Article

Not peer-reviewed version

CastClaw: A Human-Machine Collaborative System for Agentic Time Series Forecasting

Xiaoyu Tao , Mingyue Cheng * , Tian Gao , Ze Guo , Bokai Pan , Qi Liu , Shijin Wang

Posted Date: 10 June 2026

doi: 10.20944/preprints202606.0713.v1

Keywords: time series forecasting; interactive runtime system

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 June 2026 doi:10.20944/preprints202606.0713.v1

Article

CastClaw: A Human-Machine Collaborative System for Agentic Time Series Forecasting

Xiaoyu Tao, Mingyue Cheng *, Tian Gao, Ze Guo, Bokai Pan, Qi Liu and Shijin Wang

State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, Hefei, China
* Correspondence: mycheng@ustc.edu.cn

Abstract

Time series forecasting is commonly formulated as a model-centric and single-pass prediction task. However, real-world forecasting often requires task understanding, data diagnosis, contextual feature acquisition, tool-assisted modeling, reflective verification, and human feedback. In this paper, we demonstrate CastClaw, an interactive agent system for context-aware time series forecasting. CastClaw organizes forecasting as a structured runtime workflow that includes intent understanding, data profiling, iterative prediction, reflective verification, and traceable report generation. Supported by a tool library, an execution environment, and state management, CastClaw can invoke forecasting tools, compare models across different families and forecasting skills, track intermediate states, and incorporate user feedback. Through demonstrations on real-world forecasting scenarios, CastClaw shows how forecasting systems can move beyond static predictors toward interactive, evidence-grounded, and verifiable forecasting. Our code is available (https://github.com/ustc-time-series/CastClaw). The demonstration video could be found at the link (https://ustc-time-series.github.io/cast-claw/).

Keywords: time series forecasting; interactive runtime system

---

1. Introduction

Time series forecasting (TSF) is a fundamental task in data mining, with broad applications in energy management, healthcare and industrial operations [1]. The goal is to estimate future values from historical observations, thereby supporting decision-making.

Existing forecasting studies have achieved progress, ranging from statistical modeling [2] and deep neural networks [3,4] to recent foundation models [5,6]. These methods provide solutions for modeling temporal dependencies and improving prediction accuracy across forecasting scenarios. Despite their effectiveness, conventional forecasting methods mainly focus on historical numerical patterns, while the contextual features involved in real-world forecasting are often simplified or underutilized. In practice, forecasting decisions may depend not only on historical observations but also on calendar effects, weather conditions, external events, domain knowledge, and human feedback. For example, electricity load forecasting requires jointly considering historical load patterns, holidays, weather changes, and abnormal demand variations.

Recent advances in large language models (LLMs) provide new opportunities for context-aware forecasting. In contrast to conventional forecasting models that primarily rely on numerical observations, LLM-driven methods provide a natural mechanism for aligning and integrating time series with contextual features [7]. Their reasoning ability further supports task decomposition, evidence analysis, and decision revision through explicit reasoning traces [8]. Beyond context modeling, autonomous agents further extend LLMs from passive predictors to interactive task executors. By invoking external tools [9,10], retrieving information [11], and interacting with users [12], LLM agents make it possible to automate parts of the forecasting process. However, existing LLM-driven forecasting agents still mainly focus on tool invocation or model execution, while the human expert forecasting workflow

 doi:10.20944/preprints202606.0713.v1

remains insufficiently modeled [13]. Real-world forecasting rarely depends on a single model call. Human experts often inspect data patterns, retrieve evidence, compare models, and revise decisions iteratively. Yet, such interactive and evidence-driven procedures remain weakly organized in existing forecasting systems.

