TimeCopilot

Azul Garza
San Francisco, CA, USA
azul.garza.r@gmail.com

Renée Rosillo
San Francisco, CA, USA
reneerosillo@protonmail.com

Abstract

We introduce TimeCopilot, the first open-source agentic framework for forecasting that combines multiple Time Series Foundation Models (TSFMs) with Large Language Models (LLMs) through a single unified API. TimeCopilot automates the forecasting pipeline: feature analysis, model selection, cross-validation, and forecast generation, while providing natural language explanations and supporting direct queries about the future. The framework is LLM-agnostic, compatible with both commercial and open-source models, and supports ensembles across diverse forecasting families. Results on the large-scale GIFT-Eval benchmark show that TimeCopilot achieves state-of-the-art probabilistic forecasting performance at low cost. Our framework provides a practical foundation for reproducible, explainable, and accessible agentic forecasting systems.

1 Introduction

The forecasting community has seen a rapid growth of TSFMs [1]. These models are typically pre-trained on large collections of heterogeneous time series and then adapted to generate forecasts on unseen datasets [2–4]. So far, we have found approaches spanning a diverse set of architectures: non-Transformer models such as Tiny Time Mixers [5]; encoder-only designs including Moment [6] and Moirai [7]; decoder-only architectures such as Timer-XL [8], Time-MOE [9], ToTo [10], Timer [11], TimesFM [12], and Lag-Llama [13]; approaches that adapt pretrained LLMs to time series forecasting, including Chronos [14], AutoTimes [15], LLMTime [16], Time-LLM [17], and FPT [18]; and models deployed as a web API such as TimeGPT-1 [19]. Each line of work emphasizes different inductive biases and pretraining strategies, showing strong performance on benchmarks while highlighting trade-offs in scalability, interpretability, and robustness.

While this proliferation of models has expanded the options available to a seasoned forecaster when determining the best approach for a given use case, it has also increased complexity: since a particular laboratory or research group develops each model, each comes with its own API, training pipeline, and evaluation conventions. Beyond these differences, models also vary in their data input requirements (e.g., univariate vs. multivariate, patch vs. non-patch tokenization) and in their learning curve complexity, with some demanding extensive compute and expertise to deploy or use effectively. This fragmentation makes it difficult to compare models fairly, and even harder to integrate them seamlessly into production forecasting systems [20]. Recent large-scale evaluations such as GIFT-Eval [21] highlight these challenges by showing in practice how different design choices across foundation models complicate reproducibility and cross-benchmark comparisons.

Meanwhile, the broader AI community has witnessed an explosion in the use of LLMs to orchestrate tasks across domains [22, 23]. In particular, the agentic paradigm, where an LLM serves as a controller that plans, reasons, and executes tasks [24], has been increasingly adopted in applications spanning text, code, image, video, and audio generation [25]. Agents allow end-users to specify high-level goals in natural language, while the underlying system orchestrates specialized models and tools to achieve them.

NeurIPS 2025 Workshop on Recent Advances in Time Series Foundation Models (BERT^{2}S).

Within time series, however, agentic approaches are still very new. For example, [26] introduces a scalable benchmarking framework to evaluate AI agents in machine learning engineering tasks related to time series, such as data handling and code translation, rather than forecasting itself. In parallel, multimodal efforts such as [27] highlight another emerging intersection between time series and LLMs: integrating numerical signals with natural language context to produce more reliable forecasts. Together, these efforts underscore both the promise and the nascency of agentic and LLM-augmented approaches in the forecasting domain. Despite these first steps, no prior work has unified TSFMs under an agentic interface.

In this paper, we introduce `TimeCopilot`[1], the first open-source framework designed specifically for agentic forecasting, see Figure 1. `TimeCopilot` is a generative forecasting agent where LLMs act as reasoning engines to orchestrate multiple TSFMs, as well as classical statistical, machine learning, and deep learning methods, through a single unified API. This design eliminates fragmented dependencies across research groups and automates the full forecasting pipeline, from data preparation and model selection to ensembling and evaluation, while providing natural language explanations and supporting direct queries about the future. By bridging the agent paradigm with a unified hub of forecasting models, `TimeCopilot` provides a practical path toward reproducible, interactive, and accessible agentic forecasting workflows.

