A FORECASTING ODYSSEY: CHARTING THE BILLION-SCALE LANDSCAPE OF TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series forecasting—using historical patterns to predict future outcomes—has gained significant attention due to its broad practical impact nowadays. Despite decades of methodological advances and benchmark development, existing evaluation frameworks still exhibit critical limitations: (i) many studies rely on homogeneous or imbalanced datasets with limited cross-domain diversity and quality assurance; (ii) model coverage is often limited, overlooking simple yet effective baselines; and (iii) insufficient statistical and domain-specific analyses lead to superficial conclusions. To address these limitations, we introduce `TimeFCST`, a *billion-scale* benchmark with systematic quality assurance that evaluates 100 forecasting methods, roughly 5$\times$ more than prior studies, across statistical, machine learning, deep learning, and foundation models. Beyond aggregate leaderboards, we provide, to our knowledge for the first time, rigorous statistical validation of comparative findings over large-scale experiments and a failure-profiling framework spanning six error patterns. Together, these answer three questions: *What works, when does it work, and how does it fail?* Our evaluation reveals: (i) modern forecasting models lead aggregate rankings, yet statistically significant gains over well-tuned simple classical baselines emerge only among leading foundation models, especially on UTS; (ii) model effectiveness varies across domains and settings, with simple classical methods matching or even outperforming modern models in domains such as nature, retail and economics; (iii) failure profiles reveal distinct error patterns hidden by aggregate metrics: deep learning and foundation models exhibit more phase and localized errors, with foundation models showing stronger underprediction bias; and (iv) additionally, we identify saturated datasets that obscure methodological progress and show that broad, diverse benchmark coverage reveals otherwise hidden, statistically significant differences among models. In summary, `TimeFCST` charts the evolving landscape of time-series forecasting, offering insights into recent advances and directions for future research.

1 INTRODUCTION

The proliferation of sensor networks and data acquisition systems has produced vast amounts of high-fidelity signal data. These ordered, real-valued observations are commonly referred to as *time series* in scientific and industrial applications (Jeung et al., 2010; Krishnan et al., 2019). The effective extraction and analysis of the temporal information is crucial for uncovering latent patterns and making strategic decisions. Consequently, the field has experienced a substantial expansion in downstream tasks, ranging from classification (Schäfer, 2015; Middlehurst et al., 2024) and clustering (Liao, 2005; Paparrizos & Gravano, 2015; Aghabozorgi et al., 2015) to anomaly detection (Paparrizos et al., 2022; Liu & Paparrizos, 2024; Senin et al., 2014; Huang et al., 2026), similarity search (Lin et al., 2003; Schäfer & Högqvist, 2012; d'Hondt et al., 2025), and forecasting tasks (Nikolopoulos & Petropoulos, 2018; Nie et al., 2023; Liu et al., 2024a;b; 2025a; Zhao et al., 2023; Spiliotis & Petropoulos, 2024). Among these tasks, forecasting—predicting future values from historical observations—is fundamental to many real-world applications. For example, weather forecasts inform planning in agriculture and aviation (Box et al., 2015), while demand forecasts help enterprises align inventory with market needs and reduce supply-chain costs (Makridakis et al., 2022).

Over the past two decades, time-series forecasting has undergone a substantial methodological development. Traditional approaches, including statistical and machine learning (ML) methods,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: **Comparison with existing time-series forecasting benchmarks**. TimeFCST provides (i) a billion-scale dataset composition across domains, (ii) broader model category and metric coverage, (iii) reliable data quality assurance, and (iv) comprehensive evaluation protocol.

Benchmark	Dataset				Model					Eval	Quality Assurance			Benchmark Protocol		
	#Obs.	UTS	MTS	Total	Stat	ML	DL	FM	Total	Metric	Data Cleaning	Inspection	Leakage Report	HPO Set	Stats. Test	Failure Analysis
Monash (Godahewa et al., 2021)	182M	34	16	50	5	2	5	0	12	4	✓	✓	✗	✗	✗	✗
TSLib (Wang et al., 2025)	30M	0	14	14	0	0	13	0	13	2	✗	✗	✗	✗	✗	✗
TFB (Qiu et al., 2025a)	198M	28	25	53	3	3	15	0	21	8	✗	✗	✗	✗	✗	✗
GIFT-Eval (Aksu et al., 2024)	157M	35	20	55	5	0	8	4	17	2	✗	✗	✓	✗	✗	✗
fev-bench (Shchur et al., 2026)	761M	65	35	100	6	0	0	8	14	2	✗	✗	✓	✓	✗	✗
TSFM-bench (Li et al., 2025)	40M	0	21	21	0	0	7	14	21	2	✗	✗	✗	✗	✗	✗
TimeFCST (Ours)	**1B**	**64**	**64**	**128**	**18**	**16**	**47**	**19**	**100**	**8**	✓	✓	✓	✓	✓	✓

typically rely on structured assumptions or feature engineering to model temporal patterns (Box et al., 2015; Breiman, 2001). More recently, deep learning (DL) models have captured complex nonlinear dynamics through diverse architectures (Nie et al., 2023; Wu et al., 2023; Qiu et al., 2025b), while foundation models (FM) have demonstrated promising zero-shot capabilities through pre-training on large-scale, cross-domain time-series corpora (Ansari et al., 2025; Das et al., 2024b; Liu et al., 2026). Alongside these advances, numerous benchmarks have been developed to standardize evaluation across different scales and domains (Qiu et al., 2025a; Aksu et al., 2024; Shchur et al., 2026; Godahewa et al., 2021; Li et al., 2025). Nevertheless, existing evaluation practices remain limited by: (i) constrained or imbalanced dataset collections with insufficient quality assurance; (ii) incomplete coverage of model families, particularly overlooking simple yet effective classical baselines; and (iii) an emphasis on aggregate performance metrics with limited statistical and domain-specific analysis. These limitations obscure the relative strengths and weaknesses of modern forecasting methods, motivating a more comprehensive, balanced, and rigorous evaluation standard.

To address these challenges, we introduce TimeFCST, a *billion*-scale benchmark for the comprehensive and rigorous evaluation of time-series forecasting models. To ensure broad and heterogeneous data coverage, TimeFCST incorporates 128 datasets spanning seven major domains (Figure 1), balanced between 64 univariate (UTS) and 64 multivariate (MTS) datasets with diverse temporal patterns (Figures 1 and 11 in Appendix) and quality assurance. Based on this benchmark, we provide a standardized evaluation of 100 forecasting models, roughly 5× more than prior studies, ranging from classical statistical and machine learning (ML) methods to modern deep learning (DL) architectures and recently introduced foundation models (FM) with rigorous parameter tuning on TimeFCST-50M, a carefully curated hyperparameter optimization (HPO) subset with balanced distribution. Beyond aggregate performance comparisons, we conduct extensive domain- and setting-specific analyses across model families and, to our knowledge for the first time, rigorously validate these comparative findings with statistical tests over large-scale experiments. We further introduce a failure-profiling framework that characterizes model-specific error patterns across six failure modes, uncovering how similar aggregate results can arise from fundamentally different modes of forecasting error patterns. Finally, as a practical contribution, we release open-source implementations of all evaluated methods and the evaluation framework for future studies (git, 2026). Table 1 compares the key characteristics of TimeFCST with those of existing benchmarks.

Our comprehensive experimental study across the 128 datasets in TimeFCST addresses three central questions: *what works, when it works, and how it fails*. Our evaluation reveals: (i) although modern DL and FM models lead the aggregate rankings, their numerical gains over well-tuned classical baselines are often not statistically significant, especially on UTS, with a clear advantage emerging only for leading FMs; meanwhile, lightweight MLP- and RNN-based models remain highly competitive in both UTS and MTS settings; (ii) model effectiveness varies substantially across domains and evaluation settings, with simpler, more efficient classical baselines matching or outperforming DL and FM models in domains such as retail, nature, and economics and under challenging temporal dynamics such as non-stationarity; (iii) failure profiles reveal these weaknesses beyond aggregate metrics: similar aggregate performance can mask fundamentally different failure modes. Level error is most prevalent overall; DL and FM models exhibit more Phase and Localized errors, with FMs showing stronger underprediction bias; and (iv) we further identify saturated datasets that mask methodological progress and demonstrate that broad, diverse benchmark coverage can uncover otherwise hidden, statistically significant differences among models.

We start with the preliminaries and related work in the field of time-series forecasting (Section 2). Then we highlight the contributions of this work, detailed as follows:

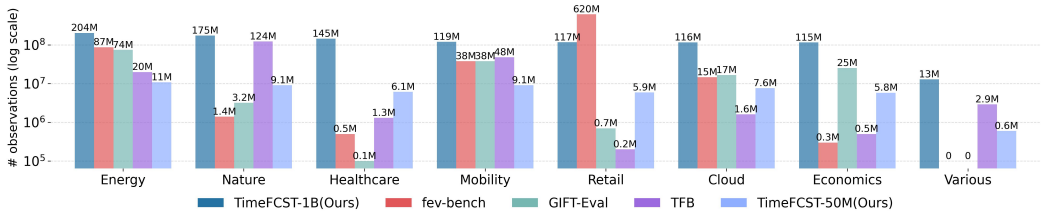

Figure 1: Domain-wise distribution of observations across time series forecasting benchmarks. Our proposed benchmarks demonstrate a more balanced and comprehensive coverage across domains.

- We construct `TimeFCST`, a billion-scale time-series forecasting benchmark with balanced coverage across domains, curated through a rigorous cleaning and quality assurance pipeline.
- We evaluate 100 representative algorithms, spanning traditional statistical and machine learning methods to state-of-the-art deep learning and foundation models, with systematic hyperparameter tuning to ensure a fair and faithful assessment of their performance.
- We move beyond aggregate metrics and leaderboard rankings by systematically characterizing domain-specific performance and introducing failure-profiling framework that reveal distinct error patterns, providing a roadmap for more targeted forecasting research.

Finally, we discuss the conclusions of the experimental benchmark on `TimeFCST` and potential directions for future time-series forecasting model advancement.

2 Preliminaries and Related Work

In this section, we start with an introduction of notation and concepts, as well as related studies on time-series forecasting models and the limitations of current benchmark studies.

2.1 Notation and Definitions

Definition 1 (Time Series). *Let $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ denote a repository of n time-series instances. Each instance $X_i \in \mathbb{R}^{T\times d}$ consists of T temporal observations across d channels.*

Based on the dimensionality of the observation space, a time-series is classified as *univariate* (UTS) if $d = 1$ and *multivariate* (MTS) if $d > 1$. While UTS analysis primarily focuses on capturing temporal dependencies, MTS introduces additional complexity by requiring models to account for inter-channel correlations and high-dimensional dynamics and dependencies.

Definition 2 (Subsequence). *Given a time-series $X \in \mathbb{R}^{T\times d}$, a subsequence $X_{s,e}$ is defined as a contiguous segment of observations starting from offset s and ending at offset e. Formally, $X_{s,e} = (\vec{x}_s, \vec{x}_{s+1}, \ldots, \vec{x}_e)$, where $1 \le s \le e \le T$. Each $\vec{x}_t \in \mathbb{R}^d$ represents the d-dimensional observation at time t. The length of the subsequence is given by $|X_{s,e}| = e - s + 1$.*

Definition 3 (Sliding Window). *Based on the definition of subsequences, a sliding window of width w is a specific subsequence $X_{i,i+w-1}$. By traversing the series X with a fixed stride s, we extract a collection of such windows: $\mathcal{W} = \{X_{1+(k-1)s,w+(k-1)s}\}_{k=1}^{K}$. Here, $K = \lfloor \frac{T-w}{s} \rfloor + 1$ represents the total number of extracted windows, where each window is a local segment of shape (w, d).*

Definition 4 (Time Series Forecasting). *Given a time-series $X \in \mathbb{R}^{T\times d}$ and a look-back window of length L, the task of forecasting is to learn a mapping $f : \mathbb{R}^{L\times d} \to \mathbb{R}^{H\times d}$ that generates a forecast $\hat{X}_{t+1,t+H} = f(X_{t-L+1,t})$ of a horizon H. The model is optimized to minimize the discrepancy between the prediction and the ground truth $X_{t+1,t+H}$, formally defined by a objective function $\mathcal{L}$:$\mathcal{L} = \psi(X_{t+1,t+H}, \hat{X}_{t+1,t+H})$, where $\psi(\cdot)$ denotes a distance metric or loss function (e.g., ℓ_1 or ℓ_2 norm) that quantifies the prediction error over the forecast horizon H. In our study, we employ a diverse set of metrics to evaluate this discrepancy (Section 4.3).*

Definition 5 (Time-Series Features and Patterns). *Given a time-series $X \in \mathbb{R}^{T\times d}$, a time-series feature is a measurable function $\phi(X)$ that summarizes a statistical or temporal property of X. A time-series pattern refers to a characteristic behavior described by one or more such features, such as trend, seasonality, stationarity, intermittency, and cross-channel correlation.*
Detailed definitions and formulations of additional characteristics, such as trend, seasonality, and correlation, are provided in Appendix A, with visualization in Figure 11 and 12.

2.2 Related Work

In this section, we briefly review representative forecasting models as well as existing benchmark and evaluation studies that motivate the design of our TimeFCST benchmark.

Time-Series Forecasting Models. Time-series forecasting has evolved substantially over the past few decades (Kim et al., 2025). Traditional methods include statistical models, which rely on structured probabilistic assumptions (Chatfield, 1978; Assimakopoulos & Nikolopoulos, 2000; De Livera et al., 2010; Permanasari et al., 2013), and machine learning methods, which formulate forecasting as a supervised regression problem using lagged observations and engineered features (Beja-Battais, 2023; Breiman, 2001; Chen & Guestrin, 2016). Over the past decade, deep learning architectures and foundation models have driven major advances in forecasting and achieved strong overall rankings in aggregate evaluations (Liu et al., 2024c; Nie et al., 2023; Jin et al., 2024; Das et al., 2024b).

Time-Series Forecasting Benchmarks and Evaluation Studies. Existing forecasting benchmarks have advanced evaluation along several complementary dimensions. The Monash Archive (Godahewa et al., 2021) and GIFT-Eval (Aksu et al., 2024) expand dataset diversity and evaluation scale, while TSLib (Wang et al., 2025), TFB (Qiu et al., 2025a), and TSFM-Bench (Li et al., 2025) standardize comparisons across different model families and learning settings. Fev-Bench (Shchur et al., 2026) further emphasizes realistic task formulations and principled evaluation protocols. Despite this progress, several common limitations remain, potentially leading to biased evaluations and misleading conclusions about model effectiveness. We summarize these limitations below.

3 Common Limitations in Existing Forecasting Benchmarks

In this section, we summarize common limitations from two perspectives: dataset construction and model evaluation. These limitations motivate the design and construction of TimeFCST, which aims to provide a large-scale and domain-specific evaluation platform.

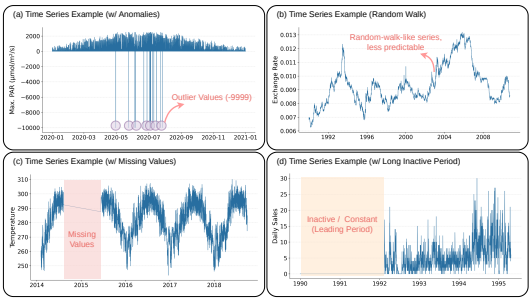

Figure 2: Examples of low-quality time series in public datasets. (a) Anomalous data; (b) Random walk, noisy data; (c) Missing values; (d) Constant segments, a long inactive period.

Limitations in Benchmark Datasets. Public time-series datasets often suffer from data quality issues such as missing values, abnormal spikes, and long inactive regions, which can distort evaluation and lead to misleading predictability. Likewise, purely stochastic series provide limited meaningful signal for evaluating forecasting capability. Beyond data quality, existing benchmarks also exhibit substantial domain imbalance, with observations concentrated in only a few application areas while others remain underrepresented (Figure 1). Such imbalance can bias evaluation results, since models effective in domains like Energy or Mobility may not generalize to Retail forecasting. Accordingly, a robust forecasting benchmark should integrate clearly defined quality assurance protocols and ensure extensive, balanced coverage of relevant domains towards rigorous and representative evaluation.

Limitations in Model Collection and Evaluation Protocols. Although modern architectures have achieved strong performance, traditional statistical and machine learning methods remain highly competitive in many forecasting settings with proper parameter tuning despite substantially lower complexity, while many simple baselines are heavily overlooked in existing benchmark studies. Moreover, reliance on aggregate scores can obscure substantial performance heterogeneity across domains, data patterns, and evaluation settings (Saqur et al., 2026). Statistical significance testing is needed to distinguish robust improvements from marginal fluctuations, while fine-grained error and failure analyses are necessary to reveal where and how different models struggle. Without such analyses, aggregate rankings alone may provide an incomplete understanding of model effectiveness. Next, we discuss our benchmark which solves both of these limitations.

4 TimeFCST: Our Proposed Benchmark

We propose TimeFCST, a large-scale time-series forecasting benchmark designed for comprehensive evaluation. In the following, we go through (i) dataset preparation and quality control; (ii) model search space and optimization protocol; and (iii) evaluation pipeline, metrics, and failure profile. An overview of benchmark pipeline and taxonomy can be seen from Figure 3.

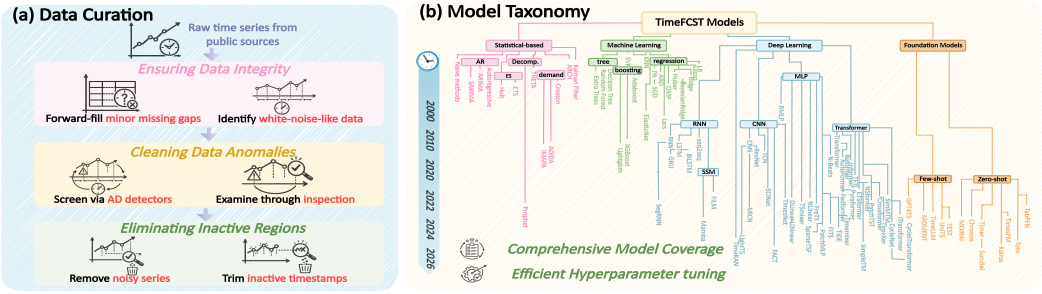

Figure 3: Overview of TimeFCST. (a) Data curation pipeline: raw public time series are processed through a multi-stage curation pipeline. (b) Model taxonomy: a comprehensive coverage of forecasting paradigms of 100 models across traditional techniques and modern methods.

4.1 Dataset Preparation and Quality Assurance

Data Acquisition and Standardization. To capture diverse real-world forecasting scenarios, we collect 128 public time-series datasets spanning multiple domains, including energy, nature, healthcare, mobility, retail, cloud computing, and economics[1] (as shown in Figure 1). The datasets cover heterogeneous temporal resolutions, from high-frequency measurements to yearly observations, and include both univariate and multivariate forecasting settings (Figure 11 and 12 in Appendix). Such diversity is important for evaluating model generalization across varying domains and data patterns.