To address these limitations, we present **CastClaw**, an interactive agent system for context-aware time series forecasting. CastClaw reformulates forecasting from a single-pass model call into a structured runtime workflow. Given a forecasting task, CastClaw first performs intent understanding to clarify task settings, data conditions, modeling preferences, and evaluation requirements. It then conducts qualitative and quantitative data profiling to identify domain knowledge, contextual features, temporal patterns, and data characteristics. Based on the profiled evidence, CastClaw enters an iterative prediction stage, where it invokes forecasting tools, compares candidate models across different families, and updates the forecasting state according to validation performance. A reflective verification stage further examines model effectiveness, failure cases, and refinement directions before generating a traceable forecast report. CastClaw is supported by three runtime-level foundations. A tool library provides unified access to data analysis, skill generation, model execution, evaluation, reflection, file operations, and contextual search. An execution environment supports reproducible experiments through project-level workspaces, script execution, databases, servers, and user interfaces. A state management mechanism records task files, forecasting states, generated skills, reports, results, and user feedback. Together, these components enable CastClaw to maintain a controllable and inspectable forecasting process rather than producing predictions through isolated model calls.

The main contributions of this paper are summarized as follows:

- We present CastClaw, an interactive agent system that reformulates time series forecasting as a structured runtime workflow rather than a single-pass model prediction task.
- We design a context-aware forecasting process that integrates intent understanding, data profiling, iterative prediction, reflective verification, and traceable report generation.
- We develop runtime support for tool invocation, execution management, state tracking, skill reuse, and human feedback, enabling controllable forecasting execution.
- We demonstrate CastClaw on real-world forecasting scenarios, showing its ability to support evidence-grounded, verifiable, and interactive agentic forecasting.

2. Related Work

Existing forecasting methods mainly include statistical models, deep learning models, and time series foundation models. Statistical methods such as ARIMA [14] provide lightweight and interpretable solutions, while deep learning models such as PatchTST [15] improve the modeling of complex temporal dependencies [16–18]. Recent foundation models such as Chronos [19] further enhance zero-shot forecasting [20,21]. However, these methods remain largely model-centric and mainly rely on historical numerical observations or structured covariates. LLM-driven forecasting methods further introduce language models to incorporate contextual features and reasoning capabilities. Representative directions include tokenization-based, prompt-based [22], instruction-guided, alignment-based [23], and reasoning-enhanced methods [24,25]. For example, LLMTime [26] formulates forecasting as next-token prediction, Time-LLM [27] adapts frozen LLMs through prompt-based reprogramming, UniTime [28] uses domain instructions for cross-domain forecasting, TokenCast [29] bridges time series and language through symbolic temporal representations, and MemCast [30] introduces memory-driven reasoning. These studies show the potential of LLMs for context-aware forecasting, but they mostly focus on representation alignment or prediction reasoning, leaving the broader forecasting workflow underexplored. Recent agentic forecasting systems extend LLMs from passive predictors to interactive task executors through tool use, memory, reflection, and human feedback [31,32]. For example, Time-Copilot [33] supports tool-augmented forecasting, FLAIRR-TS [34] introduces retrieval and iterative refinement, Cast-R1 [35] models forecasting as sequential decision-making with state tracking, and

AlphaCast [36] explores human–LLM co-reasoning. Nevertheless, existing systems often treat these capabilities as separate components rather than organizing them into a unified runtime workflow.

3. Method

3.1. System Overview

CastClaw is an interactive runtime system for context-aware agentic time series forecasting. As shown in Figure 1, CastClaw consists of an upper forecasting workflow and a lower runtime support layer. The forecasting workflow includes task understanding, data profiling, skill-guided prediction, reflective verification, and final feedback. The runtime support layer provides the execution environment, tool library, and state management mechanism required to execute and trace the whole forecasting process.