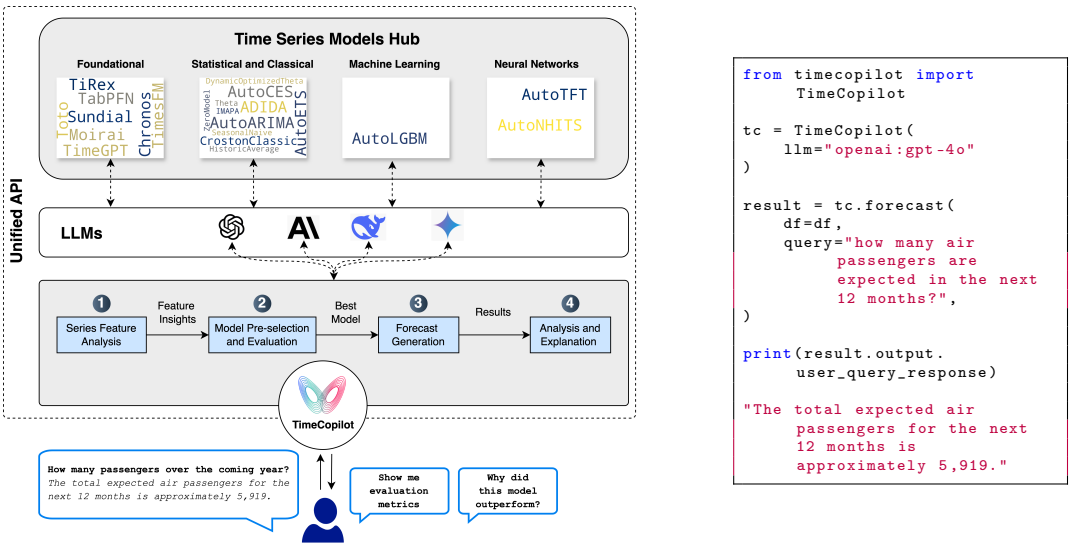

Figure 1: (Right) Overall `TimeCopilot`'s architecture. (Left) `TimeCopilot` Agent API usage.

2 Design Principles

`TimeCopilot` was designed to automate classical forecasting pipelines [28, 29]. Traditionally, a forecaster explores a time series, extracts relevant information (features, plots, or diagnostics), formulates hypotheses about which models may perform well, and then uses cross-validation and benchmarking to select the final model [30]. This cycle of exploration, hypothesis, and empirical validation has long been considered the cornerstone of forecasting practice.

`TimeCopilot` leverages LLMs in two ways: (i) to decide how to act at each step of this process, using their capacity to orchestrate multiple tools and reason over diverse signals, and (ii) to help forecasters explain both the decisions made during model selection and the forecasts themselves in natural language, making the results transparent and accessible. This combination of automation and explainability makes `TimeCopilot` the first agentic interface tailored to time series forecasting.

As Figure 1 shows, the framework is LLM-agnostic: it can interface with commercial APIs (e.g., OpenAI [22], Anthropic) as well as open-source models such as DeepSeek [31] or LLaMA [32], giving users flexibility to choose models according to cost, availability, or deployment constraints.

[1] `TimeCopilot` is available via `uv add timecopilot` and documentation at `https://timecopilot.dev`.

2.1 `TimeCopilot` Agent

The `TimeCopilot` **Agent** orchestrates the forecasting workflow in three structured steps: (i) **Time Series Feature Analysis:** computes focused diagnostics (trend, seasonality, stationarity) that directly inform model choice [33, 34] (ii) **Model Selection and Evaluation:** proposes candidate models starting from simple statistical baselines, documents their assumptions, evaluates them through cross-validation, and escalates to more complex models only when needed. [29] (iii) **Final Model Selection and Forecasting:** chooses the best model based on metrics, generates forecasts while interpreting patterns, uncertainty, and reliability. Beyond automating these steps, the agent is designed for explainability: users can query not only the forecasts but also the reasoning behind each decision. This transparency differentiates `TimeCopilot` from black-box agentic systems and is critical for trust in forecasting applications.