Data Curation and Quality Control. We apply a multi-stage curation pipeline in TimeFCST combining automated screening with further inspection for data quality assurance (Figure 2). As shown in Figure 3, the pipeline first addresses missing observations and temporal inconsistencies, followed by anomaly detection and further verification to distinguish recording errors from legitimate domain events. We further filter or trim series dominated by random-walk-like behavior or extended inactive regions (particularly in test regions). Finally, we recompute dataset statistics and conduct a final round of inspection to verify data quality and ensure balanced coverage across domains. Further details and corresponding methodologies are provided in Appendix E for reference.

4.2 Model Search Space and Optimization Protocol

Comprehensive Model Coverage. TimeFCST evaluates 100 representative forecasting models across four major categories: statistical (Stats), machine learning (ML), deep learning (DL), and foundation models (FM). This broad coverage spans classical baselines, modern neural architectures, and pretrained foundation models[2], which attempts to avoid underrepresenting simple yet effective classical approaches that are frequently omitted from recent benchmarks, enabling a more reliable assessment of model progress. Detailed model descriptions are provided in Appendix C.

Efficient Hyperparameter Optimization. Exhaustively tuning 100 models across all 128 datasets requires over 100,000 experimental runs, many of which incur substantial costs on full-scale datasets. We therefore introduce TimeFCST-50M that maintains a similarlly balanced data distribution in comparison to TimeFCST-1B and provides dedicated validation sets for efficient tuning (Figure 1). This design enables systematic optimization beyond a single default configuration while substantially reducing computational costs, supporting a reliable assessment of model performance.

4.3 Evaluation Framework and Model Failure Profile

We start from a standardized evaluation pipeline inspired by the protocols in TFB (Qiu et al., 2025a) and TSLib (Wang et al., 2025). We further introduce a systematic *failure-profiling* framework to characterize model errors beyond aggregate metrics. The frameworks are described below.

Data Partitioning and Forecasting Schemes: We evaluate models using both fixed- and rolling-window forecasting schemes. The fixed-window scheme evaluates a single test forecast window following the training and validation sets, whereas rolling forecasting evaluates multiple forecast origins throughout the test set to provide a more robust assessment of predictive consistency over time (Qiu et al., 2025a). We adopt a default 6:2:2 split and overlapping windows for shorter series

[1]We also include a "Various" category for datasets that inherently span multiple domains.

[2]Foundation models are evaluated in the zero-shot setting when supported and the few-shot setting otherwise, while task-specific models are evaluated in the full-shot setting.

and non-overlapping windows for longer series to improve computational efficiency. We further define frequency-aware short-, medium-, and long-term horizons that reflect practical time scales to avoid arbitrary uniform lengths; for example, horizons of 48, 168, and 336 for hourly data correspond to 2 days, 1 week, and 2 weeks, respectively. More details are provided in Appendix Tables 2 and 3.

Data Preprocessing and Evaluation Metrics: Following established forecasting practices, we standardize each input series using z-score normalization, with statistics estimated from the training split to prevent information leakage. We use Mean Absolute Scaled Error (MASE) as the primary scale-free metric, enabling comparisons across datasets with different scales and units. Additionally, we also report seven complementary metrics to capture different aspects of forecasting performance. Detailed descriptions and formulations of all eight metrics are provided in Appendix Section D.

Failure Profile Construction and Analysis: Prior studies provided initial explorations of forecasting errors through classical error decomposition, structure-aware training objectives targeting particular distortions (Ahlburg, 1984; Le Guen & Thome, 2019; Zhao et al., 2026). However, these studies typically focus on only a few error categories, restricted settings, or training objective design. In `TimeFCST`, we introduce a model-agnostic framework for systematically characterizing "how forecasting models fail." Specifically, we define a *failure mode* as a structured discrepancy between a prediction and label whose correction produces a measurable reduction in forecasting error. Further, a *failure profile* can be constructed to capture the distinct error patterns for each model. Ideally, an effective failure profile should (i) cover common and complementary error patterns with limited redundancy; (ii) assign scores that directly reflect their relevance to the observed error; and (iii) support lightweight, model-agnostic post-hoc analysis based on predictions and ground truth.

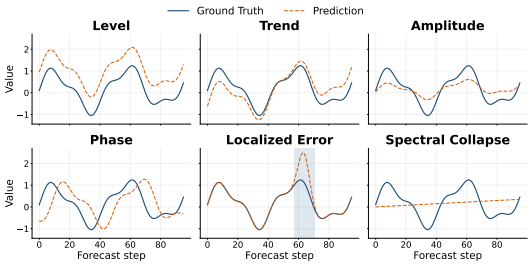

Figure 4: Examples of six major forecasting error patterns suggested in `TimeFCST`.

Based on these insights, our failure profile consists of six modes: *level*, *trend*, *amplitude*, *phase*, *localized error*, and *spectral collapse*, each capturing a distinct forecasting error pattern (see Appendix E.3 for more details). Let $\mathbf{Y}_i$ and $\hat{\mathbf{Y}}_{m,i}$ denote the ground truth and the prediction of model m for forecasting instance i, respectively, with original error $e_{m,i} = \ell(\mathbf{Y}_i, \hat{\mathbf{Y}}_{m,i})$. For each failure mode j, we define an oracle repair operator $\mathcal{C}_j$ that corrects the corresponding discrepancy while preserving the remaining prediction as much as possible:

$$\hat{\mathbf{Y}}^{(j)}_{m,i} = \mathcal{C}_j(\hat{\mathbf{Y}}_{m,i}, \mathbf{Y}_i), \qquad e^{(j)}_{m,i} = \ell(\mathbf{Y}_i, \hat{\mathbf{Y}}^{(j)}_{m,i}).$$

For example, letting $\mathcal{T}_c(\cdot)$ extract a centered trend component, the trend-repaired prediction is

$$\hat{\mathbf{Y}}^{(\text{trend})}_{m,i} = \hat{\mathbf{Y}}_{m,i} + \mathcal{T}_c(\mathbf{Y}_i) - \mathcal{T}_c(\hat{\mathbf{Y}}_{m,i}).$$

For model m and analysis group g containing instances $\mathcal{I}_g$, we define the aggregated *Recovery Score* of failure mode j as follows, which quantifies the proportion of forecasting error recoverable by repairing failure mode j based on the oracle:

$$\mathrm{RS}_{m,g,j} = \frac{1}{|\mathcal{I}_g|} \sum_{i \in \mathcal{I}_g} \left[\frac{e_{m,i} - e^{(j)}_{m,i}}{e_{m,i} + \epsilon} \right]_+ .$$

The resulting failure profile is $\mathbf{P}_{m,g} = [\mathrm{RS}_{m,g,1}, \ldots, \mathrm{RS}_{m,g,J}]$, where larger values indicate greater recoverable error associated with the corresponding failure mode. In practice, the profile can be further normalized across J failure modes, with a larger component indicating the greater relative prominence of the "error pattern" measured by the corresponding failure mode across the evaluated windows. Detailed failure-mode operators are provided in Appendix E, more result analysis is presented in Section 5.3. By revealing distinct weaknesses hidden behind similar aggregate performance, the proposed framework supports more targeted model diagnosis, selection, and improvement.

5 Experimental Results

In this section, we comprehensively evaluate 100 model performance across 128 `TimeFCST` datasets. Our analyses start from three major questions: (i) *what works* overall across global settings; (ii) *when*

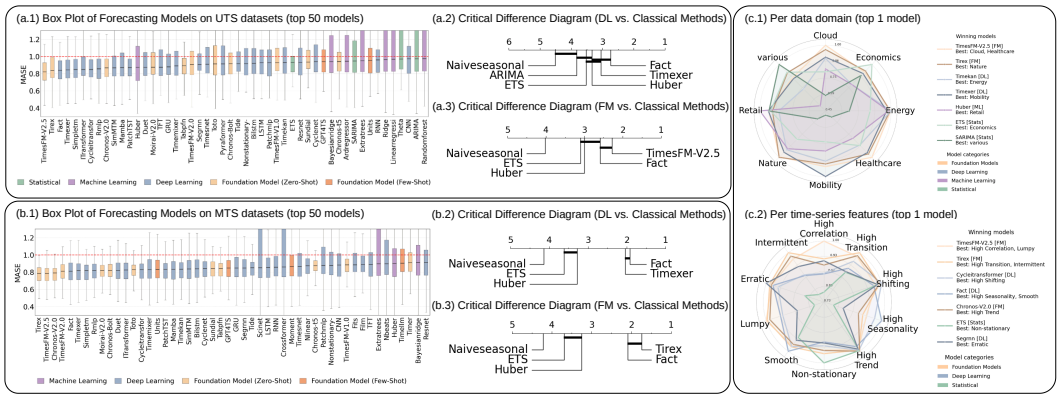

Figure 5: Global performance on TimeFCST across model categories and domains. Critical Difference (CD) diagrams illustrating the average model rankings with the Friedman-Nemenyi test. Models connected by a solid horizontal bar show no significant performance difference.

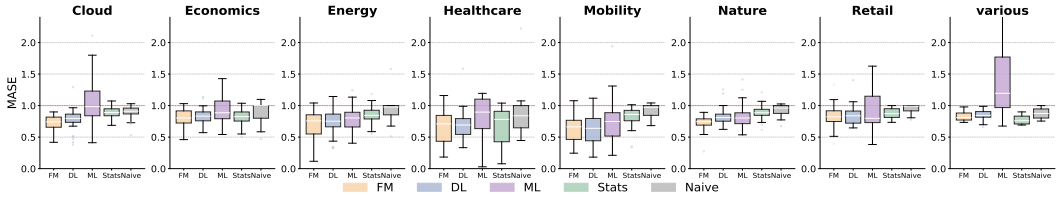

Figure 6: Performance across diverse domains. Simple classical methods remain competitive and even outperform FM and DL models in domains such as Economics, Nature, and Retail.

it works across domains and settings; and (iii) *how it fails* beyond aggregate metrics. We further examine how dataset selection shapes benchmark conclusions and model scalability. More details about experiment settings, model parameters, and statistical tests can be found in Appendix C and F.

5.1 *What works*: The Overall Performance Landscape

UTS: On aggregate metrics, foundation models (FMs), such as *TimesFM* and *TiReX*, generally occupy the leading positions in UTS, followed by modern deep learning (DL) methods such as *Fact* and *TimeXer* (Figure 5(a)). Notably, several simple statistical and machine learning (ML) methods that are often overlooked in recent benchmarks remain highly competitive: *Huber* and *ETS* rank among a top position among 100 evaluated methods in the UTS setting, outperforming approximately 70% of the model pool. Statistical tests further show that even strong deep learning methods, such as FACT, do not significantly outperform these classical baselines (Figure 5(a.2)); a statistically significant advantage emerges only among leading foundation models such as *TimesFM* (Figure 5(a.3)).

MTS: The performance landscape shifts markedly under MTS. As shown in Figure 5(b), FMs and DL models achieve statistically significant advantages over classical baselines. Many statistical and ML methods are not explicitly designed to capture cross-channel dependencies and high-dimensional dynamics, contributing to their widening performance gap on MTS. Within the DL category, it is also observed that lightweight neural models, including *RMLP*, *Mamba*, and *RNN*-based methods, can achieve consistently high rankings without elaborate designs (Figure 5 and 13 in Appendix). See top models' performance from each model family in Figure 14 and Figure 15 in Appendix.

> ***Takeaway:*** *Modern DL and FM models lead overall, yet only top FM methods significantly outperform well-tuned classical baselines on UTS, while simple neural architectures such as MLP and RNN-based models remain competitive across both UTS and MTS settings.*

5.2 *When it works*: A Domain-specific Evaluation

Domain- and Characteristic-Specific Efficacy. Moving beyond global averages, we examine model performance across application domains and data characteristics (Figure 5(c) and 6) . Although DL and FM models generally perform strongly, their dominance is highly domain-dependent. They lead most clearly in cloud, healthcare and mobility, whereas classical methods remain competitive in economics, energy, nature, retail, and various (Figure 6). The domain-level radar plot (Figure 5(c.1)) further highlights: *Huber* achieves the best in retail, while *ETS* and *SARIMA* rank first in Economics and various domains respectively. Similar heterogeneity emerges across temporal characteristics

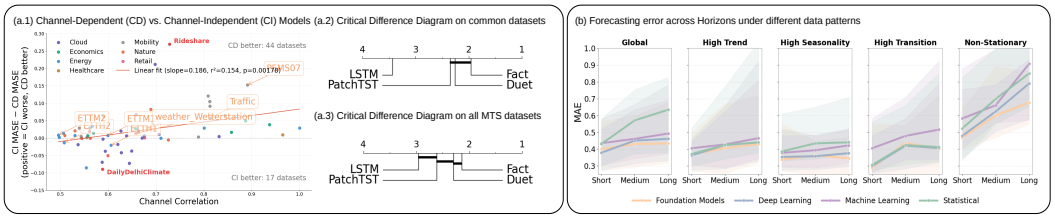

Figure 7: Performance comparison across MTS channel dependency and horizon settings.

(see Figure 16 in Appendix). Notably, despite their strong overall rankings, FMs and DL models fail to consistently outperform a simple statistical baseline such as *ETS* on non-stationary datasets (Figure 5(c.2)). These results demonstrate that model efficacy is strongly conditioned on data patterns.

Channel Dependency and Horizon-Specific Robustness. Our results reveal that Channel-Dependent (CD) models yield larger gains on strongly correlated data than Channel-Independent (CI) modeling (Figure 7(a)), as exemplified by Mobility datasets, which aligns with prior studies (Qiu et al., 2025a; Zhao et al., 2025). Moreover, Figure 7(a.2-3) shows that, across the full collection of 64 MTS datasets, leading CD models such as *Fact* significantly outperform strong CI models such as *PatchTST*, while several widely used MTS datasets provide limited discriminatory power. We further examine the implications of benchmark dataset selection in Section 5.4. Across forecasting horizons (Figure 7(b)), FMs and DL models generally maintain more stable performance from short- to long-term forecasting, with particularly clear advantages on highly seasonal data, yet they show no clear advantage over statistical methods on data with strong trends or frequent transitions.

> ***Takeaway:*** *Although FMs and DL models lead overall, classical methods match or outperform them in Retail, Economics, and non-stationary settings; the benefits of channel-dependent modeling and robustness across horizons also vary with data characteristics.*

5.3 *How it fails*: A Systematic Model Failure Profile

Failure Profiles Reveal Distinct Patterns of Model Errors. Figure 9 shows that Level is the dominant failure mode overall, followed by Phase, Localized, and Amplitude errors. Our complementary bias analysis further reveals an increasing tendency toward underprediction bias from statistical and ML methods to DL and FM models, with the strongest bias observed for FMs (Figure 17 and Figure 18 in Appendix). Additionally, similar aggregate accuracy can conceal fundamentally different weaknesses: models with comparable errors may fail in distinct ways (Figure 20), while models within the same family can exhibit different failure distributions (Figure 19). DL and FM models better capture overall shape and periodic structure, but remain susceptible to temporal misalignment (Phase) and missed sparse events (Localized) (Figure 21), whereas statistical methods are more prone to Level and Spectral Collapse. Beyond aggregate metrics, these profiles guide targeted improvements by revealing model-specific weaknesses under challenging temporal patterns and forecasting settings.

> ***Takeaway:*** *Failure profiles reveal fundamentally different error patterns behind similar aggregate performance: Level errors dominate overall; top DL and FM models are more susceptible to Phase and Localized errors, while FMs exhibit stronger underprediction bias.*

5.4 Broad and Diverse Datasets Are Essential for Reliable Benchmarking

In this section, we examine how dataset composition shapes benchmark conclusions and demonstrate the need for broad, large-scale coverage to reliably capture methodological progress.

Dataset Selection Shapes Benchmark Conclusions. Figure 7(a.2) shows that evaluations restricted to widely used MTS datasets can yield incomplete conclusions: top-performing channel-dependent (CD) DL methods, such as *Fact* and *Duet*, show no statistically significant advantage over channel-independent (CI) designs. In contrast, evaluation across the full curated collection of 64 MTS datasets reveals a clear advantage for CD models (Figure 7(a.3)), demonstrating how narrow dataset pools can obscure systematic methodological differences. Figure 8 and 22 further show that widely used datasets

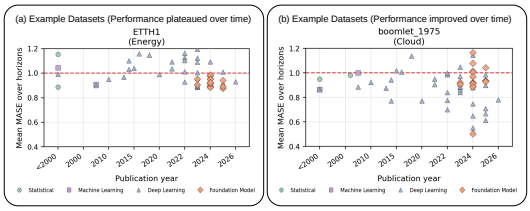

Figure 8: Performance trends over time. Some datasets have plateaued, while others continue to exhibit gains with methodological advances.

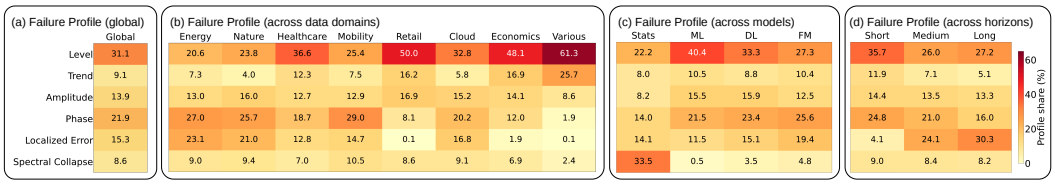

Figure 9: Failure profile analysis across (a) global; (b) data domains; (c) model families; (d) horizon settings. Models exhibit distinct failure patterns across domain, categories and evaluation settings.

such as ETT have become saturated, exhibiting limited gains from newer models, while other datasets continue to reflect methodological progress. Overreliance on saturated datasets limits discriminability and biases assessments of progress, underscoring the need for broad and diverse benchmarks.

> ***Takeaway:*** *Benchmark conclusions are highly sensitive to dataset selection: narrow dataset pools can obscure methodological differences, while saturated datasets limit the ability to measure progress. Broad and diverse coverage is therefore essential for reliable evaluation.*

5.5 Forecasting Scalability and Scaling

In this section, we examine the scalability of representative forecasting models and the scaling trends in pretrained foundation models in recent years, visualized in Figure 10.

Accuracy vs. Efficiency. Classical methods such as *ETS* provide lightweight yet robust baselines, achieving competitive accuracy across diverse domains. Foundation models shift much of the cost to large-scale offline pre-training and support zero-shot forecasting at inference time with great generalization capability. With parallel GPU execution, models such as *TimesFM*, *Chronos*, *Tirex*, and *Moirai* achieve strong accuracy with practical inference throughput. Nevertheless, they remain vulnerable to certain domains and data patterns as previously discussed. Deep learning models occupy an intermediate regime, combining strong learning capacity with GPU computation costs.

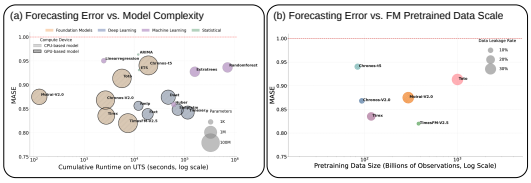

Figure 10: Forecasting error across top performing models (represented by bubble circles). (a) MASE vs. runtime, where the circle size represents the parameter counts; (b) MASE vs. pretrained dataset scale, where the circle size represents the leakage rate in pretraining set compared with `TimeFCST`.

