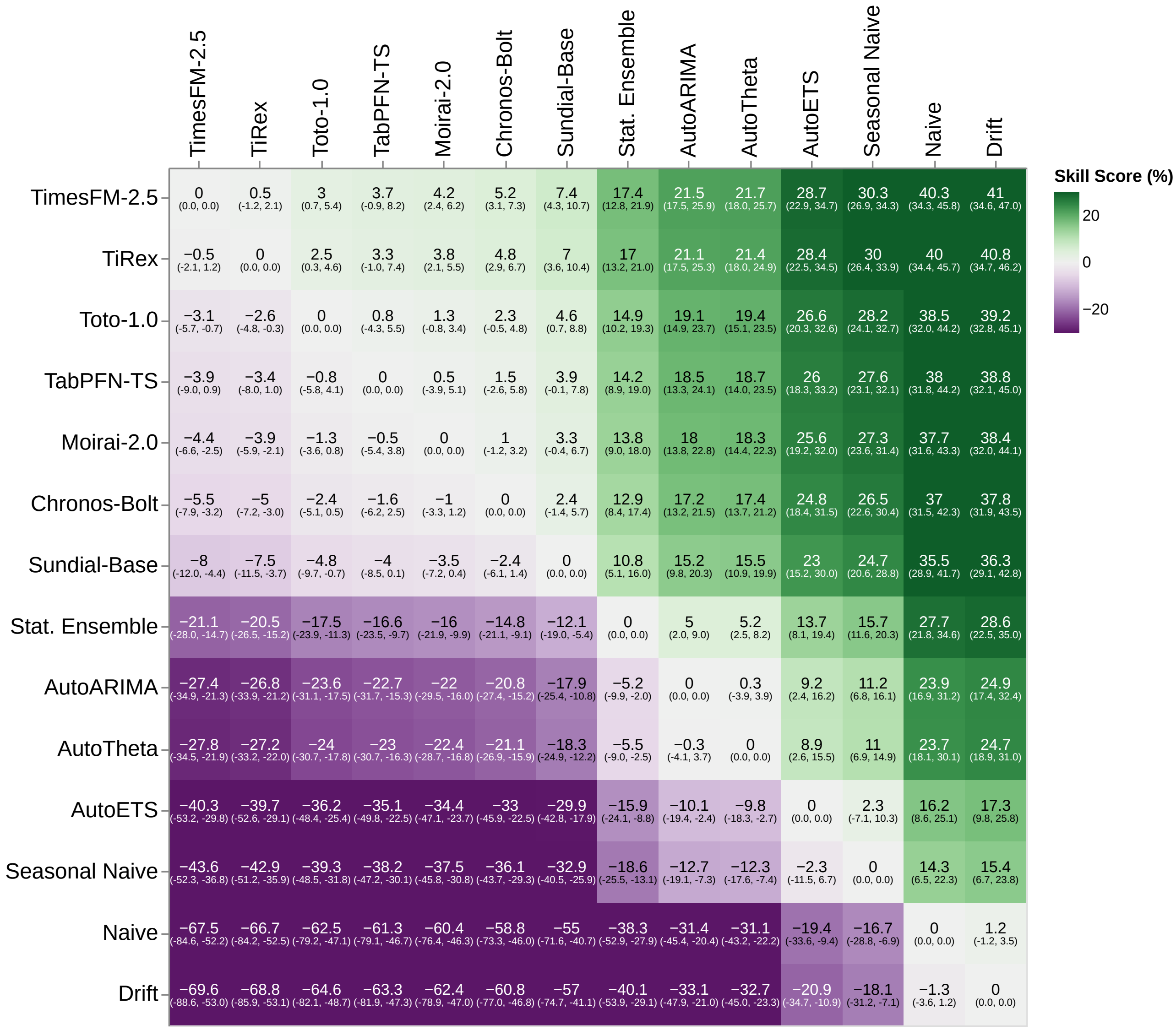


Figure 7. Pairwise skill scores S_{jk} (Equation (9)) of all models against each other under the mean absolute scaled error (MASE) metric on `fev-bench`, with 95% confidence intervals obtained via bootstrapping. Higher values are better. Note that pairwise skill score is not symmetric, $S_{jk} \neq S_{kj}$.