2.2 `TimeCopilot` Forecaster

The `TimeCopilot` **Forecaster** executes the models proposed by the agent. It provides access to the **largest unified hub of TSFMs under a single API** [2], alongside statistical, machine learning, and deep learning methods. This hub removes the need for users to manage conflicting dependencies or learn multiple fragmented interfaces. The current set of supported TSFMs includes Chronos [14], Chronos-2 [35], FlowState [36], Moirai [7], Sundial [37], TabPFN [38], TiRex [39], TimeGPT [19], TimesFM [12], and Toto [10].

Alongside foundation models, `TimeCopilot` integrates a wide suite of **statistical baselines** such as ADIDA, AutoARIMA, AutoETS, Theta, and SeasonalNaive [40], as well as Prophet [41]. For **machine learning methods**, `TimeCopilot` includes AutoLGBM [42]. For **neural networks**, it incorporates AutoNHITS [43] and AutoTFT [44] via the NeuralForecast library [45].

Finally, `TimeCopilot` supports **ensemble techniques**, such as MedianEnsemble, which can combine forecasts from heterogeneous models (statistical, ML, and foundation) to improve robustness [46]. This breadth enables direct comparison and flexible combinations, ensuring forecasters can benchmark and deploy the most suitable approach with minimal effort.

3 Benchmarks and API Usage Example

To illustrate the capabilities of `TimeCopilot` we provide both benchmark results and minimal usage examples.

3.1 Benchmarks on GIFT-Eval

Figure 2 summarizes the mean ranks and scores of foundation models on the GIFT-Eval benchmark [21], which spans 24 datasets, 144k+ time series, and 177M data points across multiple domains and frequencies. The top row reports the average rank across datasets for probabilistic accuracy (CRPS; [47]) and point accuracy (MASE; [48]), while the bottom row shows the corresponding mean scores. We include only methods submitted to GIFT-Eval that are publicly reproducible[3] . An ensemble of foundation models built with `TimeCopilot` achieved the best overall performance while maintaining reproducibility and low computational cost (approximately $24 of GPU-distributed inference).

The `TimeCopilot` results rely on its MedianEnsemble feature [4]. [49, 46], used to combine three state-of-the-art foundation models (Chronos-2 [35], TimesFM [12], and TiRex [39]) with an isotonic regression [50] to ensure monotonic quantiles for probabilistic forecasting. This approach provides robustness against outliers and model-specific biases. [5]

These results highlight three aspects of `TimeCopilot`: (i) It can orchestrate multiple foundation models through a unified interface. (ii) It provides state-of-the-art results in both probabilistic and point forecasting. (iii) It enables reproducible and cost-effective large-scale experimentation.

[2] A complete and regularly updated list of models is available at `https://timecopilot.dev/model-hub/`

[3] Live leaderboard and details available at `https://huggingface.co/spaces/Salesforce/GIFT-Eval`

[4] `https://timecopilot.dev/api/models/ensembles`

[5] The full experiment can be reproduced at `https://timecopilot.dev/experiments/gift-eval`.