Pre-training Scale and Data Leakage. Figure 10(b) relates zero-shot forecasting error to pre-training scale and estimated data leakage. Performance improves as the corpus grows to approximately 100 billion observations, but further scaling yields diminishing returns, suggesting that future progress should target specific model weaknesses rather than data volume alone. While data leakage remains an important concern, leading FMs such as *TimesFM* and *Chronos* exhibit estimated leakage rates of approximately 10% or lower. Their strong performance despite this limited overlap suggests meaningful generalization rather than gains driven primarily by training-set contamination.

> ***Takeaway:*** *Classical methods remain strong, low-cost baselines, while foundation models offer competitive zero-shot accuracy and efficient inference but require costly pre-training, with gains diminishing beyond 100B observations. Future progress should move beyond aggregate scores toward domain-specific challenges and recurring failure modes.*

6 Conclusion

In this work, we present `TimeFCST`, a billion-scale benchmark comprising 128 curated datasets across seven major domains and 100 models spanning statistical, machine learning, deep learning, and foundation-model paradigms. FMs lead overall, yet well-tuned classical and simple neural methods remain competitive. Performance and failure modes vary substantially across domains, horizons, temporal patterns, and channel dependencies, while benchmark conclusions are highly sensitive to dataset composition. We further observe diminishing FM scaling gains beyond 100 billion pre-training observations. Although currently limited to point forecasting without exogenous variables, `TimeFCST` provides a rigorous foundation for robust, taxonomy-aware forecasting evaluation, with probabilistic and covariate-aware forecasting left for future work.

AI USE STATEMENT

In this work, we used generative AI tools to improve the readability of the manuscript; assist with code for result analysis and visualization. We did not use generative AI to implement the evaluated methods, generate synthetic datasets, clean or reformat the benchmark data, perform translations, or develop theoretical claims or proofs; proof-related tasks are not applicable to this work. Additionally, we used generative AI tools for brainstorming, identifying relevant literature, and interpretation of experimental results. All AI-assisted text, code, and analyses were carefully reviewed by the authors. AI-assisted code was inspected, executed, and validated against the underlying experimental outputs, while suggested literature was manually verified using the original sources. The authors independently verified all reported results and interpretations and take full responsibility for the final content of this research work, including any text, figures, claims, code, or artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we document the complete dataset acquisition, preparation, and quality-control pipeline in Section 4.1 and Appendices B and E. The model collection and implementation details are described in Section 4.2 and Appendix C. We further specify the model training and evaluation pipeline, including evaluation metrics and experimental settings, in Section 4.3 and Appendices D, E, and F. An anonymous version of our repository is provided in (git, 2026) and includes the source code and instructions required to reproduce the training procedures and evaluation protocols; the repository will be publicly released upon publication.

REFERENCES

Anonymized github. https://anonymous.4open.science/r/TimeFCST, 2026. Accessed: 2026-09-23.

Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering–a decade review. *Information systems*, 53:16–38, 2015.

Dennis A. Ahlburg. Forecast evaluation and improvement using theil's decomposition. *Journal of Forecasting*, 3(3):345–351, 1984. doi: https://doi.org/10.1002/for.3980030313. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/for.3980030313.

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, 2024. URL https://arxiv.org/abs/2410.10393.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting, 2025. URL https://arxiv.org/abs/2510.15821.

V. Assimakopoulos and K. Nikolopoulos. The theta model: a decomposition approach to forecasting. *International Journal of Forecasting*, 16(4):521–530, October 2000. ISSN 0169-2070. doi: 10.1016/s0169-2070(00)00066-2. URL http://dx.doi.org/10.1016/S0169-2070(00)00066-2.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. Tirex: Zero-shot forecasting across long and short horizons with enhanced in-context learning, 2025. URL https://arxiv.org/abs/2505.23719.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018. URL https://arxiv.org/abs/1803.01271.

Debasish Basak, Srimanta Pal, and Dipak Patranabis. Support vector regression. *Neural Information Processing – Letters and Reviews*, 11, 11 2007.

Perceval Beja-Battais. Overview of adaboost : Reconciling its views to better understand its dynamics, 2023. URL https://arxiv.org/abs/2310.18323.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986. doi: 10.1016/0304-4076(86)90063-1.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, Hoboken, NJ, 5 edition, 2015. ISBN 9781118675021.

Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. doi: 10.1023/A:1010933404324.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, 1984.

C. Chatfield. The holt-winters forecasting procedure. *Applied Statistics*, 27(3):264, 1978. ISSN 0035-9254. doi: 10.2307/2347162. URL http://dx.doi.org/10.2307/2347162.

H. Chen, V. Luong, L. Mukherjee, and V. Singh. Simpletm: A simple baseline for multivariate time series forecasting. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=oANkBaVci5.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting, 2023. URL https://arxiv.org/abs/2303.06053.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794. ACM, August 2016. doi: 10.1145/2939672.2939785. URL http://dx.doi.org/10.1145/2939672.2939785.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014. URL https://arxiv.org/abs/1409.1259.

Robert B. Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition procedure based on loess (with discussion). *Journal of Official Statistics*, 6: 3–73, 1990.

Ben Cohen, Emaad Khwaja, Kan Wang, Charles Masson, Elise Ramé, Youssef Doubli, and Othmane Abou-Amal. Toto: Time series optimized transformer for observability, 2024. URL https://arxiv.org/abs/2407.07874.

Pádraig Cunningham and Sarah Jane Delany. k-nearest neighbour classifiers - a tutorial. *ACM Comput. Surv.*, 54(6), July 2021. ISSN 0360-0300. doi: 10.1145/3459665. URL https://doi.org/10.1145/3459665.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder, 2024a. URL https://arxiv.org/abs/2304.08424.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, 2024b. URL https://arxiv.org/abs/2310.10688.

Alysha De Livera, Rob Hyndman, and Ralph Snyder. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106: 1513–1527, 01 2010. doi: 10.1198/jasa.2011.tm09771.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7: 1–30, December 2006. ISSN 1532-4435.

Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. Simmtm: A simple pre-training framework for masked time-series modeling, 2023. URL https://arxiv.org/abs/2302.00861.

Jens E d'Hondt, Haojun Li, Fan Yang, Odysseas Papapetrou, and John Paparrizos. A structured study of multivariate time-series distance measures. In *Proceedings of the 2025 ACM SIGMOD international conference on management of data*, 2025.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2), April 2004. ISSN 0090-5364. doi: 10.1214/009053604000000067. URL http://dx.doi.org/10.1214/009053604000000067.

Graham Elliott, Thomas J. Rothenberg, and James H. Stock. Efficient tests for an autoregressive unit root. *Econometrica*, 64(4):813–836, 1996. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/2171846.

Robert F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987, July 1982. ISSN 0012-9682. doi: 10.2307/1912773. URL http://dx.doi.org/10.2307/1912773.

Kun Feng, Shaocheng Lan, Yuchen Fang, Wenchao He, Lintao Ma, Xingyu Lu, and Kan Ren. Kairos: Toward adaptive and parameter-efficient time series foundation models, 2026. URL https://arxiv.org/abs/2509.25826.

Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32:675–701, 1937. URL https://api.semanticscholar.org/CorpusID:120581754.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: A unified multi-task time series model, 2024. URL https://arxiv.org/abs/2403.00131.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006. doi: 10.1007/s10994-006-6226-1.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive, 2021. URL https://arxiv.org/abs/2105.06643.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models, 2024. URL https://arxiv.org/abs/2402.03885.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Julien Herzen, Francesco Lässig, Samuele Giuliano Piazzetta, Thomas Neuer, Léo Tafti, Guillaume Raille, Tomas Van Pottelbergh, Marek Pasieka, Andrzej Skrodzki, Nicolas Huguenin, Maxime Dumonal, Jan Kościsz, Dennis Bader, Frédérick Gusset, Mounir Benheddi, Camila Williamson, Michal Kosinski, Matej Petrik, and Gaël Grosch. Darts: User-friendly modern machine learning for time series, 2022. URL https://arxiv.org/abs/2110.03224.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, 2000. ISSN 00401706. URL http://www.jstor.org/stable/1271436.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second, 2023. URL https://arxiv.org/abs/2207.01848.

Mingyi Huang, Qinghua Liu, Paul Boniol, and John Paparrizos. Hydra: A multi-level hierarchy-driven approach for robust anomaly detection in time series. *Proc. ACM Manag. Data*, 4(3), May 2026. doi: 10.1145/3802074. URL https://doi.org/10.1145/3802074.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. Timekan: Kan-based frequency decomposition learning architecture for long-term time series forecasting, 2025. URL https://arxiv.org/abs/2502.06910.

Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 2nd edition, 2018. URL https://otexts.com/fpp2/.

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

Rob J. Hyndman, Anne B. Koehler, J. Keith Ord, and Ralph D. Snyder. *Forecasting with Exponential Smoothing: The State Space Approach*. Springer, Berlin, Germany, 2008. ISBN 978-3-540-71916-8. doi: 10.1007/978-3-540-71918-2.

Hoyoung Jeung, Sofiane Sarni, Ioannis Paparrizos, Saket Sathe, Karl Aberer, Nicholas Dawes, Thanasis G Papaioannou, and Michael Lehning. Effective metadata management in federated sensor networks. In *2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp. 107–114. IEEE, 2010.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models, 2024. URL https://arxiv.org/abs/2310.01728.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 3149–3157, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Jongseon Kim, Hyungjoon Kim, HyunGi Kim, Dongjun Lee, and Sungroh Yoon. A comprehensive survey of deep learning for time series forecasting: Architectural diversity and open challenges, 2025. URL https://arxiv.org/abs/2411.05793.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer, 2020. URL https://arxiv.org/abs/2001.04451.

Sanjay Krishnan, Aaron J Elmore, Michael Franklin, John Paparrizos, Zechao Shang, Adam Dziedzic, and Rui Liu. Artificial intelligence in resource-constrained and shared environments. *ACM SIGOPS Operating Systems Review*, 53(1):1–6, 2019.

Vincent Le Guen and Nicolas Thome. Shape and time distortion loss for training deep time series forecasting models. *Advances in neural information processing systems*, 32, 2019.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023.

Zhe Li, Xiangfei Qiu, Peng Chen, Yihang Wang, Hanyin Cheng, Yang Shu, Jilin Hu, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, and Bin Yang. Tsfm-bench: A comprehensive and unified benchmark of foundation models for time series forecasting, 2025. URL https://arxiv.org/abs/2410.11802.

T Warren Liao. Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874, 2005.

Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting, 2020. URL https://arxiv.org/abs/1912.09363.

Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pp. 2–11, 2003.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. Segrnn: Segment recurrent neural network for long-term time series forecasting, 2023. URL https://arxiv.org/abs/2308.11200.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns, 2024a. URL https://arxiv.org/abs/2409.18479.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. Sparsetsf: Modeling long-term time series forecasting with 1k parameters, 2024b. URL https://arxiv.org/abs/2405.00946.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting, 2026. URL https://arxiv.org/abs/2511.11698.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Prabhakar Kamarthi, Aditya Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. Time-mmd: Multi-domain multimodal dataset for time series analysis. *Advances in Neural Information Processing Systems*, 37:77888–77933, 2024a.

Haoxin Liu, Zhiyuan Zhao, Jindong Wang, Harshavardhan Kamarthi, and B. Aditya Prakash. LSTPrompt: Large language models as zero-shot time series forecasters by long-short-term prompting. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 7832–7840, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.466. URL https://aclanthology.org/2024.findings-acl.466/.

Haoxin Liu, Zhiyuan Zhao, Shiduo Li, and B Aditya Prakash. Evaluating system 1 vs. 2 reasoning approaches for zero-shot time-series forecasting: A benchmark and insights. *arXiv preprint arXiv:2503.01895*, 2025a.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction, 2022a. URL https://arxiv.org/abs/2106.09305.

Qinghua Liu and John Paparrizos. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 37:108231–108261, 2024.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022b. URL https://api.semanticscholar.org/CorpusID:251649164.

Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum, 2020. URL https://arxiv.org/abs/2007.07989.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting, 2023. URL https://arxiv.org/abs/2205.14415.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting, 2024c. URL https://arxiv.org/abs/2310.06625.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models, 2024d. URL https://arxiv.org/abs/2402.02368.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models, 2025b. URL https://arxiv.org/abs/2502.00816.

Greta M Ljung and George EP Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978.

Andrew W Lo and A Craig MacKinlay. Stock market prices do not follow random walks: Evidence from a simple specification test. *The review of financial studies*, 1(1):41–66, 1988.

Carl H Lubba, Sarab S Sethi, Philip Knaute, Simon R Schultz, Ben D Fulcher, and Nick S Jones. catch22: Canonical time-series characteristics, 2019. URL https://arxiv.org/abs/1901.10200.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *International journal of forecasting*, 38(4):1346–1364, 2022.

Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, 38(4):1958–2031, 2024.

C. De Mol, E. De Vito, and L. Rosasco. Elastic-net regularization in learning theory, 2008. URL https://arxiv.org/abs/0807.3423.

P. B. Nemenyi. Distribution-free multiple comparisons. Princeton University, 1963.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Konstantinos Nikolopoulos and Fotios Petropoulos. Forecasting for big data: Does suboptimality matter? *Computers & Operations Research*, 98:322–329, 2018. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2017.05.007. URL https://www.sciencedirect.com/science/article/pii/S0305054817301193.

Konstantinos Nikolopoulos, Aris A. Syntetos, John E. Boylan, Fotios Petropoulos, and Vassilios Assimakopoulos. An aggregate-disaggregate intermittent demand approach (adida) to forecasting: An empirical proposition and analysis. *Journal of the Operational Research Society*, 62(3):544–554, 2011. doi: 10.1057/jors.2010.32.

Nixtla. Nixtla/statsforecast. https://github.com/Nixtla/statsforecast, 2026. Python package. Original work published 2021.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting, 2020. URL https://arxiv.org/abs/1905.10437.

Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks, 2015. URL https://arxiv.org/abs/1511.08458.

Art B. Owen. A robust hybrid of lasso and ridge regression. 2006. URL https://api.semanticscholar.org/CorpusID:1617819.

John Paparrizos and Luis Gravano. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1855–1870. ACM, 2015.

John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S Tsay, Themis Palpanas, and Michael J Franklin. Tsb-uad: an end-to-end benchmark suite for univariate time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(8):1697–1711, 2022.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, November 2011. ISSN 1532-4435.

Yan Pei, Swarnendu Biswas, Donald S. Fussell, and Keshav Pingali. An elementary introduction to kalman filtering, 2019. URL https://arxiv.org/abs/1710.04055.

Adhistya Erna Permanasari, Indriana Hidayah, and Isna Alfi Bustoni. Sarima (seasonal arima) implementation on time series to forecast the number of malaria incidence. In *2013 International Conference on Information Technology and Electrical Engineering (ICITEE)*, pp. 203–207, 2013. doi: 10.1109/ICITEED.2013.6676239.

Aladdin Persson. aladdinpersson/machine-learning-collection. https://github.com/aladdinpersson/Machine-Learning-Collection, 2026. Python package. Original work published 2020.

Fotios Petropoulos and Nikolaos Kourentzes. Forecast combinations for intermittent demand. *Journal of the Operational Research Society*, 66(6):914–924, 2015. doi: 10.1057/jors.2014.62.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods, 2025a. URL https://arxiv.org/abs/2403.20150.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting, 2025b. URL https://arxiv.org/abs/2412.10859.

Dung Quoc Nguyen, Minh Nguyet Phan, and Ivan Zelinka. Periodic time series forecasting with bidirectional long short-term memory: Periodic time series forecasting with bidirectional lstm. In *Proceedings of the 2021 5th International Conference on Machine Learning and Soft Computing*, ICMLSC '21, pp. 60–64, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450387613. doi: 10.1145/3453800.3453812. URL https://doi.org/10.1145/3453800.3453812.

Raeid Saqur, Christoph Bergmeir, Blanka Horvath, Daniel Schmidt, Frank Rudzicz, and Terry Lyons. Seeking sota: Time-series forecasting must adopt taxonomy-specific evaluation to dispel illusory gains. *arXiv preprint arXiv:2603.15506*, 2026.

Patrick Schäfer. The boss is concerned with time series classification in the presence of noise. *Data Min. Knowl. Discov.*, 29(6):1505–1530, nov 2015. ISSN 1384-5810. doi: 10.1007/s10618-014-0377-7. URL https://doi.org/10.1007/s10618-014-0377-7.

Patrick Schäfer and Mikael Högqvist. Sfa: a symbolic fourier approximation and index for similarity search in high dimensional datasets. In *Proceedings of the 15th International Conference on Extending Database Technology*, EDBT '12, pp. 516–527, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450307901. doi: 10.1145/2247596.2247656. URL https://doi.org/10.1145/2247596.2247656.

Robin M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview, 2019. URL https://arxiv.org/abs/1912.05911.

Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P Boedihardjo, Crystal Chen, Susan Frankenstein, and Manfred Lerner. Grammarviz 2.0: a tool for grammar-based pattern discovery in time series. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part III 14*, pp. 468–472. Springer, 2014.

Oleksandr Shchur, Abdul Fatir Ansari, Caner Turkmen, Lorenzo Stella, Nick Erickson, Pablo Guerron, Michael Bohlke-Schneider, and Yuyang Wang. fev-bench: A realistic benchmark for time series forecasting, 2026. URL https://arxiv.org/abs/2509.26468.

Evangelos Spiliotis and Fotios Petropoulos. On the update frequency of univariate forecasting models. *European Journal of Operational Research*, 314(1):111–121, 2024. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2023.08.056. URL https://www.sciencedirect.com/science/article/pii/S0377221723006859.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series, 2024. URL https://arxiv.org/abs/2308.08241.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014. URL https://arxiv.org/abs/1409.3215.

Peiwang Tang and Weitai Zhang. Unlocking the power of patch: Patch-based mlp for long-term time series forecasting, 2024. URL https://arxiv.org/abs/2405.13575.

Sean J. Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1): 37–45, January 2018. ISSN 1537-2731. doi: 10.1080/00031305.2017.1380080. URL http://dx.doi.org/10.1080/00031305.2017.1380080.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. doi: 10.1111/j.2517-6161.1996.tb02080.x.

Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, September 2001. ISSN 1532-4435. doi: 10.1162/15324430152748236. URL https://doi.org/10.1162/15324430152748236.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

H. Wang, J. Shi, and L. Qing. Fact: Fine-grained across-variable convolution for multivariate time series forecasting. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=j3gNYqrHtl.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. 2023.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting, 2024a. URL https://arxiv.org/abs/2405.14616.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=INAeUQ04lT.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark, 2025. URL https://arxiv.org/abs/2407.13278.

David Wipf and Srikantan Nagarajan. A new view of automatic relevance determination. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'07, pp. 1625–1632, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting, 2022. URL https://arxiv.org/abs/2202.01381.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers, 2024. URL https://arxiv.org/abs/2402.02592.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, 2022. URL https://arxiv.org/abs/2106.13008.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis, 2023. URL https://arxiv.org/abs/2210.02186.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters, 2024. URL https://arxiv.org/abs/2307.03756.

Fan Yang and John Paparrizos. Spartan: Data-adaptive symbolic time-series approximation. *Proc. ACM Manag. Data*, 3(3), June 2025. doi: 10.1145/3725357. URL https://doi.org/10.1145/3725357.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Defu Lian, Ning An, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting, 2023. URL https://arxiv.org/abs/2311.06184.