E. fev-bench-mini: A representative subset of fev-bench

E.1. Tasks

fev-bench-mini consists of 20 tasks that are representative of the full 100 tasks comprising fev-bench (Section A).

Task	Domain	Freq.	H	W	Median length	# series	# targets	# past cov.	# known cov.	# static cov.
BOOMLET - 1282	cloud	T	60	20	16,384	1	35	0	0	0
BOOMLET - 1676	cloud	30T	96	20	10,463	1	100	0	0	0
BOOMLET - 619	cloud	T	60	20	16,384	1	52	0	0	0
BizITObs - L2C	cloud	5T	288	20	31,968	1	7	0	0	0
EPF-NP	energy	H	24	20	52,416	1	1	0	2	0
ETT	energy	15T	96	20	69,680	2	7	0	0	0
ETT	energy	H	168	20	17,420	2	7	0	0	0
GFC14	energy	H	168	20	17,520	1	1	0	1	0
Hospital Admissions	healthcare	W	13	16	246	8	1	0	0	0
Hospital Admissions	healthcare	D	28	20	1,731	8	1	0	0	0
Jena Weather	nature	H	24	20	8,784	1	21	0	0	0
M-DENSE	mobility	D	28	10	730	30	1	0	0	0
Rohlik Orders	retail	D	61	5	1,197	7	1	9	4	0
Rossmann	retail	W	13	8	133	1,115	1	1	4	10
Rossmann	retail	D	48	10	942	1,115	1	1	5	10
Solar with Weather	energy	H	24	20	49,648	1	1	2	7	0
UCI Air Quality	nature	H	168	20	9,357	1	4	0	3	0
UK COVID - Nation - Cumulative	healthcare	D	28	20	729	4	3	5	0	0
US Consumption	econ	Y	5	10	64	31	1	0	0	0
World CO2 Emissions	econ	Y	5	9	60	191	1	0	0	0

Table 15. Tasks included in fev-bench-mini.

E.2. Evaluation results

For completeness, we provide the evaluation results on `fev-bench-mini`. The overall ranking of the models, win rates and skill scores align with the scores on the full benchmark reported in Section D.

Model	Avg. win rate (%)	Skill score (%)	Median runtime (s)	Leakage (%)	# failures
TiRex	84.6	43.4	1.3	0	0
TimesFM-2.5	79.0	44.0	76.1	5	0
Toto-1.0	77.9	43.0	66.3	5	0
TabPFN-TS	71.2	46.2	275.0	0	0
Moirai-2.0	69.6	41.7	1.6	30	0
Chronos-Bolt	66.2	40.4	1.1	0	0
Sundial-Base	53.1	37.6	23.1	0	0
Stat. Ensemble	48.8	27.3	726.8	0	2
AutoARIMA	47.3	30.6	226.0	0	2
AutoETS	35.0	-9.4	14.8	0	0
AutoTheta	27.7	8.5	8.1	0	0
Seasonal Naive	17.7	0.0	2.3	0	0
Naive	13.8	-36.9	2.2	0	0
Drift	8.1	-36.0	2.2	0	0

Table 16. Marginal probabilistic forecasting performance of all models (according to the SQL metric) on the `fev-bench-mini` benchmark. The reported metrics are defined in Sections 4.1 and 7.1.

Model	Avg. win rate (%)	Skill score (%)	Median runtime (s)	Leakage (%)	# failures
TiRex	78.8	31.4	1.3	0	0
TimesFM-2.5	77.9	32.3	76.1	5	0
Toto-1.0	75.6	30.9	66.3	5	0
TabPFN-TS	66.9	35.1	275.0	0	0
Moirai-2.0	66.2	30.1	1.6	30	0
Chronos-Bolt	64.2	28.4	1.1	0	0
Sundial-Base	61.9	29.8	23.1	0	0
Stat. Ensemble	50.8	21.1	726.8	0	2
AutoARIMA	44.2	20.4	226.0	0	2
AutoTheta	35.4	14.8	8.1	0	0
AutoETS	33.8	2.3	14.8	0	0
Seasonal Naive	16.2	0.0	2.3	0	0
Naive	15.0	-17.4	2.2	0	0
Drift	13.1	-17.2	2.2	0	0

Table 17. Marginal point forecasting performance of all models (according to the MASE metric) on the `fev-bench-mini` benchmark. The reported metrics are defined in Sections 4.1 and 7.1.