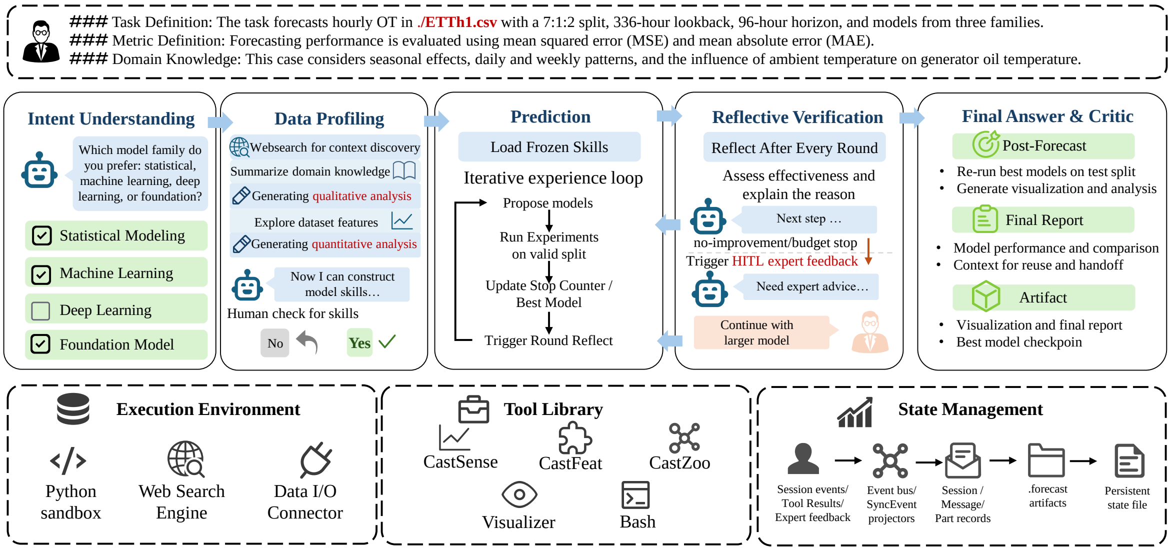

Figure 1. CastClaw workflow overview. The system turns a user forecasting task, historical time series, and contextual features into a clarified, profiled, skill-guided, and self-verified prediction workflow, supported by an execution environment, forecasting tool library, and workflow-driven state management.

3.2. Task Understanding and Data Profiling

Given a forecasting task, CastClaw first converts the user request into a structured task specification, including the target variable, dataset, data split, lookback length, prediction horizon, evaluation metrics, and candidate model families. During intent understanding, users can further specify practical constraints such as accuracy, efficiency, interpretability, and deployment requirements. CastClaw then conducts data profiling from both qualitative and quantitative perspectives. Qualitative profiling summarizes contextual features such as domain knowledge, seasonal effects, periodic patterns, and external factors that may influence the target variable. Quantitative profiling analyzes numerical characteristics of the target time series, including trend, stationarity, seasonality, volatility, missing values, and outliers. These results provide evidence for model-family selection and are stored as structured reports for later reasoning. Based on the profiling results, CastClaw constructs reusable forecasting skills that record task-specific modeling knowledge, model-selection principles, evaluation criteria, and execution constraints. These skills are reviewed and frozen before prediction, ensuring a stable and traceable strategy for subsequent experiments.

3.3. Skill-Guided Forecasting Execution

The prediction stage uses frozen forecasting skills to guide an iterative experiment loop. Instead of directly applying a fixed model, CastClaw dynamically explores candidate models according to the current task state and data conditions. In each round, the system reads the forecasting state, proposes a candidate experiment, generates an executable model specification, runs the experiment, evaluates

 doi:10.20944/preprints202606.0713.v1

the result using predefined metrics, and updates the best-performing configuration. Model-family selection is guided by both data profiling and accumulated experiment results. For relatively stationary and seasonal series, CastClaw may prioritize statistical models as efficient baselines. For non-stationary series with long-range dependencies, it may explore deep learning models. When data are limited or transfer knowledge is useful, foundation models are included as candidate solutions. Meanwhile, CastClaw maintains both per-family and global best results, enabling fair comparison among different forecasting paradigms.

3.4. Reflective Verification and Feedback

After each forecasting round, CastClaw performs reflective verification to assess whether the current result is effective and whether the search should continue. The reflection process checks recent performance, error patterns, failed executions, improvement trends, and budget constraints. It then makes a structured decision, such as keeping the current result, discarding an invalid run, rerunning an experiment, changing the search direction, or stopping the workflow. Before finalizing the prediction, CastClaw conducts post-forecast critic validation on the selected model. This step verifies the best-performing configuration on the held-out test split, compares model-family results, and generates trajectory-level visualization for inspection. When abnormal patterns or ambiguous results appear, the system can request human feedback. Such feedback is recorded as part of the workflow state and can be used to refine future skills or guide subsequent forecasting tasks.