Guoqi Yu, Jing Zou, Xiaowei Hu, Angelica I Aviles-Rivero, Jing Qin, and Shujun Wang. Revitalizing multivariate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=87CYNyCGOo.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting?, 2022. URL https://arxiv.org/abs/2205.13504.

Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures, 2022. URL https://arxiv.org/abs/2207.01186.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

Yuyang Zhao, Lian Xu, Hao Miao, Chenxi Liu, and Hao Xue. Ts-fault: Benchmarking time series forecasters against structural faults, 2026. URL https://arxiv.org/abs/2606.18539.

Zhiyuan Zhao, Alexander Rodriguez, and B Aditya Prakash. Performative time-series forecasting. *arXiv preprint arXiv:2310.06077*, 2023.

Zhiyuan Zhao, Juntong Ni, Shangqing Xu, Haoxin Liu, Wei Jin, and B Aditya Prakash. Timerecipe: A time-series forecasting recipe via benchmarking module level effectiveness. *arXiv preprint arXiv:2506.06482*, 2025.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting, 2021. URL https://arxiv.org/abs/2012.07436.

Tian Zhou, Ziqing Ma, Xue Wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, and Rong Jin. Film: Frequency improved legendre memory model for long-term time series forecasting, 2022a. URL https://arxiv.org/abs/2205.08897.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, 2022b. URL https://arxiv.org/abs/2201.12740.

Tian Zhou, PeiSong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained lm, 2023. URL https://arxiv.org/abs/2302.11939.

A TIME-SERIES DATA PATTERNS

In this section, we first go through the definition and formula of additional time series data features and patterns which are not mentioned in main content due to space limitations. We then present visual examples regarding time series patterns and demonstrate the diverse data composition of `TimeFCST`.

Definition 6 (Trend)**.** *The trend of a given time series represents the long-term patterns that occur over time. Formally, a Seasonal-Trend Decomposition Procedure Based on Loess (STL) (Cleveland et al., 1990) is used to decompose a time series X into three terms:*

$$X = X_S + X_T + X_R$$

where X_T, X_S, X_R refer to trend, seasonality, and the remainder, respectively. The strength of trend is then computed as follows.

$$Trend = max\left(0, 1 - \frac{var\,(X_R)}{var\,(X - X_S)}\right) \tag{1}$$

Definition 7 (Seasonality)**.** *Seasonality is defined as the recurring pattern in a time series that repeats at regular intervals. Similar to trend, the strength of seasonality in a time series X can be calculated:*

$$Seasonality = \max\left(0, 1 - \frac{var\,(X_R)}{var\,(X - X_T)}\right) \tag{2}$$

Definition 8 (Stationarity)**.** *Stationarity of a time series X is defined by three key properties: the mean of any observation is constant; for all observations, the variance is finite; the distance between any two observations exclusively determines the covariance. Augmented Dick-Fuller (ADF) test statistic (Elliott et al., 1996) is employed to measure stationarity, computed as follows.*

$$Stationarity = \begin{cases} True, & ADF(X) \leq 0.05 \\ False, & Otherwise \end{cases} \tag{3}$$

Definition 9 (Correlation)**.** *Correlation measures the degree to which different variables (i.e., channels) within a time-series instance share similar structural patterns. Let $X_i \in \mathbb{R}^{T\times d}$ denote a time-series instance with d variables. For each variable $j \in \{1, \ldots, d\}$, extract its Catch22 (Lubba et al., 2019) feature vector $F_i^j = \mathrm{Catch22}(X_i^{(:,j)})$, and collect all feature vectors from each channel*

$$F_i = [F_i^1, \ldots, F_i^d] \in \mathbb{R}^{22\times d}. \tag{4}$$

We compute all pairwise Pearson correlation coefficients between variables in the formula as:

$$P_i = \{r(F_i^j, F_i^k) \mid 1 \leq j < k \leq d\}. \tag{5}$$

The correlation of instance X_i is defined as

$$\mathrm{Corr}(X_i) = \mathrm{mean}(P_i) + \frac{1}{1 + \mathrm{var}(P_i)}. \tag{6}$$

Definition 10 (Transition)**.** *The transition of a time series X is defined as the trace of the covariance matrix of its symbol-to-symbol transition matrix, obtained after discretizing the series into a 3-letter alphabet (Lubba et al., 2019). Formally, given a time series X of length T, let the first zero crossing of the autocorrelation function $\tau = \mathrm{firstzero_ac}(X)$ and define the downsampled series $Y = \mathrm{Downsample}(X, \tau)$ of length T'. Let $I = \mathrm{argsort}(Y)$, then the 3-symbol representation is defined as $Z_j = \left\lfloor \frac{I_j}{T'/3} \right\rfloor, j = 1, \ldots, T'$. We construct the transition matrix $M \in \mathbb{R}^{3\times 3}$ by counting symbol transitions such that we can compute the transition value Δ as follows:*

$$\Delta = \mathrm{trace}\left(\mathrm{Cov}\left(\frac{1}{T'}\Big[\#\{j \mid Z_j = a,\ Z_{j+1} = b\}\Big]_{a,b=1}^{3}\right)\right) \tag{7}$$

Definition 11 (Shifting)**.** *Shifting is known as temporal changes in the probability distribution of the given time series. The shifting value $\delta \in (0, 1)$ is calculated as follows. Formally, given a Z-normalized time series X', let $\{s_i\}_{i=1}^m$ be a set of uniformly sampled thresholds over the interval $(\min(X'), \max(X'))$. Define*

$$M_i = \mathrm{median}(t \mid X'_t > s_i). \tag{8}$$

Let $M'_i = \mathrm{min_max}(M_i)$. The final shifting value is

$$\delta = \mathrm{median}_{1\leq i\leq m}\, M'_i \tag{9}$$

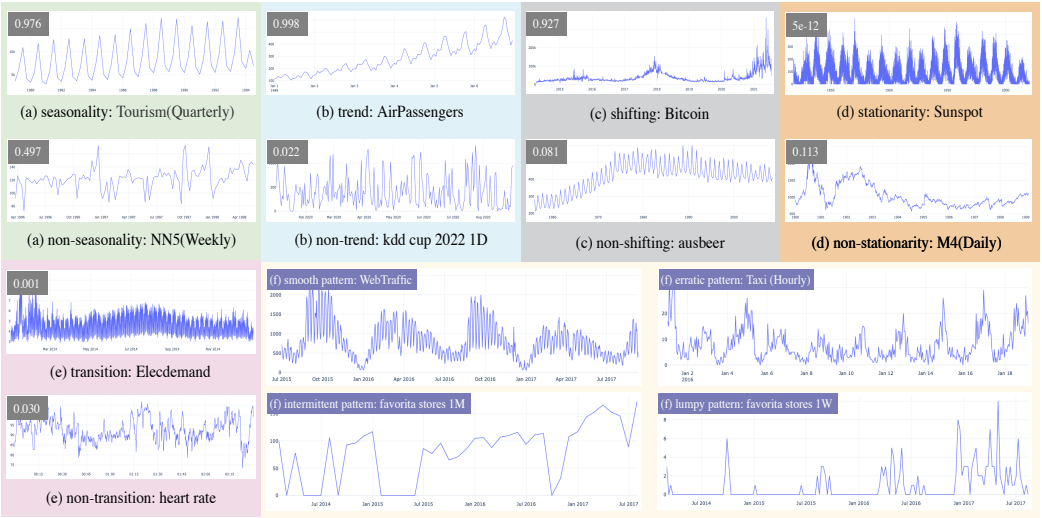

Figure 11: Qualitative examples of common time series patterns. We compare real-world datasets based on 5 major time series patterns: seasonality, trend, shifting, stationarity, transition, and 4 different distinct demand patterns, including smooth, erratic, intermittent, and lumpy behaviors.

Definition 12 (ADI)**.** *The average demand interval (ADI) measures how frequently non-zero demand occurs in a channel. Let $X_i \in \mathbb{R}^{T\times d}$ and consider channel j. Let*

$$N_{ij} = \sum_{t=1}^{T} \mathbf{1}\Big(X_i^{(t,j)} > 0\Big) \tag{10}$$

be the number of non-zero observations. The ADI of channel j is

$$\mathrm{ADI}(X_i^{(:,j)}) = \frac{T}{N_{ij}}. \tag{11}$$

Definition 13 (CV^2)**.** *The squared coefficient of variation, CV^2, quantifies the relative variability of positive demand values. For channel j, let μ_{ij} and σ_{ij} denote the mean and standard deviation of its strictly positive observations. Then CV^2 is calculated as follows:*

$$\mathrm{CV}^2(X_i^{(:,j)}) = \left(\frac{\sigma_{ij}}{\mu_{ij}}\right)^2. \tag{12}$$

Definition 14 (Demand Pattern)**.** *Demand patterns classify each time series channel based on its frequency and variability. Formally, given* ADI *and* CV^2*, channel j is categorized as*

$$\begin{cases} Smooth, & \mathrm{ADI} \le 1.32,\ \mathrm{CV}^2 \le 0.49, \\ & \textit{(frequent and stable demand)} \\ Erratic, & \mathrm{ADI} \le 1.32,\ \mathrm{CV}^2 > 0.49, \\ & \textit{(frequent but highly variable demand)} \\ Intermittent, & \mathrm{ADI} > 1.32,\ \mathrm{CV}^2 \le 0.49, \\ & \textit{(infrequent but stable demand sizes)} \\ Lumpy, & \mathrm{ADI} > 1.32,\ \mathrm{CV}^2 > 0.49, \\ & \textit{(infrequent and highly variable demand).} \end{cases} \tag{13}$$

The rich diversity of data patterns in `TimeFCST` may explain the performance variation of different categories of methods. Figures 11 and 12 further illustrate the remarkable heterogeneity of `TimeFCST`. Figure 11 presents representative examples spanning classical temporal properties (e.g., seasonality, trend, shifting, stationarity, and transition) as well as diverse demand behaviors (smooth, erratic, intermittent, and lumpy), highlighting the wide spectrum of forecasting challenges

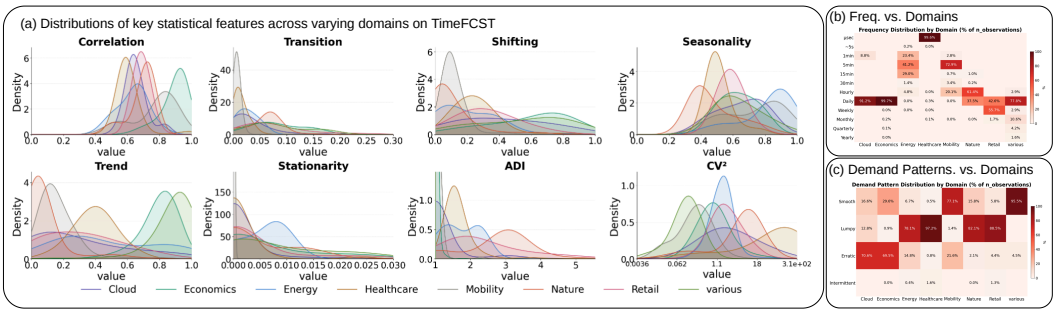

Figure 12: Data diversity and domain-specific characteristics of TimeFCST. (a) Distribution of key temporal features across data domains. (b) Heatmap of sampling frequencies, representing various granularities. (c) Classification of demand patterns categorized by domain.

encountered in practice. Figure 12 also complements these examples quantitatively by showing that such characteristics are unevenly distributed across domains. Broadly, human-centric and operational domains (e.g., Mobility) tend to exhibit smoother temporal dependencies with strong seasonality, whereas resource- and demand-driven domains (e.g., Energy, Retail, and Nature) are dominated by lumpy demand patterns and greater variability in temporal dynamics. In contrast, Healthcare datasets stand out as an extreme case, consisting primarily of ultra-high-frequency observations with sparse behaviors. Collectively, these results demonstrate that TimeFCST captures substantial diversity not only in temporal patterns, but also in sampling and domain-specific forecasting challenges. Furthermore, the performance degradation and shifts in model rankings across certain data domains reveal a failure to capture specific data patterns, offering valuable insights to guide future methodological improvements of task-specific or foundation models.

B Forecasting Dataset Task Details

Table 2 and Table 3 present the full statistical summary of all forecasting datasets included in our TimeFCST across both univariate and multivariate settings. For each dataset, we report its application domain, temporal frequency, number of series, dimensionality, total observations, sequence length range, forecasting horizon categories, and a collection of representative temporal characteristics, including seasonality, trend, shift, stationarity, transition behavior, correlation (only for multivariate datasets), and dominant demand pattern. These statistics highlight the substantial diversity of TimeFCST in terms of scale, domain coverage, and forecasting difficulty.

C Forecasting Model Details

Tables 4, 5, 6, and 7 summarize the complete forecasting model zoo included in TimeFCST with each code source, covering statistical methods, machine-learning-based approaches, deep learning architectures, and recent time-series foundation models. For each model, we report its architectural category, implementation source, and representative hyperparameter configuration. For multivariate forecasting models, we additionally indicate whether the method follows a channel-dependent (CD) design, which explicitly models cross-channel dependencies, or a channel-independent setting (CI) that processes each channel separately. Overall, the model suite is designed to provide a comprehensive coverage of forecasting methodologies.

Detailed model list with references:

- ***Statistical Baselines***: We include **18** classical methods ranging from fundamental benchmarks to sophisticated automated frameworks such as *Naive Moving Average*, *Naive Seasonal*, *Naive Mean*, *Naive Drift*, *Autoregressive*, *ARIMA* (Box et al., 2015), *SARIMA* (Permanasari et al., 2013), *Theta* (Assimakopoulos & Nikolopoulos, 2000), *ETS* (Hyndman et al., 2008), *Simple Exponential Smoothing* (Hyndman & Athanasopoulos, 2018), *Croston* (Hyndman & Athanasopoulos, 2018), *Prophet* (Taylor & Letham, 2018), *ADIDA* (Nikolopoulos et al., 2011), *IMAPA* (Petropoulos & Kourentzes, 2015), *ARCH* (Engle, 1982),

Table 2: **Dataset statistics (UTS)**. We report the dataset domain, frequency, the number of time series, channels and observations, the length range of time series, and forecasting horizons (short-term, medium-term, long-term); We also report the median of a variety of time series patterns such as seasonality, trend, shift, stationarity, transition, and the most common demand pattern.

