# LLM Agents for Time-Series: A Survey

**Yilong Chen**[1] **Xiao Qin**[1] **Chenghao Liu**[2*]
**Liang Wu**[3] **Noelle I. Samia**[1] **Kaize Ding**[1†]
[1]Northwestern University [2]Datadog AI Research [3]Nokia
[†]Corresponding author



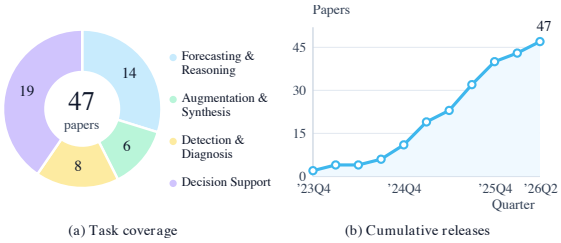
## Abstract

LLM-based agents are increasingly being developed for time-series problems, but their design choices vary substantially across task settings. This survey adopts a problem-driven taxonomy that organizes these systems by the time-series problems they address rather than by isolated technical components. We group existing systems into four categories: forecasting and reasoning, augmentation and synthesis, anomaly detection and diagnosis, and decision support. Within each category, we examine how task requirements shape agent architecture, tool use, and memory design. We further summarize representative datasets and environments, and compare reported model performance under shared or closely related settings. Overall, this survey offers a task-oriented guide to designing LLM-based agents for time-series problems and identifies open gaps for future work.

## 1 Introduction

Time-series analysis (Hamilton, 2020) plays an important role in many real-world domains, including finance (Tsay, 2005), transportation (Xu et al., 2016), and climate science (Kim et al., 2025). Representative tasks include forecasting (Chatfield, 2000; De Gooijer and Hyndman, 2006), data augmentation (Wen et al., 2020), anomaly detection (Blázquez-García et al., 2021; Schmidl et al., 2022), and decision support. These tasks have long been addressed with statistical and classical machine learning models such as ARIMA (Shumway and Stoffer, 2017), bootstrapping (Efron, 1992), and LOF (Breunig et al., 2000), and more recently with deep models such as RNNs (Medsker and Jain, 2000) and Transformers (Vaswani et al., 2017).

However, applying these methods often relies heavily on expert knowledge to design analysis pipelines and interpret results. Recent advances in large language models (LLMs) (Zhao et al., 2026) offer an alternative by leveraging general-purpose language reasoning to assist time-series analysis, for example as modules for encoding and explaining temporal patterns. Moving beyond prompt-only use, LLMs can be deployed as *agents* (Wang et al., 2024a) that plan actions, call tools, and maintain memory across multiple steps. Such agentic systems are particularly suitable for time-series applications because many time-series tasks require updates from new observations, decisions under evolving conditions, and tool-augmented integration of heterogeneous evidence rather than a fixed pipeline (Cheng et al., 2026; Tao et al., 2026; Zhang et al., 2025c). Motivated by this shift, this survey reviews recent progress on LLM-based agentic systems for time-series tasks.




(a) Task coverage (b) Cumulative releases

Figure 1: Task coverage and release timeline of 47 surveyed LLM-agent papers for time-series tasks.

Recent time-series leaderboards further suggest that agentic systems are becoming competitive with state-of-the-art methods by harnessing strong time-series models rather than replacing them (Aksu et al., 2024). By coordinating foundation models, retrieval, validation, and domain tools within task-specific workflows, these systems shift the practical question from which model to use toward how agent architectures, tools, and memory should be designed for each task. This motivates our problem-driven taxonomy of time-series agentic systems.

**Positioning.** Existing LLM-for-time-series surveys (Chang et al., 2026; Zhang et al., 2025c) typically organize methods by individual LLM capabil-

[*]This work was completed prior to joining Datadog.

ities (e.g., planning, reasoning, memory, and tool use), rather than by how these capabilities are integrated for concrete time-series tasks. Conversely, domain-general LLM-agent surveys (Wang et al., 2024a; Guo et al., 2024; Li et al., 2024; Ferrag et al., 2025) seldom address time-series-specific challenges such as streaming inputs, temporal dependence, distribution shift, and action–data coupling. We therefore adopt a *problem-driven taxonomy* that groups methods by target task and analyzes recurring design patterns in architecture, tool use, and memory. A detailed comparison with related surveys is provided in Appendix A.