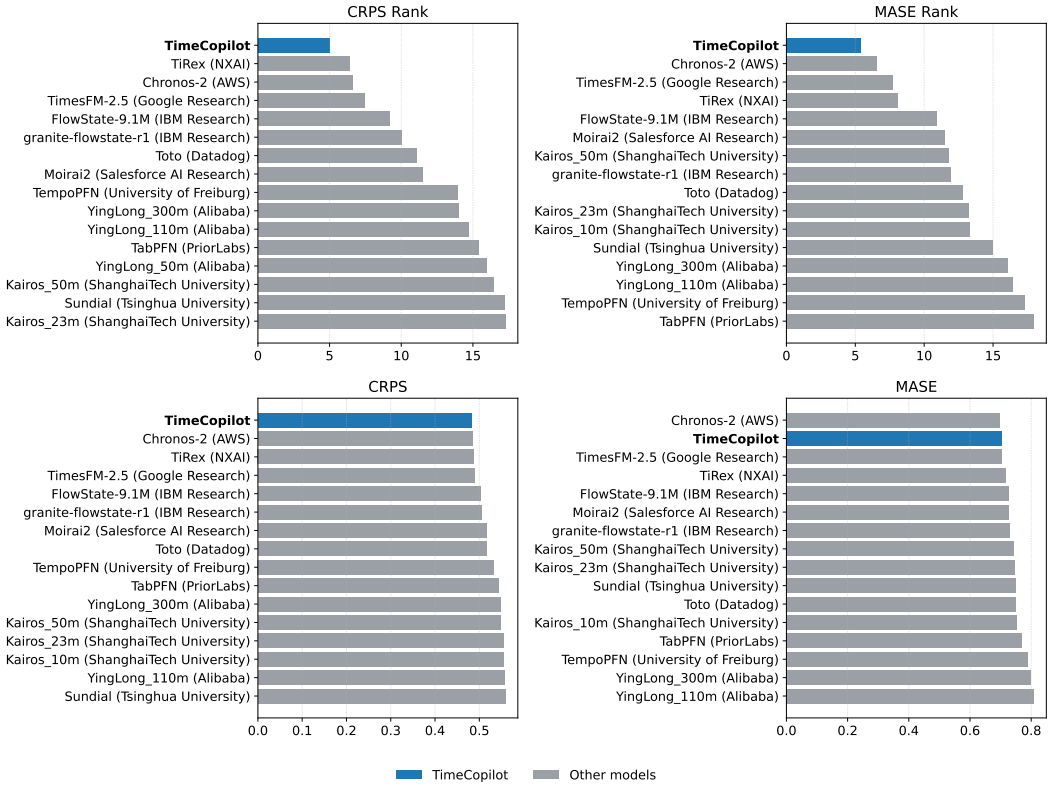

Figure 2: Performance of `TimeCopilot` and baseline models on the GIFT-Eval benchmark [21]. Lower values indicate better forecast performance. The top-16 models are shown, for each metric (CRPS and MASE). The top row shows the mean ranks across datasets, while the bottom row shows the corresponding mean scores.

3.2 API Usage Example

`TimeCopilot` provides two main entry points: the **Agent** (end-to-end orchestration with natural language explanations and queries about the future) see Figure 1, and the **Forecaster** (direct control of specific models under a unified API), see Appendix 5 for more details. This simple interface allows users to switch seamlessly between agent-driven automation and manual benchmarking of specific models, while keeping the same data structure and evaluation utilities.

4 Conclusion and Future Work

We presented `TimeCopilot`, the first open-source agentic interface and framework that combines multiple TSFMs with LLMs to make the forecasting workflow explainable, transparent, and accessible to a wider audience. `TimeCopilot` addresses the challenge of unifying diverse forecasting models under a single API, while leveraging LLMs to automate pipeline decisions and provide natural language explanations for both model choices and forecasts.

`TimeCopilot`'s future work encompasses: i) **Integrating with Model Context Protocol (MCP)** enabling seamless interaction with external tools and services through standardized interfaces, ii) **Expanding use cases** beyond academic benchmarks to domains such as energy, climate, finance, and supply chain forecasting, and iii) **Extending forecasting capabilities** to support hierarchical forecasting, multivariate forecasting with multivariate explanations, and advanced evaluation settings such as coherence across multiple aggregation levels.

Acknowledgments

We would like to express our gratitude to Martín Juárez, Rodrigo Mendoza-Smith, and Salomón Márquez for their helpful contributions and support.