Dataset	Domain	Freq	#Series	#Channels	#Observations	Length Range	ShortH	MedH	LongH	Seasonality	Trend	Shift	Stationarity	Transition	Demand Pattern
Extended_Wikipedia_Web_Traffic_Daily_Dataset	Cloud	Daily	29984	1	75138457	[289, 2557]	30	-	-	0.728	0.13	0.508	0.0	0.015	Erratic
WebTraffic	Cloud	Daily	29737	1	23038505	[271, 803]	30	-	-	0.515	0.259	0.308	0.0	0.015	Erratic
Wike2000	Cloud	Daily	9535	1	7551720	[792, 792]	14	-	-	0.611	0.494	0.216	0.004	0.028	Smooth
Bitcoin	Economics	Daily	18	1	74925	[2659, 4581]	14	30	60	0.674	0.853	0.867	0.564	0.074	Erratic
CIF_2016	Economics	Monthly	62	1	6733	[57, 120]	12	-	-	0.628	0.943	0.679	0.9	0.111	Smooth
NN5	Economics	Daily	111	1	4916856	[791, 87801]	14	30	60	0.806	0.749	0.112	0.0	0.002	Smooth
NN5(Weekly)	Economics	Weekly	111	1	12543	[113, 113]	13	-	-	0.516	0.383	0.336	0.0	0.021	Smooth
Tourism(Monthly)	Economics	Monthly	366	1	109280	[91, 333]	18	-	-	0.873	0.934	0.703	0.89	0.074	Erratic
Tourism(Quarterly)	Economics	Quarterly	426	1	42514	[45, 130]	8	-	-	0.876	0.974	0.73	0.915	0.074	Smooth
Tourism(Yearly)	Economics	Yearly	342	1	9545	[26, 47]	6	-	-	0.663	0.982	0.743	0.949	0.167	Smooth
ausbeer	Economics	Quarterly	1	1	211	[211, 211]	8	-	-	0.96	0.985	0.081	0.096	0.08	Smooth
AustralianElectricity	Energy	half hourly	5	1	1155264	[230736, 232272]	48	192	-	0.959	0.932	0.076	0.0	0.001	Smooth
Elecdemand	Energy	half hourly	1	1	17520	[17520, 17520]	48	192	336	0.928	0.907	0.046	0.001	0.001	Smooth
Solar(5Min.)	Energy	every 5 min	2000	1	83766083	[8726, 104952]	144	-	-	0.913	0.438	0.06	0.0	0.023	Lumpy
SolarPower	Energy	every 4 seconds	1	1	500000	[500000, 500000]	30	75	150	0.934	0.496	0.033	0.0	0.033	Lumpy
Wind_Farms	Energy	every minute	29	1	10036864	[2975, 500000]	48	144	288	0.695	0.415	0.228	0.0	0.051	Lumpy
kdd_cup_2022_10T	Energy	every 10 minutes	134	1	4727177	[35194, 35279]	14	-	-	0.474	0.007	0.089	0.0	0.001	Lumpy
kdd_cup_2022_1D	Energy	Daily	134	1	32562	[243, 243]	14	30	-	0.546	0.365	0.086	0.0	0.001	Erratic
kdd_cup_2022_30T	Energy	every 30 minutes	134	1	1575525	[11731, 11758]	48	192	-	0.475	0.01	0.088	0.0	0.001	Lumpy
us_gasoline	Energy	Weekly	1	1	1578	[1578, 1578]	13	-	-	0.554	0.874	0.316	0.001	0.074	Smooth
CoronaVirus_2019_confirmed	Healthcare	Daily	252	1	112866	[187, 494]	14	30	-	0.756	0.995	0.775	0.946	0.167	Erratic
CoronaVirus_2019_deaths	Healthcare	Daily	184	1	75414	[146, 483]	14	30	-	0.654	0.996	0.775	0.928	0.167	Erratic
CoronaVirus_2019_recovered	Healthcare	Daily	236	1	99133	[141, 494]	14	30	-	0.665	0.995	0.786	0.966	0.167	Erratic
Hospital	Healthcare	Monthly	767	1	64428	[84, 84]	18	-	-	0.678	0.558	0.393	0.061	0.04	Smooth
ecdc_ili	Healthcare	Weekly	25	1	4650	[89, 216]	13	-	-	0.814	0.387	0.269	0.003	0.015	Erratic
heart_rate	Healthcare	every 5 seconds	1	1	1800	[1800, 1800]	12	60	180	0.518	0.326	0.104	0.0	0.03	Smooth
physionet_2012	Healthcare	irregular	3238	1	296587	[80, 189]	6	-	-	0.656	0.549	0.383	0.006	0.037	Smooth
AirPassengers	Mobility	Monthly	1	1	144	[144, 144]	12	24	-	0.987	0.998	0.785	0.992	0.167	Smooth
Pedestrian_Counts	Mobility	Hourly	65	1	3131770	[1176, 96424]	48	168	-	0.948	0.516	0.153	0.0	0.008	Erratic
Taxi(Half_Hourly)	Mobility	half hourly	4742	1	4096924	[221, 1488]	48	-	-	0.806	0.194	0.14	0.0	0.005	Erratic
Taxi(Hourly)	Mobility	Hourly	5000	1	4114178	[149, 1488]	12	-	-	0.806	0.199	0.137	0.0	0.005	Erratic
UberTLC_daily	Mobility	Daily	1	1	181	[181, 181]	14	-	-	0.825	0.863	0.591	0.624	0.033	Smooth
UberTLC_hourly	Mobility	Hourly	1	1	4345	[4345, 4345]	48	168	-	0.956	0.573	0.27	0.0	0.017	Smooth
VehicleTrips	Mobility	Daily	306	1	38370	[60, 243]	12	-	-	0.749	0.593	0.232	0.113	0.018	Smooth
BrazilianCitiesTemperature	Nature	Monthly	12	1	9029	[481, 1320]	12	30	-	0.607	0.196	0.314	0.0	0.012	Erratic
Flood_Modeling_Dataset_1	Nature	Hourly	667	1	34931367	[926, 125139]	48	168	-	0.383	0.041	0.031	0.0	0.069	Lumpy
Flood_Modeling_Dataset_2	Nature	Hourly	661	1	34254046	[926, 123796]	48	168	-	0.378	0.031	0.039	0.0	0.072	Lumpy
Flood_Modeling_Dataset_3	Nature	Hourly	607	1	28979647	[862, 113882]	48	168	-	0.394	0.05	0.034	0.0	0.07	Lumpy
SaugeenRiverFlow	Nature	Daily	1	1	23741	[23741, 23741]	14	30	60	0.544	0.029	0.053	0.0	0.002	Erratic
Sunspot	Nature	Daily	1	1	73924	[73924, 73924]	14	30	60	0.687	0.269	0.28	0.0	0.001	Erratic
USBirths	Nature	Daily	1	1	7305	[7305, 7305]	14	30	-	0.93	0.928	0.67	0.013	0.052	Smooth
Weather	Nature	Daily	2994	1	42172024	[989, 65981]	14	30	-	0.72	0.015	0.058	0.0	0.004	Smooth
Car_Parts	Retail	Monthly	267	1	1861289	[166, 13770]	12	30	-	0.505	0.001	0.224	0.0	0.256	Lumpy
M5	Retail	Daily	29897	1	46279060	[271, 1968]	60	-	-	0.554	0.087	0.251	0.0	0.062	Lumpy
favorita_stores_1D	Retail	Daily	1570	1	2367756	[114, 1688]	14	-	-	0.671	0.431	0.288	0.001	0.021	Smooth
favorita_stores_1M	Retail	Monthly	1466	1	73643	[37, 54]	8	-	-	0.709	0.764	0.407	0.211	0.068	Smooth
favorita_stores_1W	Retail	Weekly	1569	1	337858	[79, 240]	13	-	-	0.7	0.65	0.4	0.047	0.052	Smooth
favorita_transactions_1D	Retail	Daily	51	1	79389	[118, 1688]	14	-	-	0.738	0.597	0.193	0.0	0.006	Smooth
favorita_transactions_1M	Retail	Monthly	45	1	2415	[39, 54]	8	-	-	0.94	0.804	0.148	0.656	0.021	Smooth
favorita_transactions_1W	Retail	Weekly	50	1	11277	[97, 240]	13	-	-	0.829	0.676	0.273	0.013	0.052	Smooth
rossmann_1D	Retail	Daily	1115	1	1049806	[757, 942]	14	-	-	0.808	0.088	0.13	0.0	0.009	Smooth
rossmann_1W	Retail	Weekly	1115	1	148232	[108, 133]	13	-	-	0.537	0.275	0.226	0.0	0.015	Smooth
M1(Monthly)	various	Monthly	617	1	55998	[48, 150]	8	-	-	0.782	0.901	0.561	0.542	0.074	Smooth
M1(Quarterly)	various	Quarterly	172	1	9282	[36, 114]	8	-	-	0.743	0.983	0.728	0.843	0.167	Smooth
M1(Yearly)	various	Yearly	66	1	2242	[26, 58]	6	-	-	0.668	0.977	0.809	0.832	0.167	Smooth
M3(Monthly)	various	Monthly	1088	1	143988	[78, 144]	18	-	-	0.637	0.927	0.635	0.47	0.074	Smooth
M3(Quarterly)	various	Quarterly	697	1	35522	[35, 72]	8	-	-	0.665	0.984	0.673	0.749	0.167	Smooth
M3(Yearly)	various	Yearly	198	1	8329	[26, 47]	6	-	-	0.623	0.954	0.651	0.537	0.093	Smooth
M4(Daily)	various	Daily	4225	1	10022640	[76, 9933]	14	-	-	0.618	0.922	0.609	0.521	0.074	Smooth
M4(Hourly)	various	Hourly	414	1	373372	[748, 1008]	48	-	-	0.986	0.91	0.251	0.108	0.009	Smooth
M4(Monthly)	various	Monthly	4996	1	1164417	[78, 2812]	18	-	-	0.71	0.958	0.633	0.516	0.074	Smooth
M4(Quarterly)	various	Quarterly	4906	1	497181	[35, 874]	8	-	-	0.65	0.977	0.716	0.75	0.167	Smooth
M4(Weekly)	various	Weekly	359	1	371579	[93, 2610]	13	-	-	0.656	0.944	0.501	0.42	0.074	Smooth
M4(Yearly)	various	Yearly	4681	1	195371	[26, 585]	6	-	-	0.631	0.991	0.761	0.936	0.167	Smooth

GARCH (Bollerslev, 1986), *Kalman Filter* (Pei et al., 2019), and *HOLT* (Hyndman & Athanasopoulos, 2018). These models provide a baseline for understanding the inherent predictability of the data.

- ***Machine Learning Models***: This category involves **16** representative algorithms, including *RandomForest* (Breiman, 2001), *LightGBM* (Ke et al., 2017), *DecisionTree* (Breiman et al., 1984), *K-Neighbors* (Cunningham & Delany, 2021), *Extratrees* (Geurts et al., 2006), SVR (Basak et al., 2007), linear regression models like *Linear Regression*, *Lasso* (Tibshirani, 1996), *Ridge* (Hoerl & Kennard, 2000), *Huber* (Owen, 2006), *ElasticNet* (Mol et al., 2008), *SGDRegressor* (Liu et al., 2020), *BayesianRidge* (Tipping, 2001), *ARDRegression* (Wipf & Nagarajan, 2007), *Lars* (Efron et al., 2004), and boosting models such as XGBoost (Chen & Guestrin, 2016). These models test the archive's suitability for feature-based methods.

- ***Deep Learning Architectures***: We evaluate **47** state-of-the-art deep architectures, covering diverse structural designs such as Transformers (*Autoformer* (Wu et al., 2022), *PatchTST* (Nie et al., 2023), *Crossformer* (Zhang & Yan, 2023), *iTransformer* (Liu et al., 2024c), *Pyraformer* (Liu et al., 2022b), *Non-stationary Transformer* (Liu et al., 2023), *Reformer* (Kitaev et al., 2020), *Informer* (Zhou et al., 2021), *Transformer* (Vaswani et al., 2023), *FEDformer* (Zhou et al., 2022b), *TFT* (Lim et al., 2020), *ETSformer* (Woo et al., 2022), *CycleNet* (Lin et al., 2024a), *CycleiTransformer* (Lin et al., 2024a), *DUET* (Qiu et al., 2025b), *SimMTM* (Dong et al., 2023), *SimpleTM* (Chen et al., 2025), and *TimeXer* (Wang et al., 2024b)), Linear-based networks (*DLinear* (Zeng et al., 2022), *TSMixer* (Chen et al., 2023), *NLinear* (Zeng et al., 2022), *LDlinear* (Yu et al., 2024), *RMLP* (Li et al., 2023), *FreTS* (Yi et al., 2023), *TimeMixer* (Wang et al., 2024a), *PatchMLP* (Tang & Zhang, 2024), *FITS* (Xu et al., 2024), *TiDE* (Das et al., 2024a), *N-Beats* (Oreshkin et al., 2020), *SparseTSF* (Lin et al., 2024b)), convolution-based networks (*CNN* (O'Shea & Nash, 2015), *TimesNet* (Wu

Table 3: **Dataset statistics (Multivariate, MTS)**. We report the dataset domain, frequency, the number of time series, channels and observations, the length range of time series, and forecasting horizons (short-term, medium-term, long-term); We also report the median of a variety of time series patterns such as correlation across different channels, seasonality, trend, shift, stationarity, transition, and the most common demand pattern.

Dataset	Domain	Freq	#Series	#Channels	#Observations	Length Range	ShortH	MedH	LongH	Correlation	Seasonality	Trend	Shift	Stationarity	Transition	Demand Pattern
boomlet_1062	Cloud	every 1minute	1	21	344064	[16384, 16384]	60	240	720	0.663	0.567	0.294	0.094	0.037	0.04	Lumpy
boomlet_1209	Cloud	every 1minute	1	53	868352	[16384, 16384]	24	-	-	0.589	0.524	0.494	0.389	0.022	0.054	Lumpy
boomlet_1225	Cloud	every 1minute	1	49	802816	[16384, 16384]	60	240	720	0.649	0.499	0.392	0.238	0.006	0.029	Lumpy
boomlet_1230	Cloud	every 1minute	1	23	376832	[16384, 16384]	24	-	-	0.537	0.523	0.392	0.355	0.002	0.03	Lumpy
boomlet_1282	Cloud	every 1minute	1	35	573440	[16384, 16384]	24	-	-	0.671	0.469	0.346	0.375	0.073	0.027	Lumpy
boomlet_1487	Cloud	every 1minute	1	54	884736	[16384, 16384]	60	240	720	0.626	0.374	0.251	0.063	0.0	0.038	Lumpy
boomlet_1631	Cloud	every 1minute	1	40	418520	[10463, 10463]	60	240	720	0.594	0.512	0.663	0.418	0.014	0.068	Lumpy
boomlet_1676	Cloud	every 1minute	1	100	1046300	[10463, 10463]	60	240	720	0.644	0.473	0.541	0.313	0.068	0.04	Lumpy
boomlet_1855	Cloud	every 1minute	1	52	272012	[5231, 5231]	60	240	720	0.587	0.698	0.489	0.548	0.047	0.047	Lumpy
boomlet_1975	Cloud	every 1minute	1	75	392325	[5231, 5231]	24	-	-	0.698	0.771	0.554	0.294	0.019	0.015	Lumpy
boomlet_2187	Cloud	every 1minute	1	100	523100	[5231, 5231]	24	-	-	0.638	0.665	0.236	0.378	0.004	0.037	Lumpy
boomlet_285	Cloud	every 1minute	1	75	1228800	[16384, 16384]	60	240	720	0.706	0.628	0.084	0.225	0.0	0.155	Lumpy
boomlet_619	Cloud	every 1minute	1	52	851968	[16384, 16384]	60	240	720	0.832	0.78	0.178	0.146	0.0	0.032	Lumpy
boomlet_772	Cloud	every 1minute	1	67	1097728	[16384, 16384]	60	240	720	0.627	0.502	0.367	0.24	0.008	0.036	Lumpy
boomlet_963	Cloud	every 1minute	1	28	458752	[16384, 16384]	60	240	720	0.62	0.501	0.41	0.259	0.002	0.034	Lumpy
FRED-MD	Economics	Monthly	1	107	77896	[728, 728]	12	30	60	0.657	0.567	0.906	0.592	0.574	0.114	Erratic
NASDAQ	Economics	Daily	4818	6	105874440	[136, 14664]	14	30	-	0.914	0.674	0.806	0.653	0.351	0.069	Erratic
NYSE	Economics	Daily	500	5	4255690	[251, 1762]	14	30	-	0.94	0.576	0.809	0.575	0.486	0.081	Smooth
National_Stock_Exchange	Economics	Daily	3	12	7440	[248, 248]	14	30	-	0.828	0.547	0.666	0.416	0.182	0.036	Smooth
australian_tourism	Economics	Monthly	1	96	3456	[36, 36]	6	-	-	0.557	0.715	0.447	0.167	0.224	0.02	Smooth
gvar	Economics	Quarterly	26	6	27768	[178, 178]	8	-	-	0.563	0.678	0.892	0.674	0.416	0.127	Intermittent
Appliances_Energy_Prediction	Energy	every 10 minutes	1	28	552580	[19735, 19735]	48	144	576	0.71	0.649	0.697	0.532	0.03	0.067	Smooth
ETTH1	Energy	Hourly	1	7	121940	[17420, 17420]	48	168	336	0.606	0.731	0.853	0.205	0.001	0.02	Erratic
ETTH2	Energy	Hourly	1	7	121940	[17420, 17420]	48	168	336	0.508	0.615	0.906	0.404	0.022	0.042	Lumpy
ETTM1	Energy	every 15 min	1	7	487760	[69680, 69680]	32	96	384	0.599	0.76	0.614	0.203	0.0	0.027	Erratic
ETTM2	Energy	every 15 min	1	7	487760	[69680, 69680]	32	96	384	0.496	0.615	0.906	0.406	0.003	0.038	Lumpy
Power_Consumption_of_Tetouan_City	Energy	every 10 minutes	1	8	419328	[52416, 52416]	48	144	576	0.567	0.859	0.801	0.151	0.0	0.026	Erratic
Residential_Power_and_Battery_Data	Energy	every 1 minute	41	3	37647018	[30367, 500000]	120	240	-	0.518	0.597	0.24	0.234	0.0	0.002	Erratic
Spanish_energy	Energy	Hourly	1	26	911560	[35060, 35060]	48	168	336		0.335	0.245	0.283	0.0	0.03	Intermittent
Spanish_weather	Energy	Hourly	1	13	455832	[35064, 35064]	48	168	336	0.561	0.733	0.504	0.431	0.0	0.058	Lumpy
WIND	Energy	every 15 minutes	1	7	340711	[48673, 48673]	32	96	384	0.506	0.557	0.639	0.243	0.007	0.028	Smooth
electricity_2016	Energy	Hourly	1	321	8443584	[26304, 26304]	48	168	-	0.801	0.945	0.795	0.194	0.005	0.01	Smooth
electricityloaddiagrams20112014	Energy	every 15 min	1	370	51894720	[140256, 140256]	48	96	192	0.675	0.912	0.885	0.361	0.009	0.05	Lumpy
gasrate_co2	Energy	Daily	1	2	592	[296, 296]	14	30	-	1.0	0.826	0.554	0.495	0.037	0.044	Lumpy
gefcom2012_load	Energy	Daily	21	24	31752	[63, 63]	14	-	-	0.551	0.519	0.598	0.255	0.012	0.022	Smooth
gefcom2012_wind	Energy	Hourly	1	7	185983	[26569, 26569]	48	168	336	0.498	0.713	0.022	0.072	0.0	0.01	Erratic
COVID-19_Deaths	Healthcare	Daily	1	266	56392	[212, 212]	14	-	-		0.331	0.731	0.638	0.662	0.13	Lumpy
ILY	Healthcare	Hourly	1	7	6944	[992, 992]	12	24	60	0.693	0.748	0.646	0.698	0.169	0.038	Erratic
MIT-BIH	Healthcare	microsecond	20	2	4000000	[100000, 100000]	90	360	720	1.0	0.515	0.275	0.096	0.0	0.008	Lumpy
Sleep_EDFX	Healthcare	microsecond	171	6	96405040	[6937, 100000]	50	250	500	0.588	0.495	0.393	0.251	0.159	0.011	Lumpy
Bike_Sharing	Mobility	Hourly	1	11	191169	[17379, 17379]	48	168	336	0.539	0.727	0.489	0.186	0.0	0.074	Lumpy
METR-LA	Mobility	every 5 min	1	207	7094304	[34272, 34272]	48	144	288	0.789	0.49	0.107	0.03	0.0	0.004	Smooth
Metro_Interstate_Traffic_Volume	Mobility	Hourly	1	5	202875	[40575, 40575]	48	168	336	0.561	0.588	0.127	0.122	0.0	0.125	Lumpy
MexicoCityBikes	Mobility	every 1 minute	10	4	3360088	[21024, 141860]	60	240	720	0.554	0.38	0.145	0.013	0.0	0.001	Smooth
PEMS-BAY	Mobility	every 5 min	1	325	16941600	[52128, 52128]	48	144	-	0.844	0.618	0.1	0.091	0.0	0.005	Smooth
PEMS03	Mobility	every 5 min	1	358	9382464	[26208, 26208]	48	144	-	0.813	0.87	0.096	0.072	0.0	0.006	Smooth
PEMS04	Mobility	every 5 min	1	307	5216544	[16992, 16992]	48	144	-	0.809	0.854	0.077	0.085	0.0	0.007	Smooth
PEMS07	Mobility	every 5 min	1	883	24921792	[28224, 28224]	48	144	-	0.891	0.854	0.098	0.087	0.0	0.005	Smooth
PEMS08	Mobility	every 5 min	1	170	3035520	[17856, 17856]	48	144	-	0.812	0.85	0.112	0.095	0.0	0.005	Smooth
Rideshare	Mobility	Hourly	156	15	1246464	[541, 541]	12	24	60	0.732	0.7	0.816	0.443	0.263	0.073	Lumpy
Traffic	Mobility	Hourly	1	862	15122928	[17544, 17544]	48	96	168	0.814	0.88	0.263	0.171	0.0	0.011	Erratic
Traffic_Flow_Forecasting	Mobility	every 15 minutes	36	11	831996	[2101, 2101]	32	96	-	0.655	0.955	0.333	0.101	0.0	0.012	Smooth
vessel_ais	Mobility	every 7 minutes	40	8	20476928	[63885, 64000]	48	144	288	0.539	0.483	0.182	0.128	0.0	0.005	Smooth
Czelan	Nature	every 10 minutes	1	11	578171	[52561, 52561]	48	192	336	0.689	0.742	0.761	0.143	0.135	0.056	Lumpy
DailyDelhiClimate	Nature	Daily	1	4	6300	[1575, 1575]	14	30	60	0.588	0.8	0.149	0.07	0.039	0.017	Smooth
KDDCup2018	Nature	Hourly	54	6	2817230	[3056, 10905]	48	168	336	0.506	0.605	0.387	0.614	0.0	0.018	Erratic
Oikolab	Nature	Hourly	1	8	800456	[100057, 100057]	48	168	336	0.53	0.642	0.547	0.029	0.0	0.002	Smooth
Temperature-Rain	Nature	Daily	422	76	23252200	[725, 725]	14	30	60	0.73	0.552	0.131	0.238	0.013	0.066	Lumpy
Zafnoo_env	Nature	half hourly	1	11	211475	[19225, 19225]	48	192	336	0.599	0.765	0.686	0.147	0.002	0.03	Lumpy
Zafnoo_sapf	Nature	half hourly	1	6	112470	[18745, 18745]	48	192	336	0.509	0.803	0.637	0.26	0.0	0.041	Lumpy
beijing_multisite_air_quality	Nature	Hourly	12	12	5049216	[35064, 35064]	48	168	336		0.585	0.036	0.183	0.0	0.032	Lumpy
italy_air_quality	Nature	Hourly	1	13	121641	[9357, 9357]	48	168	336	0.562	0.599	0.611	0.296	0.0	0.029	Lumpy
weather_Wetterstation	Nature	every 10 minutes	1	21	1106784	[52704, 52704]	48	144	576	0.658	0.652	0.674	0.23	0.0	0.053	Lumpy
Dominick	Retail	Weekly	3889	86	64502697	[59, 399]	13	-	-	0.676	0.6	0.425	0.337	0.099	0.067	Lumpy

et al., 2023), *MICN* (Wang et al., 2023), *LightTS* (Zhang et al., 2022), *SCINet* (Liu et al., 2022a), *ResNet* (He et al., 2015), *FACT* (Wang et al., 2026), and *TCN* (Bai et al., 2018)), recurrent networks (*RNN* (Schmidt, 2019), *GRU* (Cho et al., 2014), *LSTM* (Hochreiter & Schmidhuber, 1997), *BiLSTM* (Quoc Nguyen et al., 2021), *Seq2Seq* (Sutskever et al., 2014), *SegRNN* (Lin et al., 2023)), State Space Models (*Mamba* (Gu & Dao, 2024) and *FiLM* (Zhou et al., 2022a)), and emerging Kolmogorov-Arnold Networks (*TimeKAN* (Huang et al., 2025)). This unprecedentedly comprehensive scope is intended to reduce systematic bias in favor of specific deep learning architectures.