The remainder of this survey introduces the necessary background and core design dimensions in Sections 2–3, presents the problem-driven taxonomy in Section 4, and reviews practical resources and future research directions in Sections 5–6.

## 2 Background and Foundations

**Time-Series Data and Representations.** A time series may arise from a continuous-time process, but practical systems usually operate on observed or tokenized indices. Let $\{\tau_t\}_{t=1}^{T}$ satisfy $\tau_1 < \cdots < \tau_T$, and denote

$$\mathbf{X}_{\tau_1:\tau_T} = (\mathbf{x}_{\tau_1}, \ldots, \mathbf{x}_{\tau_T}), \quad \mathbf{x}_{\tau_t} \in \mathbb{R}^d,$$

where $d$ covers observed channels, covariates, or spatially distributed sensors. For irregular, noisy, or partially observed data, systems typically construct $\mathbf{z}_{\tau_t} = \phi(\mathbf{X}_{\tau_1:\tau_t})$ to summarize historical context.

**Time-Series Modeling.** Time-series modeling transforms temporal observations into forecasts, anomaly scores, learned representations, or explanatory summaries. Statistical models (e.g., ARIMA, ARCH/GARCH, HMMs) encode assumptions about dependence, stationarity, latent regimes, and uncertainty (Shumway and Stoffer, 2017; Engle, 1982; Bollerslev, 1986; Rabiner, 1989). Deep models (e.g., LSTM, TCN, DeepAR) learn temporal patterns (Hochreiter and Schmidhuber, 1997; Bai et al., 2018; Salinas et al., 2020), and LLM-based models (e.g., Time-LLM, LSTPrompt) represent sequences through tokenized or multimodal inputs for language-based reasoning or explanation (Jin et al., 2024; Liu et al., 2024a). As standalone components, however, these models typically do not choose workflows, call tools, or revise memory and hypotheses based on intermediate feedback.

**From Inference to Agentic Systems.** Static analysis pipelines are often inadequate for interactive time-series tasks that require evidence gathering, feedback integration, or adaptive action selection. We view LLM agents as observe–act–update systems that combine reasoning, planning, tool use, and memory (Wang et al., 2024a; Zhang et al., 2025c), and use this distinction to separate agentic systems from prompt-only LLM applications.

## 3 Fundamental Design Dimensions

Before the task taxonomy, we summarize three design dimensions of time-series agentic systems: architecture, tools, and memory.

### 3.1 Architecture

We distinguish single-agent and multi-agent architectures; for multi-agent systems, we further describe their architectural patterns as cooperative, competitive, or mixed (Zhu et al., 2024).

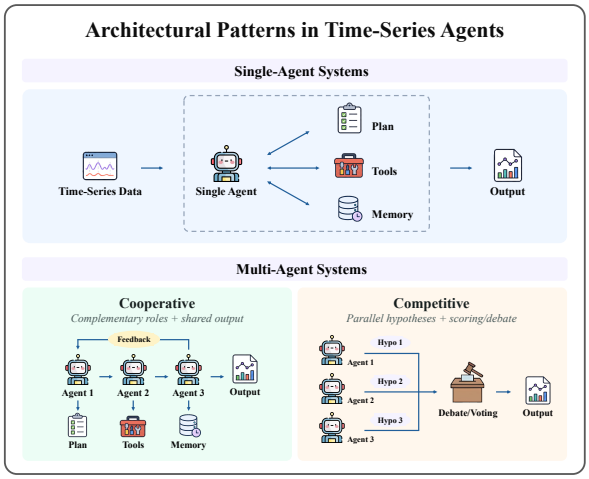



Figure 2: An illustration of the main architectural patterns for time-series agent systems.

**Single-Agent Systems.** A single-agent system uses one agent to plan reasoning steps, call tools, maintain memory, and produce outputs or actions from current observations (Yao et al., 2023).

**Cooperative Multi-Agent Systems.** Cooperative systems assign agents complementary roles or subtasks that are not directly interchangeable, such as sequential stages or planner–executor structures, so agents coordinate toward a shared output (Li et al., 2024; Wooldridge, 2009).