3.5. Runtime Infrastructure

CastClaw is supported by three runtime components. The execution environment provides an isolated project workspace, python runtime, fixed execution scripts, local database, server, and user interface. This environment ensures that forecasting experiments are executed as reproducible runtime actions rather than textual suggestions. The tool library defines the executable actions available to the forecasting agent, including task initialization, dataset analysis, skill generation, state updating, model generation, experiment evaluation, reflection, shell execution, file operation, and web-based context acquisition. These tools connect high-level reasoning with concrete forecasting operations. The state management mechanism records the complete forecasting process, including task definitions, frozen skills, intermediate reports, experiment runs, historical records, best-result summaries, contextual documents, and user feedback. Through state synchronization, CastClaw supports long-running workflow control, failure recovery, skill reuse, and traceable report generation. Together, these components make CastClaw a transparent, controllable, and extensible forecasting system.

4. Demonstration

The demonstration presents CastClaw as an interactive runtime system for end-to-end context-aware time series forecasting. As shown in Figure 2, users can specify a forecasting task in natural language, provide historical time series and contextual features, inspect and revise the generated plan, and follow the complete workflow from data profiling to prediction and reflective verification. The demonstration covers ambiguity handling for underspecified forecasting requests, context discovery and domain knowledge summarization, skill-guided model execution across different model families, and iterative refinement based on validation performance and user feedback. It also shows how CastClaw maintains workflow states, reuses forecasting skills, and generates traceable final reports, enabling users to understand, compare, and control the forecasting process.

Figure 2. CastClaw system interface for forecasting task configuration, report preview, and workspace management during interactive forecasting execution.

5. Experiments

5.1. Experimental Settings

We evaluate CastClaw on eight widely used time series forecasting benchmarks, including ETTh1, ETTh2, ETTm1, ETTm2, Electricity, Traffic, Weather, and ILI, covering energy, transportation, meteorology, and public health scenarios with diverse sampling frequencies and temporal patterns. Following common forecasting protocols, we set the look-back window to 336 and the prediction horizon to 96 for most datasets, while using 168 and 24 for ILI due to its shorter sequence length and lower sampling frequency. We compare CastClaw with representative methods from three paradigms: statistical models, including ARIMA [14], Prophet [37], and Theta [38]; deep forecasting models, including PatchTST [15], ConvTimeNet [39], iTransformer [40], DLinear [41], and Autoformer [42]; and time series foundation models, including Chronos [19], TimesFM [5], and Sundial [6]. Forecasting performance is evaluated using mean squared error (MSE) and mean absolute error (MAE), where lower values indicate better performance.

5.2. Main Results

Table 1 reports the forecasting performance on eight benchmark datasets. Overall, CastClaw achieves the best or highly competitive results on most datasets, demonstrating strong generalization across diverse forecasting scenarios, across heterogeneous datasets with different scales, seasonalities, noise characteristics, and varying levels of complexity. Compared with statistical methods, deep forecasting models, and recent time series foundation models, CastClaw obtains lower or comparable MSE and MAE in most cases. These results suggest the potential benefit of structured workflow control, as CastClaw coordinates data analysis, model training, code generation, evaluation, and iterative revision within an agentic process. By adapting modeling strategies to observed data patterns and refining intermediate results based on execution feedback, CastClaw can better handle heterogeneous distributions and complex temporal patterns. The overall performance gain further indicates the potential utility of workflow-level coordination rather than dataset-specific tuning alone.

 doi:10.20944/preprints202606.0713.v1

Table 1. Forecasting performance comparison across datasets. Lower MSE and MAE indicate better performance. The best results are highlighted in **bold**, and the second-best results are <u>underlined</u>.