- ***Time-Series Foundation Models***: To assess the frontier of the field, we incorporate **19** recently emerged foundation models, including *zero-shot* forecasters such as *MOIRAI v1* (Woo et al., 2024), *MOIRAI v2* (Liu et al., 2026), *Chronos t5* (Ansari et al., 2024), *Chronos bolt* (Ansari et al., 2024), *Chronos v2* (Ansari et al., 2025), *Timer* (Liu et al., 2024d), *SunDial* (Liu et al., 2025b), *Kairos* (Feng et al., 2026), *TiRex* (Auer et al., 2025), *TimesFM v1* (Das et al., 2024b), *TimesFM v2* (Das et al., 2024b), *TimesFM v2.5* (Das et al., 2024b), *Toto* (Cohen et al., 2024), *TabPFN* (Hollmann et al., 2023) and *few-shot* learners like *GPT4TS* (Zhou et al., 2023), *MOMENT* (Goswami et al., 2024), *TimeLLM* (Jin et al., 2024), *UniTS* (Gao et al., 2024), and *TEST* (Sun et al., 2024). This allows for a rigorous cross-comparison between domain-specific training and zero-shot transfer capabilities. For simplicity in subsequent sections, model names without explicit version suffixes refer to their latest versions (e.g., *Chronos* for v2, *TimesFM* for v2.5, and *MOIRAI* for v2).

Table 4: **18 Statistical forecasting models included in the benchmark.** We illustrate the type of model, the source of implementation, and key hyperparameters.

Model	Type	Source	Key Hyperparameters
naivemovingaverage	naive	Darts (Herzen et al., 2022)	`input_chunk_length=12`
naiveseasonal	naive	Darts (Herzen et al., 2022)	`K=12`
naivemean	naive	Darts (Herzen et al., 2022)	–
naivedrift	naive	Darts (Herzen et al., 2022)	–
autoregressive	auto-regressive	Statsmodels (Seabold & Perktold, 2010)	`lags=7`
arima	auto-regressive	Statsmodels (Seabold & Perktold, 2010)	`order=(1, 1, 1)`
sarima	auto-regressive	Statsmodels (Seabold & Perktold, 2010)	`seasonal_order=(1, 1, 1, 12)`
theta	decomposition	Statsmodels (Seabold & Perktold, 2010)	`period=12`
prophet	decomposition	Meta (Taylor & Letham, 2018)	`growth="linear"`
ets	exponential-smoothing	Statsmodels (Seabold & Perktold, 2010)	`error="add"`
simpleexpsmoothing	exponential-smoothing	Statsmodels (Seabold & Perktold, 2010)	`smoothing_level=0.6`
holt	exponential-smoothing	Statsmodels (Seabold & Perktold, 2010)	`initialization_method="estimated"`
croston	intermittent-demand	statsforecast (Nixtla, 2026)	`n_windows=4`
adida	intermittent-demand	statsforecast (Nixtla, 2026)	`n_windows=4`
imapa	intermittent-demand	statsforecast (Nixtla, 2026)	`n_windows=4`
arch	others	statsforecast (Nixtla, 2026)	–
garch	others	statsforecast (Nixtla, 2026)	–
kalmanfilter	others	KalmanFilter(Pei et al., 2019)	–

Table 5: **16 machine-learning-based forecasting models included in the benchmark.** We illustrate the type of model, the source of implementation, and key hyperparameters.

Model	Type	Source	Key Hyperparameters
decisiontree	tree-based	Sklearn (Pedregosa et al., 2011)	–
randomforest	tree-based	Sklearn (Pedregosa et al., 2011)	`n_estimators=100`
extratrees	tree-based	Sklearn (Pedregosa et al., 2011)	`n_estimators=100`
xgboost	boosting	Sklearn (Pedregosa et al., 2011)	`max_depth=6, learning_rate=0.1`
lightgbm	boosting	Sklearn (Pedregosa et al., 2011)	`lag=10`
linearregression	regression	Sklearn (Pedregosa et al., 2011)	–
lasso	regression	Sklearn (Pedregosa et al., 2011)	`alpha=0.1`
ridge	regression	Sklearn (Pedregosa et al., 2011)	`alpha=1.0`
elasticnet	regression	Sklearn (Pedregosa et al., 2011)	`alpha=0.1`
sgdregressor	regression	Sklearn (Pedregosa et al., 2011)	–
bayesianridge	regression	Sklearn (Pedregosa et al., 2011)	–
ardregressor	regression	Sklearn (Pedregosa et al., 2011)	`max_iter=300`
huber	regression	Sklearn (Pedregosa et al., 2011)	`lag=10`
lars	regression	Sklearn (Pedregosa et al., 2011)	`nonzero_coefs=10`
kneighbors	similarity	Sklearn (Pedregosa et al., 2011)	`n_neighbors=5, lag=10`
svr	kernel-based	Sklearn (Pedregosa et al., 2011)	`C=1.0, epsilon=0.1`

D FORECASTING EVALUATION METRICS

In this section, we go through additional widely used forecasting evaluation metric which are also reported in `TimeFCST` benchmark.

Mean Absolute Error (MAE) MAE measures the average absolute difference between the forecast and the ground truth. It is easy to interpret because it has the same unit as the original time series. A lower MAE indicates that the model's predictions are globally closer to the true values. Let y_i denote the ground-truth value, $\hat{y}_i$ denote the predicted value, and n denote the number of observations being predicted. MAE can be calculated as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{14}$$

Mean Squared Error (MSE) MSE measures the average squared forecasting error. Because the errors are squared, large mistakes would receive heavy penalties than small mistakes: therefore, MSE is a desired metric when large forecasting errors are strongly discouraged. See Eq. 14.

Mean Absolute Scaled Error (MASE) MASE measures the forecasting error relative to the error of a *naive seasonal* forecasting baseline (Hyndman & Koehler, 2006). Unlike percentage-based metrics such as MAPE, MASE is scale-independent and remains well-defined even when the time series contains zeros. Because of its robustness and comparability across datasets, MASE is widely used in large-scale forecasting benchmarks (Qiu et al., 2025a; Aksu et al., 2024; Shchur et al., 2026). Specifically, the metric is formulated as

$$\text{MASE} = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\frac{n}{T-m}\sum_{i=m+1}^{T}|y_i - y_{i-m}|}, \tag{15}$$

where m represents the seasonal period of the time series. Under this formulation, a normalized score smaller than 1 directly indicates that the forecasting model outperforms the seasonal naive baseline, while a value greater than 1 implies inferior performance.

Table 6: **47 deep learning models included in the benchmark.** 'CD' indicates channel dependence during multivariate time series forecasting.

Model	Type	CD	Source	Key Hyperparameters
autoformer	Transformer-based	✓	THUML (Wang et al., 2025)	e_layers=2, d_model=512
patch_tst	Transformer-based	✗	THUML (Wang et al., 2025)	e_layers=1, d_model=512
crossformer	Transformer-based	✓	THUML (Wang et al., 2025)	e_layers=2, factor=3
itransformer	Transformer-based	✓	THUML (Wang et al., 2025)	e_layers=2, des="Exp"
pyraformer	Transformer-based	✓	THUML (Wang et al., 2025)	e_layers=2, factor=3
nonstationary-transformer	Transformer-based	✓	THUML (Wang et al., 2025)	p_hidden_layers=2
reformer	Transformer-based	✗	THUML (Wang et al., 2025)	e_layers=2, factor=3
informer	Transformer-based	✓	THUML (Wang et al., 2025)	e_layers=2, factor=3
fedformer	Transformer-based	✓	THUML (Wang et al., 2025)	e_layers=2, moving_avg=25
etsformer	Transformer-based	✓	THUML (Wang et al., 2025)	e_layers=2, top_k=5
transformer	Transformer-based	✓	THUML (Wang et al., 2025)	e_layers=2, n_heads=8
cyclenet	Transformer-based	✗	CycleNet (Lin et al., 2024a)	e_layers=3, use_revin=True
cycleitransformer	Transformer-based	✓	CycleNet (Lin et al., 2024a)	e_layers=3, use_revin=True
tft	Transformer-based	✓	THUML (Wang et al., 2025)	e_layers=1, moving_avg=3
timexer	Transformer-based	✓	THUML (Wang et al., 2025)	e_layers=1, moving_avg=25
sim_mtm	Transformer-based	✗	SimMTM (Dong et al., 2023)	e_layers=2, factor=5
simpletm	Transformer-based	✗	SimpleTM (Chen et al., 2025)	wv="db2", factor=5
duet	Transformer-based	✓	DUET (Qiu et al., 2025b)	noisy_gating=True, moving_avg=25
dlinear	MLP-based	✗	THUML (Wang et al., 2025)	e_layers=2, moving_avg=25
tsmixer	MLP-based	✓	THUML (Wang et al., 2025)	e_layers=2, d_model=512
nlinear	MLP-based	✗	NLinear (Zeng et al., 2022)	enc_in=1, individual=True
ldlinear	MLP-based	✗	CycleNet (Lin et al., 2024a)	enc_in=1
rmlp	MLP-based	✗	CycleNet (Lin et al., 2024a)	–
frets	MLP-based	✗	THUML (Wang et al., 2025)	–
patchmlp	MLP-based	✗	PatchMLP (Tang & Zhang, 2024)	e_layers=4, d_model=512
fits	MLP-based	✗	FITS (Xu et al., 2024)	H_order=6, loss="MSE"
tide	MLP-based	✓	THUML (Wang et al., 2025)	downsampling_layers=3
nbeats	MLP-based	✗	N-Beats (Oreshkin et al., 2020)	nb_blocks_per_stack=3
sparse_tsf	MLP-based	✗	SparseTSF (Lin et al., 2024b)	e_layers=2, factor=3
timemixer	MLP-based	✓	THUML (Wang et al., 2025)	downsampling_window=2
cnn	CNN-based	✓	MLC (Persson, 2026)	–
lightts	CNN-based	✗	THUML (Wang et al., 2025)	chunk_size=24
tcn	CNN-based	✓	TCN (Bai et al., 2018)	kernel_size=3, num_filters=3
micn	CNN-based	✓	THUML (Wang et al., 2025)	e_layers=2
scinet	CNN-based	✓	THUML (Wang et al., 2025)	e_layers=2
resnet	CNN-based	✓	ResNet (He et al., 2015)	num_layers=50
fact	CNN-based	✓	FACT (Wang et al., 2026)	dilation=[1, 2], num_kernels=6
timesnet	CNN-based	✓	THUML (Wang et al., 2025)	top_k=5, num_kernels=3
rnn	RNN-based	✓	MLC (Persson, 2026)	hidden_size=1024
gru	RNN-based	✓	MLC (Persson, 2026)	hidden_size=1024
lstm	RNN-based	✓	MLC (Persson, 2026)	num_fc_layers=2
bilstm	RNN-based	✓	THUML (Wang et al., 2025)	hidden_size=1024
seq2seq	RNN-based	✓	Seq2Seq (Sutskever et al., 2014)	num_kernels=3
segrnn	RNN-based	✓	THUML (Wang et al., 2025)	downsampling_window=1
mamba	SSM	✓	Mamba (Gu & Dao, 2024)	expand=2, d_conv=4
film	SSM	✗	THUML (Wang et al., 2025)	moving_avg=25
timekan	KAN	✗	TimeKAN (Huang et al., 2025)	downsampling_window=2

Table 7: **19 foundation models included in the benchmark.** We illustrate the type of model, the source of implementation, and key hyperparameters.

Model	Type	CD	Source	Key Hyperparameters
moirai_v1	zero-shot forecaster	✓	MOIRAI v1 (Woo et al., 2024)	model_size="base", patch_size=8
moirai	zero-shot forecaster	✗	MOIRAI v2 (Liu et al., 2026)	model_size="small", patch_size=8
chronos_t5	zero-shot forecaster	✗	Chronos (Ansari et al., 2024)	quantile_levels=[0.5]
chronos_bolt	zero-shot forecaster	✗	Chronos (Ansari et al., 2024)	quantile_levels=[0.5]
chronos	zero-shot forecaster	✓	Chronos (Ansari et al., 2025)	quantile_levels=[0.5]
timer	zero-shot forecaster	✗	Timer (Liu et al., 2024d)	max_context_length=2800
sundial	zero-shot forecaster	✗	SunDial (Liu et al., 2025b)	num_samples=20
kairos	zero-shot forecaster	✗	Kairos (Feng et al., 2026)	preserve_positivity=False
tirex	zero-shot forecaster	✓	TiRex (Auer et al., 2025)	–
timesfm_v1	zero-shot forecaster	✗	TimesFM v1 (Das et al., 2024b)	num_layers=20
timesfm_v2	zero-shot forecaster	✗	TimesFM v2 (Das et al., 2024b)	num_layers=50
timesfm	zero-shot forecaster	✗	TimesFM (Das et al., 2024b)	max_context=1024
toto	zero-shot forecaster	✓	Toto (Cohen et al., 2024)	num_samples=32
tabpfn	zero-shot forecaster	✓	TabPFN (Hollmann et al., 2023)	–
gpt4ts	few-shot learner	✓	GPT4TS (Zhou et al., 2023)	freeze=False, pretrain=True
moment	few-shot learner	✗	MOMENT (Goswami et al., 2024)	patch_len=8
timellm	few-shot learner	✗	TimeLLM (Jin et al., 2024)	llm_model="BERT"
units	few-shot learner	✓	UniTS (Gao et al., 2024)	prompt_num=10
test	few-shot learner	✓	TEST (Sun et al., 2024)	use_encoder=True, patch_size=11

Modified Symmetric Mean Absolute Percentage Error (MSMAPE) MSMAPE is a modified version of SMAPE designed to improve numerical stability. By adding a small constant ϵ to the denominator, it avoids excessive penalties when both the true and predicted values are very small. This makes it more suitable for sparse, intermittent, or low-volume time series.

$$\text{MSMAPE} = \frac{100}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i| + \epsilon}. \tag{16}$$

Root Mean Squared Error (RMSE) RMSE is the square root of MSE. Like MAE, it is expressed in the same unit as the original series, but it still penalizes large errors more strongly than MAE. A lower RMSE indicates better overall forecasting accuracy, and it can be calculated as below:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}. \tag{17}$$

Mean Absolute Percentage Error (MAPE) MAPE measures the average absolute error as a percentage of the true value. It is scale-independent and easy to compare across datasets. However, it becomes unstable when the true value is near zero. MAPE is calculated as follows:

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \tag{18}$$

Symmetric Mean Absolute Percentage Error (SMAPE) SMAPE is a percentage-based error metric that normalizes the absolute error by the average magnitude of the true and predicted values. Compared with MAPE, it reduces the asymmetry between over-forecasting and under-forecasting. However, it can still be ill-defined when both the true and predicted values are close to zero.

$$\text{SMAPE} = \frac{100}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2}. \tag{19}$$

Weighted Absolute Percentage Error (WAPE) WAPE measures the total absolute forecasting error relative to the total magnitude of the observed values. It is often more stable than MAPE because it aggregates errors before normalization, useful for datasets with multiple small or zero values.

$$\text{WAPE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} |y_i|}. \tag{20}$$

E TIMEFCST BENCHMARK DETAILS

E.1 DATA PREPARATION AND QUALITY ASSURANCE

To facilitate a comprehensive evaluation of time-series forecasting, we initiated an extensive data acquisition process aimed at capturing the vast diversity inherent in real-world temporal dynamics. We initially assembled a massive repository of raw time-series candidates in `TimeFCST`, spanning a broad spectrum of domains including energy, cloud computing, healthcare, nature, mobility, economics, and retail. These sources encompass a wide range of temporal resolutions—from high-frequency seconds to yearly intervals—challenging a model's adaptability across varying scales and complex season patterns in real data.

Following this initial collection, we first conducted and implemented a rigorous preliminary check process to ensure the basic integrity of the benchmark. This stage involved the systematic removal of "low-quality" datasets characterized by an excessive portion of missing values, severe temporal misalignment across multivariate channels, or purely stochastic properties (e.g., *white noise* and *random walk signals*), which are not suitable for model evaluation. After this preliminary filtering, we maintained a high-quality archive of 128 datasets with a total scale of *over one billion* observations, balanced between 64 univariate (UTS) and 64 multivariate (MTS) datasets. While this version represents an expansive and diverse archive for broad exploration, it serves as the foundation for the subsequent refined and curated 1B-scale "clean" version used in our primary benchmarking.

To streamline the utilization of such a large-scale archive, all datasets have been unified into a standardized tabular format. Considering the heterogeneous original source or file structures, each dataset in `TimeFCST` is transformed into a uniform schema with clearly distinguishable fields for Sample ID, Timestamp, and Channel identifiers. This unified representation significantly reduces the engineering overhead for data loading and preprocessing, enabling researchers to efficiently navigate this multi-billion scale archive without the burden of complex data wrangling.

E.2 DATA CURATION AND QUALITY CONTROL

As outlined in Section E.1, we implement a rigorous, multi-stage curation and cleaning pipeline to ensure the integrity and predictability of the `TimeFCST` archive. This process culminates in a ***one-billion-scale*** curated dataset, serving as a comprehensive and ready-to-use archive for large-scale research. Given the inherent heterogeneity of large data sources, we systematically address several critical issues commonly observed in public datasets through an automated screening process, complemented by meticulous human inspection. Our pipeline specifically targets: (i) Missing Value and Irregular Sampling Handling, (ii) Anomaly and Artifact Mitigation, (iii) Stochasticity and Random Walk Detection, and (iv) Constant and Inactive Region Filtering. Detailed descriptions of each stage are provided below. Examples of time-series are shown in Figure 2.

Missing Value and Irregular Sampling Handling: Missing values in time-series data often arise from various real-world constraints, such as sensor malfunctions, network transmission interruptions, or maintenance-related downtime. While certain advanced architectures possess an inherent capacity

to process missing data, the majority of conventional and contemporary forecasting methodologies operate under the assumption of continuous and complete input sequences. To bridge this gap, we carefully handle missing entries while maintaining experimental integrity. Specifically, we employ a forward-fill imputation method for datasets with minor gaps, as it effectively preserves the temporal continuity of the signal while strictly preventing temporal data leakage—ensuring that information from the future does not leak to past observations. We discard datasets with such high levels of missing data to preserve a strong signal-to-noise ratio in the benchmark.