References

[1] Siva Rama Krishna Kottapalli, Karthik Hubli, Sandeep Chandrashekhara, Garima Jain, Sunayana Hubli, Gayathri Botla, and Ramesh Doddaiah. Foundation models for time series: A survey, 2025. URL https://arxiv.org/abs/2504.04011.

[2] Qianli Ma, Zhen Liu, Zhenjing Zheng, Ziyang Huang, Siying Zhu, Zhongzhong Yu, and James T. Kwok. A survey on time-series pre-trained models, 2024. URL https://arxiv.org/abs/2305.10716.

[3] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. Meta-learning framework with applications to zero-shot time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 9242–9250, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17115/16922.

[4] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022. URL https://arxiv.org/abs/2108.07258.

[5] Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H. Nguyen, Wesley M. Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series, 2024. URL https://arxiv.org/abs/2401.03955.

[6] 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.

[7] 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.

[8] Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting, 2025. URL https://arxiv.org/abs/2410.04803.

[9] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts, 2025. URL https://arxiv.org/abs/2409.16040.

[10] 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.

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

[12] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023. URL https://arxiv.org/abs/2310.10688.

[13] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Hassen, Anderson Schneider, et al. Lag-llama: Towards foundation models for time series forecasting. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023. URL https://arxiv.org/abs/2310.08278.

[14] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar 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, 2024. URL https://arxiv.org/abs/2403.07815.

[15] Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models, 2024. URL https://arxiv.org/abs/2402.02370.

[16] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters, 2024. URL https://arxiv.org/abs/2310.07820.

[17] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023. URL https://arxiv.org/abs/2310.01728.

[18] Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Pretrained transformers as universal computation engines, 2021. URL https://arxiv.org/abs/2103.05247.

[19] Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. Timegpt-1. 2023. URL https://arxiv.org/abs/2310.03589.

[20] Christoph Bergmeir. LLMs and Foundational Models: Not (Yet) as Good as Hoped. *Foresight: The International Journal of Applied Forecasting*, (73):33–38, Q2 2024. URL https://cbergmeir.com/papers/Bergmeir2024LLMs.pdf.

[21] 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.

[22] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[24] Junwei Liu, Kaixin Wang, Yixuan Chen, Xin Peng, Zhenpeng Chen, Lingming Zhang, and Yiling Lou. Large language model-based agents for software engineering: A survey, 2024. URL https://arxiv.org/abs/2409.02977.

[25] Ranjan Sapkota, Konstantinos I. Roumeliotis, and Manoj Karkee. Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges, 2025. URL https://arxiv.org/abs/2505.10468.

[26] Yifu Cai, Xinyu Li, Mononito Goswami, Michał Wiliński, Gus Welter, and Artur Dubrawski. Timeseriesgym: A scalable benchmark for (time series) machine learning engineering agents, 2025. URL https://arxiv.org/abs/2505.13291.

[27] Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, and Alexandre Drouin. Context is key: A benchmark for forecasting with essential textual information, 2025. URL https://arxiv.org/abs/2410.18959.

[28] J S Armstrong, editor. *Principles of forecasting: a handbook for researchers and practitioners*. Kluwer Academic Publishers, 2001. ISBN 0792379306. URL http://amazon.com/dp/0792379306.

[29] Rob J. Hyndman, George Athanasopoulos, Azul Garza, Cristian Challu, Max Mergenthaler, and Kevin G. Olivares. *Forecasting: Principles and Practice, the Pythonic Way*. OTexts, Melbourne, Australia, 2024. URL https://otexts.com/fppy. Accessed on 22 August 2025.

[30] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: 100,000 time series and 61 forecasting methods. volume 36, pages 54–74, 2020. doi: https://doi.org/10.1016/j.ijforecast.2019.04.014. URL https://www.sciencedirect.com/science/article/pii/S0169207019301128. M4 Competition.