Model	ETTh1		ETTh2		ETTm1		ETTm2		Electricity		Traffic		Weather		ILI	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ARIMA	0.1735	0.3143	1.1227	0.8746	0.0837	0.2158	0.3756	0.4753	1.1938	0.9006	3.6773	1.5435	0.0010	0.0225	1.4019	0.9258
Prophet	0.3328	0.4148	0.5435	0.5683	0.0985	0.2295	0.2827	0.4070	1.0204	0.8267	1.8974	1.1709	0.0030	0.0405	1.2690	0.9445
Theta	0.1844	0.3262	0.8911	0.7843	0.0868	0.2200	0.5412	0.5823	1.5817	1.0454	2.1449	1.2264	0.0011	0.0230	1.3542	0.9006
PatchTST	0.1171	0.2684	0.3002	0.4302	0.0504	0.1881	0.1334	<u>0.2598</u>	0.4008	0.4498	0.1688	0.2878	0.0010	<u>0.0222</u>	1.2380	0.9230
ConvTimeNet	0.1255	0.2746	0.2907	0.4223	0.0486	0.1637	0.1317	0.2650	0.2373	0.3559	0.1681	0.2853	0.0060	0.0561	1.6382	1.1408
iTransformer	0.1225	0.2725	0.2863	0.4106	0.0502	0.1653	0.1312	0.2609	0.2570	0.3690	0.1438	0.2280	0.0013	0.0276	1.2291	**0.8099**
DLinear	0.1255	0.2735	<u>0.2467</u>	0.3836	0.0483	<u>0.1621</u>	<u>0.1305</u>	0.2615	0.2006	0.3154	0.1336	**0.2039**	0.0069	0.0645	1.3181	1.0338
Autoformer	0.1262	0.2817	0.2867	0.4314	0.1047	0.2538	0.2433	0.3657	0.4725	0.5103	0.2831	0.3838	**0.0005**	0.0453	2.1923	1.3642
Chronos	0.1233	<u>0.2576</u>	0.2568	**0.3731**	0.0543	0.1853	0.1529	0.2870	0.2006	0.3043	0.1441	0.2166	0.0011	0.0225	1.5307	1.0192
TimesFM	<u>0.1158</u>	0.2586	0.2608	0.3875	0.0654	0.1885	0.1764	0.2853	<u>0.1925</u>	<u>0.3029</u>	0.1440	0.2194	0.0010	0.0224	<u>1.0063</u>	<u>0.8157</u>
Sundial	0.1184	**0.2535**	0.2709	<u>0.3830</u>	0.0521	0.1833	0.1535	0.2885	0.1943	0.3093	<u>0.1335</u>	0.2130	0.0014	0.0270	1.2120	0.8454
CastClaw	**0.1137**	0.2596	**0.2465**	0.3885	**0.0469**	**0.1605**	**0.1290**	**0.2572**	**0.1846**	**0.2965**	**0.1296**	<u>0.2090</u>	<u>0.0009</u>	**0.0210**	**0.9380**	0.8440

6. Conclusion

We presented CastClaw, an interactive agent system for context-aware time series forecasting. CastClaw moves beyond the conventional model-centric and single-pass forecasting paradigm by organizing forecasting as a structured runtime workflow, including intent understanding, data profiling, iterative prediction, reflective verification, and traceable report generation. Supported by a tool library, an execution environment, and state management, CastClaw enables forecasting tools to be invoked, models and forecasting skills to be compared, intermediate states to be tracked, and user feedback to be incorporated during runtime. Through real-world demonstrations, CastClaw shows how forecasting systems can become more interactive, evidence-grounded, and verifiable, providing a practical step toward controllable forecasting.

GenAI Usage Disclosure: Generative AI tools were used for language polishing and grammar refinement. All technical content, system design, experiments, and conclusions were developed and verified by the authors. CastClaw itself uses large language models as part of its agentic forecasting workflow for task understanding, tool selection, reflective verification, and report generation, which is an explicit component of the demonstrated system.