Furthermore, we rigorously verify the temporal regularity of the sampled intervals. To ensure a standardized and reliable baseline, the vast majority of the `TimeFCST` archive is curated to be regularly sampled, facilitating the evaluation of models that rely on consistent seasonal patterns. However, we acknowledge that irregular sampling is a prevalent challenge in event-driven recording protocols. Instead of forcing these sequences into a uniform grid through aggressive interpolation—which might distort the ground-truth seasonality—we explicitly categorize a specialized subset that retains its original non-uniform interval nature. This subset allows for a targeted assessment of model robustness against irregular temporal dynamics without compromising the overall consistency. Finally, for both regular and irregular cases, we perform cross-channel temporal alignment so that all channels are synchronized across timestamps, resulting in a standardized data loading pipeline.

Anomaly and Artifact Mitigation: While the majority of real-world temporal signals exhibit relatively smooth transitions, we observed non-negligible anomalies across various source data, often attributable to sensor malfunctions or data logging artifacts. A recurring debate in the forecasting community is whether models should be trained to be robust against such noise; however, the presence of these anomalies can significantly distort the overall evaluation.

Based on this, we implement a systematic protocol to detect and mitigate these artifacts. To ensure objectivity, we leverage an ensemble of three widely-used anomaly detection algorithms, including KMeansAD, KNNAD, and HYDRA (Liu & Paparrizos, 2024; Huang et al., 2026) and flag sequences that exhibit high consensus anomaly scores. We acknowledge that the precise identification of anomalies often necessitates substantial domain-specific prior knowledge, as certain "anomalous" fluctuations may represent rare but legitimate physical events. Consequently, our mitigation efforts focus primarily on "spiky" anomalies—extreme deviations (e.g., erroneous values like $-9,999$ or unrealistic spikes) that are highly likely to be the result of sensor failure or a data-entry error. Once identified under such automated screening process, these localized artifacts are treated using the same imputation rationale applied to missing values to ensure temporal consistency. Finally, to prevent the unintended removal of critical signals, all flagged anomalies undergo a final round of meticulous human inspection. This approach ensures that our cleaning actions are both necessary and justified.

Stochasticity and Random Walk Detection: To ensure the benchmark provides meaningful signals for evaluating forecasting efficacy, we systematically exclude datasets that exhibit purely stochastic or unpredictable behavior. Specifically, we target two categories: (1) White Noise, identified via the Ljung-Box test (Ljung & Box, 1978); these sequences lack any temporal autocorrelation, meaning past observations provide zero information about future values. (2) Random Walk processes, characterized by the form $y_t = y_{t-1} + \epsilon_t$, where ϵ_t is a noise term. Such a process is fundamentally unpredictable because its best unbiased estimator is simply the most recent observation (the naive forecast). We employ the Augmented Dickey-Fuller (ADF) test complemented by the Variance Ratio (VR) test.

We validate all flagged candidates by running both naive and SOTA models, ensuring that the final removed series cannot offer genuine predictive challenges for benchmarking purpose. Consequently, we remove datasets with a high proportion of random walk sequences, such as the ExchangeRate dataset, which is widely recognized as an unsuitable benchmark dataset (Lo & MacKinlay, 1988).

Constant and Inactive Region Filtering: In contrast to stochastic signals that lack predictability, certain sequences exhibit a lack of temporal variance, rendering them "too simple" to provide meaningful evaluative value. This category primarily includes constant and inactive regions—extended intervals where the signal remains unchanged. Such phenomena are typically artifacts of sensor cold-starts, device malfunctions, or periods of system inactivity rather than legitimate seasonal patterns (e.g., zero-demand during scheduled off-hours). If left unaddressed, these regions can bias evaluation metrics by artificially deflating error rates, as even a naive model can achieve near-zero error.

To mitigate this, we rigorously scan the archive for datasets characterized by long inactive "heads" or "tails." We perform precision trimming on these sequences, removing the stagnant prefixes or suffixes while preserving the high-quality, dynamic segments in the middle. Furthermore, we examine series with a significant portion of inactive intervals in their interior. For these cases, we evaluate the structural integrity of the remaining data: if the "active" segments before or after the gap are of sufficient length and quality, they are retained as independent sub-series; otherwise, the entire flagged series is discarded to prevent models from learning or being evaluated on trivial, non-representative patterns. This curation ensures that the benchmark focuses exclusively on active, dynamic temporal behaviors that challenge a model's performance.

Consistent with the aforementioned quality control philosophy, all sequences processed through the automated filters—whether involving imputation, trimming, or removal—undergo a final stage of meticulous human inspection to ensure the automated operations accurately preserve the underlying physical semantics of the data.

E.3 Failure Mode and Failure Profile Details

Our failure profile consists of six modes: *level*, *trend*, *amplitude*, *phase*, *localized error*, and *spectral collapse*. Let $\mathbf{Y}_i = (Y_{i,1}, \dots, Y_{i,H})$ and $\hat{\mathbf{Y}}_{m,i} = (\hat{Y}_{m,i,1}, \dots, \hat{Y}_{m,i,H})$ denote the ground truth and the prediction of model m for forecasting instance i, respectively. We use mean absolute error (MAE):

$$e_{m,i} = \ell(\mathbf{Y}_i, \hat{\mathbf{Y}}_{m,i}) = \frac{1}{H} \left\| \mathbf{Y}_i - \hat{\mathbf{Y}}_{m,i} \right\|_1 . \tag{21}$$

For each failure mode j, we define an oracle repair operator $\mathcal{C}_j$ that corrects the corresponding discrepancy while preserving the remaining prediction as much as possible:

$$\hat{\mathbf{Y}}^{(j)}_{m,i} = \mathcal{C}_j \left(\hat{\mathbf{Y}}_{m,i}, \mathbf{Y}_i \right), \qquad e^{(j)}_{m,i} = \ell \left(\mathbf{Y}_i, \hat{\mathbf{Y}}^{(j)}_{m,i} \right). \tag{22}$$

Level. Level failure represents a constant vertical offset between the prediction and the ground truth. Let $\mathbf{1} \in \mathbb{R}^H$ be the all-ones vector. We estimate the MAE-optimal additive correction

$$b^\star_{m,i} = \arg\min_{b \in \mathbb{R}} \ell \left(\mathbf{Y}_i, \hat{\mathbf{Y}}_{m,i} + b\mathbf{1} \right), \tag{23}$$

which, for a single forecasting window, is the median prediction residual:

$$b^\star_{m,i} = \mathrm{median}_t \left(Y_{i,t} - \hat{Y}_{m,i,t} \right). \tag{24}$$

The repaired prediction is

$$\hat{\mathbf{Y}}^{(\mathrm{level})}_{m,i} = \hat{\mathbf{Y}}_{m,i} + b^\star_{m,i}\mathbf{1}. \tag{25}$$

Trend. Trend failure describes an incorrect temporal slope or low-order evolution after the level discrepancy has been removed. Let

$$u_t = \frac{t - (H+1)/2}{H}, \qquad \mathbf{u} = (u_1, \dots, u_H), \tag{26}$$

be a centered time basis. Starting from the level-repaired prediction, the linear trend coefficient is

$$a^\star_{m,i} = \arg\min_{a \in \mathbb{R}} \ell \left(\mathbf{Y}_i, \hat{\mathbf{Y}}^{(\mathrm{level})}_{m,i} + a\mathbf{u} \right), \tag{27}$$

and the joint level-and-trend repair is

$$\hat{\mathbf{Y}}^{(\mathrm{level+trend})}_{m,i} = \hat{\mathbf{Y}}^{(\mathrm{level})}_{m,i} + a^\star_{m,i}\mathbf{u}. \tag{28}$$

In the implementation, To separate its effect from that of Level, Trend is measured as the additional recovery after Level failure correction:

$$\mathrm{RS}_{m,i,\mathrm{trend}} = \left[\frac{e^{(\mathrm{level})}_{m,i} - e^{(\mathrm{level+trend})}_{m,i}}{e_{m,i} + \epsilon} \right]_+ . \tag{29}$$

This conditional definition prevents the same level discrepancy from being attributed to both Level and Trend in the final error analysis.

Amplitude. Amplitude failure occurs when the prediction captures the temporal structure but underestimates or overestimates its variation magnitude. Let $\mathcal{L}(\mathbf{x})$ denote the least-squares linear component of $\mathbf{x}$ over the basis $(\mathbf{1}, \mathbf{u})$. We decompose the prediction into

$$\hat{\mathbf{Y}}_{m,i} = \mathcal{L}(\hat{\mathbf{Y}}_{m,i}) + \mathbf{r}_{m,i}, \qquad \mathbf{r}_{m,i} = \hat{\mathbf{Y}}_{m,i} - \mathcal{L}(\hat{\mathbf{Y}}_{m,i}). \tag{30}$$

We then estimate a nonnegative scale coefficient

$$s^{\star}_{m,i} = \arg\min_{s \geq 0} \ell\left(\mathbf{Y}_i, \mathcal{L}(\hat{\mathbf{Y}}_{m,i}) + s\mathbf{r}_{m,i}\right), \tag{31}$$

and define

$$\hat{\mathbf{Y}}^{(\text{amplitude})}_{m,i} = \mathcal{L}(\hat{\mathbf{Y}}_{m,i}) + s^{\star}_{m,i}\mathbf{r}_{m,i}. \tag{32}$$

Because the linear component is held fixed, this operator changes the variation magnitude without directly correcting level or trend.

Phase. Phase failure represents a temporal misalignment between otherwise similar forecast and ground-truth shapes. Let $\mathcal{S}_\tau(\cdot)$ denote an integer shift by τ time steps with endpoint padding. To avoid selecting a shift from level or scale differences, we first construct a standardized detrended temporal shape as follows:

$$\mathcal{Z}(\mathbf{x}) = \frac{\mathbf{x} - \mathcal{L}(\mathbf{x})}{\text{std}\left(\mathbf{x} - \mathcal{L}(\mathbf{x})\right) + \epsilon}. \tag{33}$$

The shift is selected within a restricted search range:

$$\tau^{\star}_{m,i} = \arg\min_{|\tau| \leq \lfloor 0.1H \rfloor} \ell\left(\mathcal{Z}(\mathbf{Y}_i), \mathcal{S}_\tau\left(\mathcal{Z}(\hat{\mathbf{Y}}_{m,i})\right)\right). \tag{34}$$

The repaired prediction is evaluated on the original scale:

$$\hat{\mathbf{Y}}^{(\text{phase})}_{m,i} = \mathcal{S}_{\tau^{\star}_{m,i}}\left(\hat{\mathbf{Y}}_{m,i}\right). \tag{35}$$

We exclude improvements that arise solely from endpoint padding rather than from better alignment in the interior of the forecasting window.

Localized Error. Localized error captures a forecasting failure concentrated within a short contiguous interval (we set 5% of the entire horizon window as a default range). Let

$$L_i = \max\left(1, \lfloor 0.05H \rfloor\right) \tag{36}$$

be the local repair budget. We identify the interval with the largest accumulated absolute error:

$$g^{\star}_{m,i} = \arg\max_{1 \leq s \leq H - L_i + 1} \sum_{t=s}^{s+L_i-1} \left| Y_{i,t} - \hat{Y}_{m,i,t} \right|, \tag{37}$$

and define

$$\hat{Y}^{(\text{localized})}_{m,i,t} = \begin{cases} Y_{i,t}, & g^{\star}_{m,i} \leq t < g^{\star}_{m,i} + L_i, \\ \hat{Y}_{m,i,t}, & \text{otherwise.} \end{cases} \tag{38}$$

Thus, the operator measures how much error can be removed by correcting only a small localized region. To prevent accidental error concentration from receiving excessive credit, the implementation calibrates this recovery against a permutation-based null distribution.

Spectral Collapse. Spectral collapse occurs when the ground truth contains substantial detrended variation but the prediction collapses toward a nearly flat or purely linear trajectory. Define

$$\mathbf{v}_i = \mathbf{Y}_i - \mathcal{L}(\mathbf{Y}_i), \qquad \hat{\mathbf{v}}_{m,i} = \hat{\mathbf{Y}}_{m,i} - \mathcal{L}(\hat{\mathbf{Y}}_{m,i}). \tag{39}$$

The operator is activated only when the ground truth is non-flat and the prediction-to-ground-truth variation ratio is sufficiently small:

$$\frac{\text{std}(\hat{\mathbf{v}}_{m,i})}{\text{std}(\mathbf{v}_i) + \epsilon} \leq \gamma, \tag{40}$$

where γ is a strict collapse threshold. We compute the discrete Fourier transform of $\mathbf{v}_i$, retain at most the three largest nonzero-frequency components, and reconstruct the corresponding signal $\mathbf{q}_{m,i}$. Its contribution is controlled by

$$\alpha^{\star}_{m,i} = \arg \min_{0 \le \alpha \le 1} \ell \left(\mathbf{Y}_i, \hat{\mathbf{Y}}_{m,i} + \alpha \mathbf{q}_{m,i} \right). \tag{41}$$

The repaired prediction is

$$\hat{\mathbf{Y}}^{(\text{spectral})}_{m,i} = \hat{\mathbf{Y}}_{m,i} + \alpha^{\star}_{m,i} \mathbf{q}_{m,i}. \tag{42}$$

If the collapse condition is not satisfied, the mode is assigned zero recovery.

F Experimental Setting Details

In this section, we introduce the experiment settings applied in this benchmark study, including the platform and implementation, statistical test and model training and evaluation details.

Platform and Implementation. We use a cluster equipped with 2xAMD EPYC 7713 64-Core processors on the Ubuntu 22.04.3 LTS operating system with 4 NVIDIA A100 80GB GPU cards to conduct our benchmarking experiments. We implement all forecasting models in Python 3.11 to ensure the fairness and consistency of evaluation. We also list the primary dependencies as follows: torch 2.4.1, numpy 1.26.4, pandas 2.1.4, scikit-learn 1.4.2, statsmodels 0.14.1, and scipy 1.11.4. To promote the reproducibility of our evaluation results and analysis, we release our source code repository and collected datasets (git, 2026). For all model experiments, we run 3 random seeds and take average to reduce the effect of randomness. For each model, we perform hyper-parameter searches across multiple sets, with a limit of 4 sets to improve evaluation efficiency.

Statistical Test. To more reliably measure each model's overall performance, following (Demšar, 2006; Paparrizos & Gravano, 2015; Yang & Paparrizos, 2025), we employ Friedman test (Friedman, 1937) and post-hoc Nemenyi test (Nemenyi, 1963) with 95% confidence level. In this paper, we present the average ranking results with Critical Difference (CD) diagrams (shown in Figure 5 and 13), where models connected by a solid horizontal line indicates there is no significant performance difference between compared methods.

Model Training and Evaluation. To facilitate a rigorous benchmark study, we employ full-shot training for all task-specific models. However, to balance computational efficiency with the demands of full-shot fine-tuning for foundation models, we report zero-shot performance where publicly accessible (e.g., via public model weights or APIs). Otherwise, if fine-tuning was supported at the time of benchmark preparation, we report 10% few-shot performance of each model.

G Experimental Analysis Details

This section provides additional experimental details and analyses that could not be fully included in the main text due to space constraints, offering further evidence and insights to support our primary findings and future research in this direction.

Global view of Model Performance across Categories. Our analysis begins with Univariate Time Series (UTS) forecasting, providing a granular comparison of traditional and modern methodologies. A distinguishing feature of our study is the inclusion of a vast array of statistical and machine learning (ML) methods, which are commonly overlooked in contemporary forecasting literature. As illustrated in the Critical Difference (CD) diagram for the UTS top-performing statistical/ML methods (Figure 14 (a) and (b)), the majority of these classical approaches significantly outperform the *Naiveseasonal* baseline across the 64 UTS datasets. Notably, regression-based methods and robust estimators such as *Huber* exhibit a statistically significant performance gain over the naive baseline. This observation is further corroborated by the performance distribution in Figure 5(a.1), where two statistical and ML models successfully secure positions within the top 30 out of the 100 evaluated algorithms on our global leaderboard, outperforming a wide range published forecasting methods and establishing a strong baseline in the univariate forecasting scenarios.

When evaluating all categories (Figure 14(c) and (d)), Foundation Models (FMs) and Deep Learning (DL) architectures—such as *TimesFM*, *TiReX*, and *Chronos*, followed by *FACT*, *Timexer* and

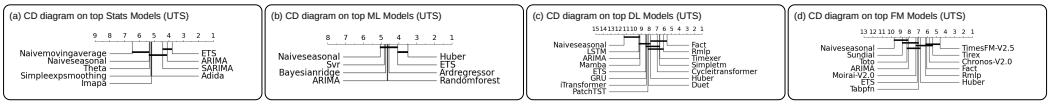

Figure 14: UTS Performance comparison across top models from each model family.

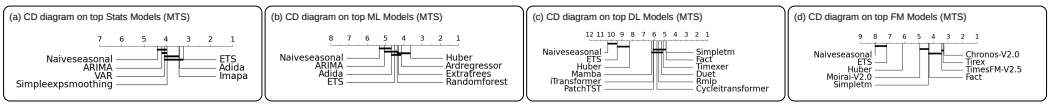

Figure 15: MTS Performance comparison across top models from each model family.

Rmlp—occupy the leading positions, though many top-tier deep learning-based models do not significantly outperform the best classical ML methods (e.g., *Huber*) in univariate settings, while only top FM models (e.g., *TimesFM*) show the statistical difference in performance (Figure 14(c-d)). Conversely, the landscape shifts in Multivariate Time Series (MTS) forecasting (Figure 14, where FMs and DL models exhibit a clear, statistically significant advantage over classical statistical and machine learning methods. As previously illustrated in Section 5.1, this performance gap stems from modern architectures' superior ability to handle high-dimensional and complex temporal dependencies—properties that are highly prevalent in MTS datasets. Notably, these top-performing FMs achieve their rankings in a zero-shot setting.

The Undervalued Efficacy of Simple Baselines. A major limitation of current forecasting benchmarks lies in their systemic omission of well-established statistical methods and simpler deep learning architectures, which are often prematurely dismissed in favor of overly complex models. However, a global, cross-domain view of our results reveals that these approaches remain exceptionally simple, computationally efficient, and highly competitive. Traditional statistical models like *ETS*, *ARIMA*, and robust machine learning estimators like *Huber* consistently provide strong, reliable baselines across diverse scenarios (especially in UTS setting), serving as valuable reference points that prevent overestimating the progress of more complex architectural designs in applications (Figure 5(a.1)).

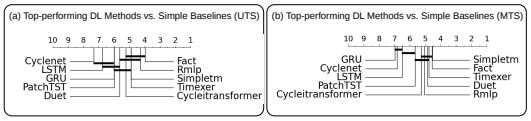

Figure 13: Forecasting performance comparison of the top DL methods against simple MLP and RNN baselines. Statistical testing indicates that despite architectural innovation, many state-of-the-art learning-based methods do not significantly outperform RNN/MLP + RevIN with proper hyperparameter tuning, particularly in univariate settings.

This oversight is not restricted to statistical models alone. As illustrated in the Critical Difference diagram (Figure 13), several highly publicized, state-of-the-art architectures struggle to strongly outperform a basic MLP- or RNN-based model when coupled with simple, proven design components—such as Reversible Instance Normalization (RevIN) to handle distribution shift and proper parameter tuning (Kim et al., 2021; Zhao et al., 2025; Nie et al., 2023; Li et al., 2023). Despite its simplicity and effectiveness, this class of modified classic networks is widely neglected in contemporary literature. Ultimately, these findings underscore that a truly reliable evaluation framework must not only track the frontier of foundation models but also actively maintain these straightforward, highly efficient baselines to measure genuine, incremental algorithmic progress.