[31] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong

Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

[32] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

[33] Azul Garza, Kin Gutierrez, Cristian Challu, Jose Moralez, Ricardo Olivares, and Max Mergenthaler. tsfeatures: Time series feature extraction. https://github.com/Nixtla/tsfeatures, 2023. Accessed: 2025-08-22.

[34] Rob Hyndman, Yanfei Kang, Pablo Montero-Manso, Mitchell O'Hara-Wild, Thiyanga Talagala, Earo Wang, and Yangzhuoran Yang. *tsfeatures: Time Series Feature Extraction*, 2024. URL https://pkg.robjhyndman.com/tsfeatures/. R package version 1.1.1.9000, https://github.com/robjhyndman/tsfeatures.

[35] 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.

[36] Lars Graf, Thomas Ortner, Stanisław Woźniak, and Angeliki Pantazi. Flowstate: Sampling rate invariant time series forecasting, 2025. URL https://arxiv.org/abs/2508.05287.

[37] 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, 2025. URL https://arxiv.org/abs/2502.00816.

[38] Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: How tabpfn-v2 outperforms specialized time series forecasting models, 2025. URL https://arxiv.org/abs/2501.02945.

[39] 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.

[40] Azul Garza, Max Mergenthaler Canseco, Cristian Challú, and Kin G. Olivares. StatsForecast: Lightning fast forecasting with statistical and econometric models. PyCon Salt Lake City, Utah, US 2022, 2022. URL https://github.com/Nixtla/statsforecast.

[41] S J Taylor and B Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018. doi: 10.1080/00031305.2017.1380080.

[42] Nixtla contributors. mlforecast: Scalable machine learning for time series forecasting. https://github.com/Nixtla/mlforecast, 2023. Accessed: 2025-08-22.

[43] Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 6989–6997, 2023. URL https://ojs.aaai.org/index.php/AAAI/article/view/25854.

[44] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764, 2021.

[45] Kin G. Olivares, Cristian Challú, Azul Garza, Max Mergenthaler Canseco, and Artur Dubrawski. NeuralForecast: User friendly state-of-the-art neural forecasting models. PyCon Salt Lake City, Utah, US 2022, 2022. URL https://github.com/Nixtla/neuralforecast.

[46] J M Bates and C W J Granger. The combination of forecasts. *Operational Research Quarterly*, 20(4):451–468, 1969. doi: 10.1057/jors.1969.103.

[47] T Gneiting and M Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and its Application*, 1(1):125–151, 2014. doi: 10.1146/annurev-statistics-062713-085831.

[48] R J Hyndman and A B Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006. doi: 10.1016/j.ijforecast.2006.03.001.

[49] Fotios Petropoulos and Ivan Svetunkov. A simple combination of univariate models. *International Journal of Forecasting*, 36(1):110–115, 2020. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2019.01.006. URL https://www.sciencedirect.com/science/article/pii/S0169207019300585. M4 Competition.

[50] R. E. Barlow and H. D. Brunk. The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67(337):140–147, 1972. doi: 10.1080/01621459.1972.10481216. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1972.10481216.

5 Appendix

Code snippet demonstrating the `TimeCopilot` Forecaster API, which benchmarks and deploys statistical, machine-learning, neural, and foundational models through a single unified interface.

```
from timecopilot import TimeCopilotForecaster
from timecopilot.models.foundation.toto import Toto
from timecopilot.models.ml import AutoLGBM
from timecopilot.models.neural import AutoNHITS
from timecopilot.models.prophet import Prophet
from timecopilot.models.stats import AutoARIMA, SeasonalNaive


df = pd.read_csv(
    "https://timecopilot.s3.amazonaws.com/public/data/air_passengers.csv",
    parse_dates=["ds"],
)

tcf = TimeCopilotForecaster(
    models=[
        AutoARIMA(),
        AutoLGBM(),
        AutoNHITS(),
        Prophet(),
        SeasonalNaive(),
        Toto(),
    ]
)

fcst_df = tcf.forecast(df=df, h=12)
cv_df = tcf.cross_validation(df=df, h=12)
```