References

1. Cheng, M.; Liu, Z.; Tao, X.; Liu, Q.; Zhang, J.; Pan, T.; Zhang, S.; He, P.; Zhang, X.; Wang, D.; et al. A comprehensive survey of time series forecasting: Concepts, challenges, and future directions. *Authorea Preprints* **2025**.
2. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: principles and practice*; OTexts, 2018.
3. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature* **2015**, *521*, 436–444.
4. Lim, B.; Zohren, S. Time-series forecasting with deep learning: a survey. *Philosophical transactions of the royal society a: mathematical, physical and engineering sciences* **2021**, *379*, 20200209. https://doi.org/10.1098/rsta.2020.0209.
5. Das, A.; Kong, W.; Sen, R.; Zhou, Y. A Decoder-Only Foundation Model for Time-Series Forecasting. In Proceedings of the International Conference on Machine Learning, 2024.
6. Liu, Y.; Qin, G.; Shi, Z.; Chen, Z.; Yang, C.; Huang, X.; Wang, J.; Long, M. Sundial: A Family of Highly Capable Time Series Foundation Models. *arXiv preprint arXiv:2502.00816* **2025**.
7. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In Proceedings of the Advances in Neural Information Processing Systems, 2020, Vol. 33, pp. 1877–1901.
8. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems, 2022, Vol. 35, pp. 24824–24837.
9. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. In Proceedings of the International Conference on Learning Representations, 2023.

10. Schick, T.; Dwivedi-Yu, J.; Dessi, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language Models Can Teach Themselves to Use Tools. In Proceedings of the Advances in Neural Information Processing Systems, 2023, Vol. 36, pp. 68539–68551.
11. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Kuttler, H.; Lewis, M.; Yih, W.t.; Rocktaschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Proceedings of the Advances in Neural Information Processing Systems, 2020, Vol. 33, pp. 9459–9474.
12. Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. A Survey on Large Language Model based Autonomous Agents. *Frontiers of Computer Science* **2024**, *18*, 186345. https://doi.org/10.1007/s11704-024-40231-1.
13. Cheng, M.; Tao, X.; Liu, Q.; Guo, Z.; Chen, E. Position: Beyond Model-Centric Prediction–Agentic Time Series Forecasting. *arXiv preprint arXiv:2602.01776* **2026**.
14. Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time series analysis: forecasting and control*; John Wiley & Sons, 2015.
15. Nie, Y.; Nguyen, N.H.; Sinthong, P.; Kalagnanam, J. A Time Series is Worth 64 Words: Long-Term Forecasting with Transformers. In Proceedings of the International Conference on Learning Representations, 2023.
16. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2021, Vol. 35, pp. 11106–11115.
17. Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; Long, M. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In Proceedings of the International Conference on Learning Representations, 2023.
18. Wang, Y.; Wu, H.; Dong, J.; Qin, G.; Zhang, H.; Liu, Y.; Qiu, Y.; Wang, J.; Long, M. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems* **2024**, *37*, 469–498.
19. Ansari, A.F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S.S.; Pineda-Arango, S.; Kapoor, S.; et al. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research* **2024**.
20. Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; Sahoo, D. Unified Training of Universal Time Series Forecasting Transformers. *arXiv preprint arXiv:2402.02592* **2024**.
21. Garza, A.; Challu, C.; Mergenthaler-Canseco, M. TimeGPT-1. *arXiv preprint arXiv:2310.03589* **2023**.
22. Cao, D.; Jia, F.; Arik, S.O.; Pfister, T.; Zheng, Y.; Ye, W.; Liu, Y. TEMPO: Prompt-Based Generative Pre-Trained Transformer for Time Series Forecasting. In Proceedings of the International Conference on Learning Representations, 2024.
23. Chang, C.; Wang, W.Y.; Peng, W.C.; Chen, T.F. LLM4TS: Aligning Pre-Trained LLMs as Data-Efficient Time-Series Forecasters. *ACM Transactions on Intelligent Systems and Technology* **2025**. https://doi.org/10.1145/3719207.
24. Cheng, M.; Wang, J.; Wang, D.; Tao, X.; Liu, Q.; Chen, E. Can slow-thinking llms reason over time? empirical studies in time series forecasting. In Proceedings of the Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining, 2026, pp. 99–110.
25. Luo, Y.; Zhou, Y.; Cheng, M.; Wang, J.; Wang, D.; Pan, T.; Zhang, J. Time series forecasting as reasoning: A slow-thinking approach with reinforced llms. *arXiv preprint arXiv:2506.10630* **2025**.
26. Gruver, N.; Finzi, M.; Qiu, S.; Wilson, A.G. Large Language Models Are Zero-Shot Time Series Forecasters. In Proceedings of the Advances in Neural Information Processing Systems, 2023, Vol. 36.
27. Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J.Y.; Shi, X.; Chen, P.Y.; Liang, Y.; Li, Y.F.; Pan, S.; et al. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In Proceedings of the International Conference on Learning Representations, 2024.
28. Liu, X.; Hu, J.; Li, Y.; Diao, S.; Liang, Y.; Hooi, B.; Zimmermann, R. UniTime: A Language-Empowered Unified Model for Cross-Domain Time Series Forecasting. In Proceedings of the Proceedings of the ACM Web Conference, 2024.
29. Tao, X.; Zhang, S.; Cheng, M.; Wang, D.; Pan, T.; Pan, B.; Zhang, C.; Wang, S. From values to tokens: An llm-driven framework for context-aware time series forecasting via symbolic discretization. *arXiv preprint arXiv:2508.09191* **2025**.
30. Tao, X.; Cheng, M.; Guo, Z.; Yu, S.; Liu, Y.; Liu, Q.; Wang, S. MemCast: Memory-Driven Time Series Forecasting with Experience-Conditioned Reasoning. *arXiv preprint arXiv:2602.03164* **2026**.
31. Yeh, C.C.M.; Lai, V.; Saini, U.S.; Fan, X.; Fan, Y.; Wang, J.; Dai, X.; Zheng, Y. Empowering Time Series Forecasting with LLM-Agents. *arXiv preprint arXiv:2508.04231* **2025**.