Domain-Specific Efficacy and Feature Sensitivity. As previously discussed in Section 5.2, model dominance is highly domain-dependent. Such performance variation of different categories of methods can be explained by the rich diversity of data patterns in `TimeFCST`. Figures 11 and 12 further illustrate the remarkable heterogeneity in dataset collections. Figure 11 presents representative examples spanning classical temporal properties (e.g., seasonality, trend, shifting, stationarity, and transition) as well as diverse demand behaviors (smooth, erratic, intermittent, and lumpy), highlighting the wide spectrum of forecasting challenges encountered in practice. Figure 12 also complements these examples quantitatively by showing that such characteristics are unevenly distributed across domains. Broadly, human-centric and operational domains (e.g., Cloud, Economics, and Mobility) tend to exhibit smoother temporal dependencies with strong trends or erratic behaviors at relatively fine temporal resolutions, whereas resource- and demand-driven domains (e.g., Energy, Retail, and Nature) are dominated by lumpy demand patterns and greater variability in temporal dynamics.

In contrast, Healthcare datasets stand out as an extreme case, consisting primarily of ultra-high-frequency observations with sparse behaviors. Collectively, these results demonstrate that TimeFCST captures substantial diversity not only in temporal patterns, but also in sampling and domain-specific forecasting challenges. Furthermore, the performance degradation and shifts in model rankings across certain data domains reveal a failure to capture specific data patterns, offering valuable insights to guide future methodological improvements of task-specific or foundation models.

To further interpret these dynamics, Figure 16 examines model sensitivity under different data characteristics. The ridge plots show that forecasting errors can shift substantially between low- and high-regime datasets, but the direction and magnitude of these shifts vary across model families. In particular, high transition and non-stationary regimes tend to induce broader or more right-skewed error distributions for several model categories, indicating increased risk of large forecasting errors. This effect is especially visible for some machine learning methods, while foundation models and deep learning models generally exhibit more concentrated error distributions but are still affected by challenging dynamics. Seasonality and trend also lead to heterogeneous behavior: they do not uniformly benefit or degrade all models, suggesting that their impact depends on both the data regime and the modeling assumptions. Overall, these results reinforce that model performance is strongly conditioned on temporal characteristics, and such feature-level sensitivities can be obscured by aggregate benchmark scores.

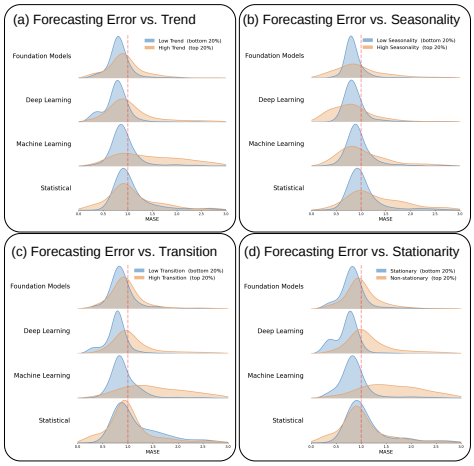

Figure 16: Performance sensitivity to data characteristics, visualized by ridge plots, where magnitude of forecasting performance degradation varies across model families.

Horizons Meet Data Dynamics. Figure 7 further shows that forecasting degradation across horizons is strongly shaped by temporal data patterns and model-family characteristics. Under high trend and seasonality, foundation models maintain strong robustness as the horizon increases, suggesting their ability to exploit recurring structures over longer prediction windows. However, this advantage does not transfer uniformly to all regimes. In high-trend and non-stationary datasets, most model families exhibit increasing forecasting errors as the horizon becomes longer. This degradation is more pronounced for machine learning and statistical methods, which deteriorate further over longer prediction windows given complex data patterns. This pattern helps explain their lower average rankings across varying datasets versus horizons from the global evaluation perspective. Foundation models and deep learning models are generally more stable, but they also suffer from clear long-horizon degradation under challenging dynamics. More broadly, future forecasting evaluation should not focus solely on increasingly long horizons or aggregated accuracy scores. Instead, benchmarks are suggested to account for the intrinsic challenges induced by data characteristics, such as seasonality, trend, and non-stationary behaviors.

Diverse Dataset Coverage Is Essential for Reliable Channel-Dependency Evaluation. Following the discussion in Section 5.2, we revisit the long-standing discussion on Channel-Independent (CI) versus Channel-Dependent (CD) modeling strategies for multivariate time-series forecasting. Building on prior studies that relate the effectiveness of cross-channel modeling to dataset characteristics (Qiu et al., 2025a; Zhao et al., 2025), our benchmark extends this analysis to a larger and more diverse evaluation space, providing a clearer picture across domains and data characteristics regimes. We select top CD and CI models respectively and calculate the performance difference across all 64 MTS datasets in TimeFCST. Figure 7(a) shows that the relative advantage of CD models is not uniform across datasets or application domains. For example, mobility datasets such as Traffic and PEMS07 exhibit a more visible CD advantage, suggesting that explicitly modeling inter-channel interactions can be beneficial when channels are strongly coupled. In contrast, other domains show much smaller differences between CI and CD strategies, indicating that the need for comprehensive cross-channel modeling highly depends on the dataset characteristics.

Figure 7(a) further shows this observation to inter-channel correlation. The CI–CD MASE difference increases with channel correlation, suggesting that CD models become more advantageous when the

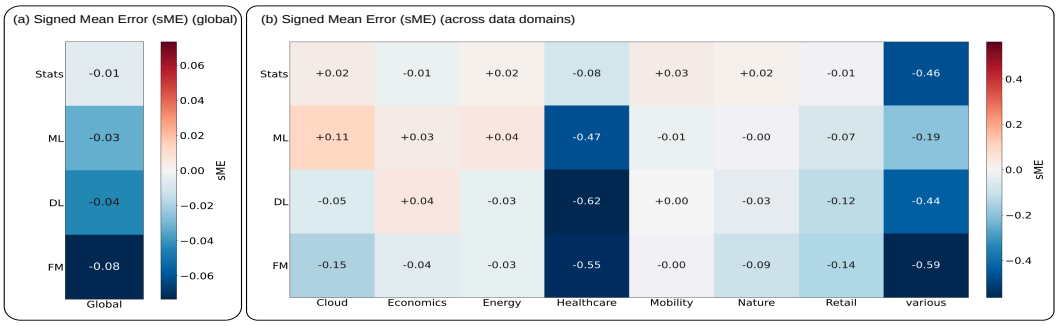

Figure 17: Forecast bias across model families and domains, measured by signed Mean Error (sME). Positive and negative values indicate the tendency of overprediction and underprediction, respectively. Across model families, negative sME values reveal a general tendency toward underprediction especially for deep learning and foundation models, while bias patterns vary substantially across diverse data domains in the benchmark.

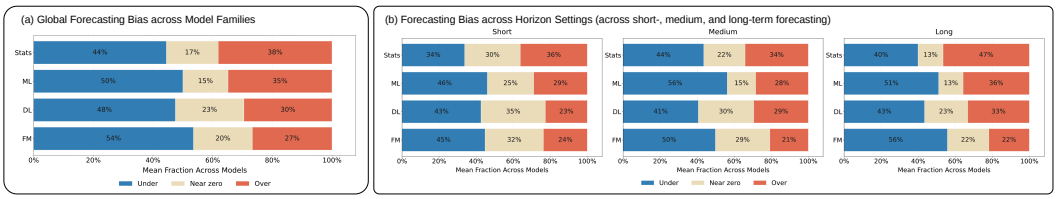

Figure 18: Forecast bias across model families and forecasting horizons, measured by the proportions of underprediction, near-zero bias, and overprediction. Foundation models, especially, exhibit an increasingly strong tendency toward underprediction as the forecasting horizon lengthens.

channels contain stronger shared structure. Meanwhile, several commonly used MTS benchmark datasets, such as ETTh, ETTm variants, and Weather, are highlighted in the plot and lie in regions where the CI–CD performance gap is negligible. This implies that repeatedly debating channel dependency on a narrow set may provide limited diagnostic insight. To address this limitation, `TimeFCST` includes a broad collection of datasets spanning diverse temporal patterns and channel-dependency structures. Across the full MTS suite, CD modeling outperforms its CI counterpart on 44 of the 64 datasets, revealing statistically significant differences among channel-modeling strategies (Figure 7(a.2–3)). This coverage enables a systematic assessment of when CD modeling is genuinely beneficial. These findings underscore that reliable MTS benchmarks should explicitly account for inter-channel dependency structures and include sufficiently diverse datasets to support meaningful methodological comparison and development.

Failure Profiles Reveal Distinct Patterns of Model Errors. As shown in Figure 9, Level mismatch is the dominant failure mode overall, accounting for 31.1% of the profile, followed by Phase (21.9%), Localized Error (15.3%), and Amplitude (13.9%). However, these profiles vary substantially across model families and forecasting settings. Statistical methods are particularly prone to Spectral Collapse (33.5%), producing overly smooth forecasts that fail to preserve important frequency components, whereas ML methods are dominated by Level mismatch (40.4%). DL and FM models exhibit more balanced profiles; notably, FMs distribute their errors primarily across Level, Phase, and Localized Error, indicating that their strong aggregate performance does not eliminate diverse structural weaknesses. Across domains, Phase errors are most prominent in Energy, Mobility, and Nature, consistent with their strong periodic and seasonal dynamics, while Level mismatch dominates Retail, Economics, and Various. Failure patterns also evolve with the forecasting horizon: Phase errors decrease from 24.8% to 16.0%, whereas Localized Error increases sharply

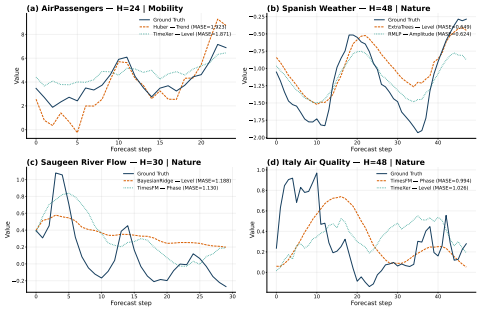

Figure 20: Example of distinct failure modes. Similar metric performance can exhibit different error patterns.

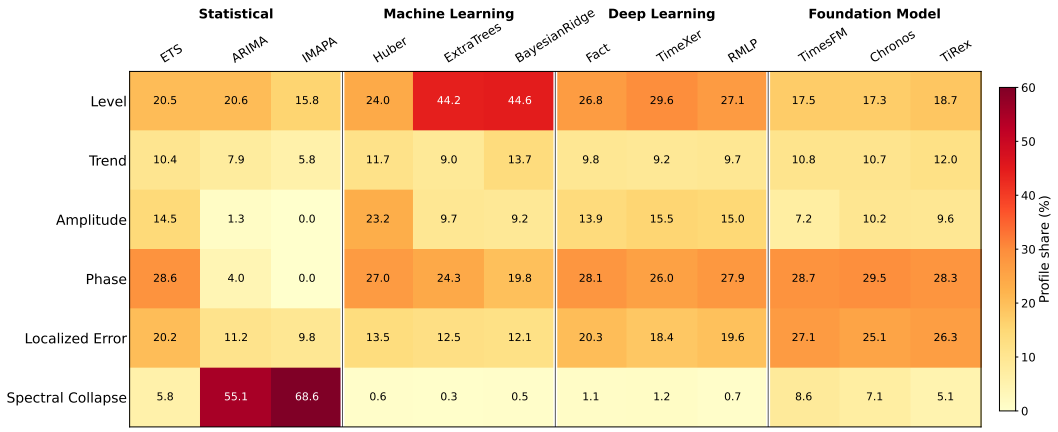

Figure 19: Failure profile analysis of top models.

from 4.1% to 30.3%, suggesting that long-term forecasts increasingly mismatch transient events and local temporal structures compared to shorter horizon settings.

Because the Level profile is direction-agnostic, we further conduct a complementary forecast-bias analysis using signed Mean Error (sME), where positive and negative values indicate overprediction and underprediction, respectively. As shown in Figures 17 and 18, the global sME becomes progressively more negative from statistical (-0.01) and ML (-0.03) methods to DL (-0.04) and FM (-0.08) models, revealing a stronger tendency toward underprediction among modern models. This tendency is particularly pronounced in Healthcare and Various, although substantial domain-level heterogeneity remains. Using thresholds of -5% and $+5\%$, we further categorize forecasts as underprediction, near-zero bias, or overprediction. FMs exhibit the clearest horizon-dependent pattern: their underprediction proportion increases from 45% at short horizons to 50% and 56% at medium and long horizons, respectively. Together, these results complement aggregate accuracy with actionable diagnostics, highlighting which structural errors dominate and whether their level-related discrepancies systematically underestimate or overestimate future values.

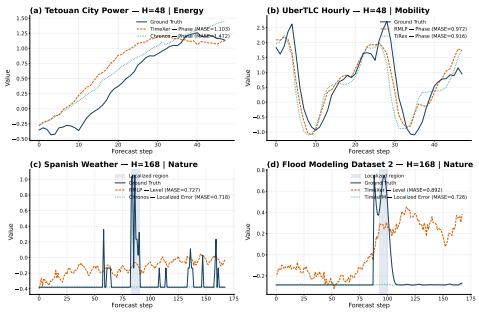

Figure 21: Examples of dominant failure modes in leading DL and FM models, primarily involving level, phase, and localized failure modes in the forecasting windows.

Beyond family-level aggregates, Figure 19 reveals substantial model-level heterogeneity within the same category, particularly among statistical and ML methods. For example, ARIMA and IMAPA are dominated by Spectral Collapse, whereas ETS exhibits more Phase and Localized errors; top DL and FM models show comparatively consistent profiles. Figure 20 further demonstrates that forecasts with comparable MASE can exhibit qualitatively different error patterns, confirming that aggregate metrics alone cannot adequately characterize how models fail. Representative predictions in Figure 21 illustrate the dominant errors of leading DL and FM models: although they often capture the overall shape and periodic structure, temporal misalignment produces Phase errors, while missed spikes and transient events lead to Localized errors. These results highlight the value of failure profiles for distinguishing model weaknesses hidden behind similar aggregate performance.

Forecasting Model Progress in the past Decades. Figure 8 and 22 trace model performance by publication year across representative datasets from each domain, revealing markedly different trajectories of methodological progress. Several widely used datasets appear saturated: performance on the *ETT* datasets in Energy and *Weather* in Nature has largely plateaued, with recent model generations providing limited improvements over earlier approaches. More counterintuitively, datasets such as *Ausbeer* in Economics and *Flood Modeling* in Nature exhibit signs of performance regression, with newer models often producing higher forecasting errors. In contrast, datasets such as *Boomlet* in

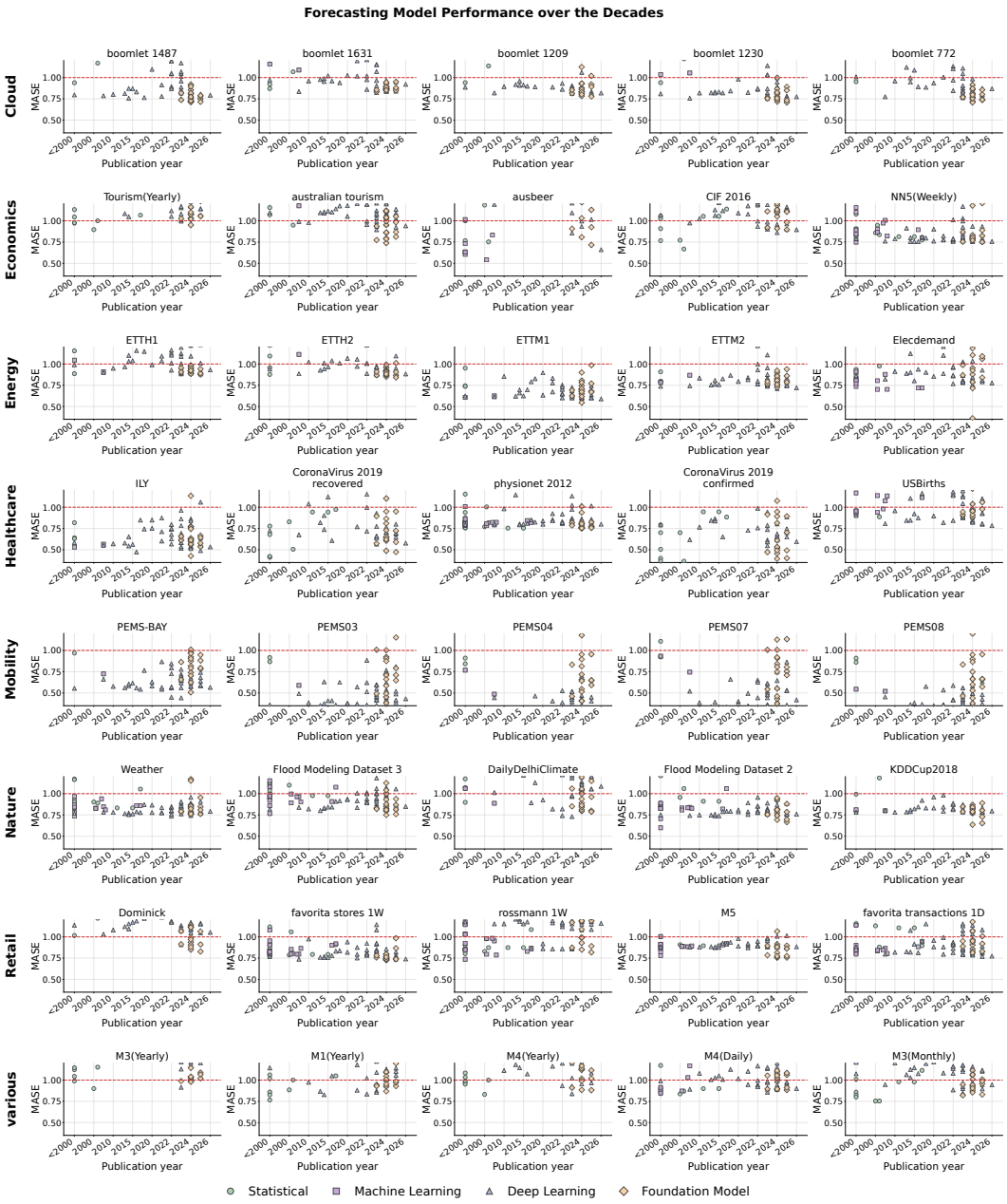

Figure 22: Forecasting model performance over time (5 example datasets for each domain). Datasets exhibit distinct patterns of progress across model families. Continued from Figure 8.

Cloud and *Dominick* in Retail show consistent improvements across successive model generations, from classical statistical and ML methods to DL and recently emerged FMs, highlighting the benefits of global representation learning and cross-dataset pretraining. These contrasting trajectories demonstrate that model publication year or architectural novelty does not translate uniformly into improved forecasting performance. Overreliance on saturated or unrepresentative datasets can therefore obscure genuine advances or create misleading impressions of progress, underscoring the need for broad and diverse benchmark coverage.