 doi:10.20944/preprints202606.0713.v1

32. Pan, B.; Cheng, M.; Liu, Z.; Yu, S.; Tao, X.; Wu, Y.; Liu, Q.; Lian, D.; Chen, E. CastFlow: Learning Role-Specialized Agentic Workflows for Time Series Forecasting. *arXiv preprint arXiv:2604.27840* **2026**.
33. Garza, A.; Rosillo, R. TimeCopilot. *arXiv preprint arXiv:2509.00616* **2025**.
34. Jalori, G.; Verma, P.; Arik, S.O. FLAIRR-TS: Forecasting LLM-Agents with Iterative Refinement and Retrieval for Time Series. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP, 2025.
35. Tao, X.; Cheng, M.; Jiang, C.; Gao, T.; Zhang, H.; Liu, Y. Cast-R1: Learning tool-augmented sequential decision policies for time series forecasting. *arXiv preprint arXiv:2602.13802* **2026**.
36. Zhang, X.; Gao, T.; Cheng, M.; Pan, B.; Guo, Z.; Liu, Y.; Tao, X.; Liu, Q. Alphacast: A human wisdom-llm intelligence co-reasoning framework for interactive time series forecasting. *arXiv preprint arXiv:2511.08947* **2025**.
37. Taylor, S.J.; Letham, B. Forecasting at scale. *The American Statistician* **2018**, *72*, 37–45.
38. Assimakopoulos, V.; Nikolopoulos, K. The theta model: a decomposition approach to forecasting. *International journal of forecasting* **2000**, *16*, 521–530.
39. Cheng, M.; Yang, J.; Pan, T.; Liu, Q.; Li, Z.; Wang, S. Convtimenet: A deep hierarchical fully convolutional model for multivariate time series analysis. In Proceedings of the Companion Proceedings of the ACM on Web Conference 2025, 2025, pp. 171–180.
40. Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; Long, M. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In Proceedings of the International Conference on Learning Representations, 2024.
41. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are Transformers Effective for Time Series Forecasting? In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2023, Vol. 37, pp. 11121–11128.
42. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In Proceedings of the Advances in Neural Information Processing Systems, 2021, Vol. 34, pp. 22419–22430.