**Competitive Multi-Agent Systems.** Competitive systems instantiate agents or agent-generated candidates as alternatives: they produce competing hypotheses, forecasts, explanations, rules, or actions, which are then resolved by scoring, ranking, voting, debate, or an explicit judge (Zhu et al., 2024).

**Mixed Multi-Agent Systems.** Mixed systems combine cooperative role decomposition with competitive comparison, debate, or selection within the same workflow (Zhu et al., 2024).

### 3.2 Tools

Tools are callable interfaces to external programs invoked by the LLM agent (Wang et al., 2024c). In time-series agents, they enable access to temporal data, specialized computation, and verifiable feedback that cannot be obtained reliably from parametric knowledge alone. Common tool families include *(i) Database APIs*, which provide structured access to stored data; *(ii) Search & Retrieval APIs*, which return relevant external information or historical records; *(iii) Data Processing Tools*, which transform raw inputs into more useful representations, such as through alignment or dynamic time warping (DTW); *(iv) Statistical & ML Models*, which produce predictions, scores, or learned representations; and *(v) Simulators, Solvers & Optimizers*, which evaluate actions, enforce constraints, or compute solutions in structured environments.

### 3.3 Memory

Memory in time-series agents helps maintain coherence across reasoning steps, especially in temporally evolving settings (Zhang et al., 2024d). We summarize memory by functional role: *(i) Evidence logs*, which record intermediate traces of a decision process, such as intermediate predictions, retrieved evidence, tool outputs, candidate actions, and validation results; *(ii) Pattern library*, which stores retrievable historical cases or typical data fragments, such as recurring market patterns, fault signatures, or similar past situations; and *(iii) Analysis strategies*, which capture reusable experience about how to analyze a situation, such as which tools to use, which features to focus on, or how to interpret signals before acting.

## 4 Taxonomy

We define an *LLM agent for time series* as a system in which an LLM must make at least one decision that changes a multi-step time-series workflow, such as selecting an action or tool, updating memory, revising a hypothesis, or coordinating stages. We classify the system as single-agent when one LLM fills this role and as multi-agent when two or more LLMs take distinct decision-making roles and interact through cooperation, competition, or both. We exclude (i) one-shot prompting, (ii) pipelines in which the LLM serves only as a static encoder or post-hoc explainer, and (iii) domain-general agents that are not designed for time-series constraints or temporally grounded evaluation.

We adopt a problem-driven taxonomy that groups systems by the time-series problems they address, and discuss design dimensions within each setting. This choice is user-oriented: readers are often more concerned with what kinds of designs are suitable for a given time-series problem than with design dimensions in isolation. A problem-driven view therefore provides clearer guidance for selecting and understanding agent designs in practice. Under this taxonomy, we identify four problem categories: *Forecasting & Reasoning*, *Augmentation & Synthesis*, *Anomaly Detection & Diagnosis*, and *Decision Support*. Within each category, we further distinguish several sub-problems. Table 3 summarizes representative methods; Figure 3 shows the full taxonomy.

### 4.1 Time-Series Forecasting & Reasoning

Time-series forecasting and reasoning share a common requirement: the agent must produce outputs that are not only plausible in language, but also grounded in explicit numerical or contextual evidence. In both settings, a fixed context window or a coarse global summary is often insufficient, because the correct conclusion may depend on local temporal patterns, historical analogs, reusable prototypes, or aligned external context such as news (Zhang et al., 2025e). As a result, effective systems usually do not treat context as a static input. Instead, they actively construct evidence beyond the context window, for example by retrieving relevant slices on demand (Jalori et al., 2025) or querying prototype memories (Jiang et al., 2025).

**Time-Series Forecasting.** Time-series forecasting agents aim to predict future values from historical observations under non-stationarity, noise, and horizon-dependent uncertainty. In forecasting, agent systems are mainly shaped by: (i) multi-stage workflows, and (ii) competing hypotheses.

*Multi-stage workflows* are central in forecasting because prediction is rarely a single-step task. Such workflows may involve data diagnosis, preprocessing, model selection, contextual analysis, prediction, and validation, so errors in early stages can invalidate later conclusions. A common design is therefore to ground each stage in explicit intermediate evidence rather than free-form reasoning alone. TimeSeriesScientist (Zhao et al., 2025) is

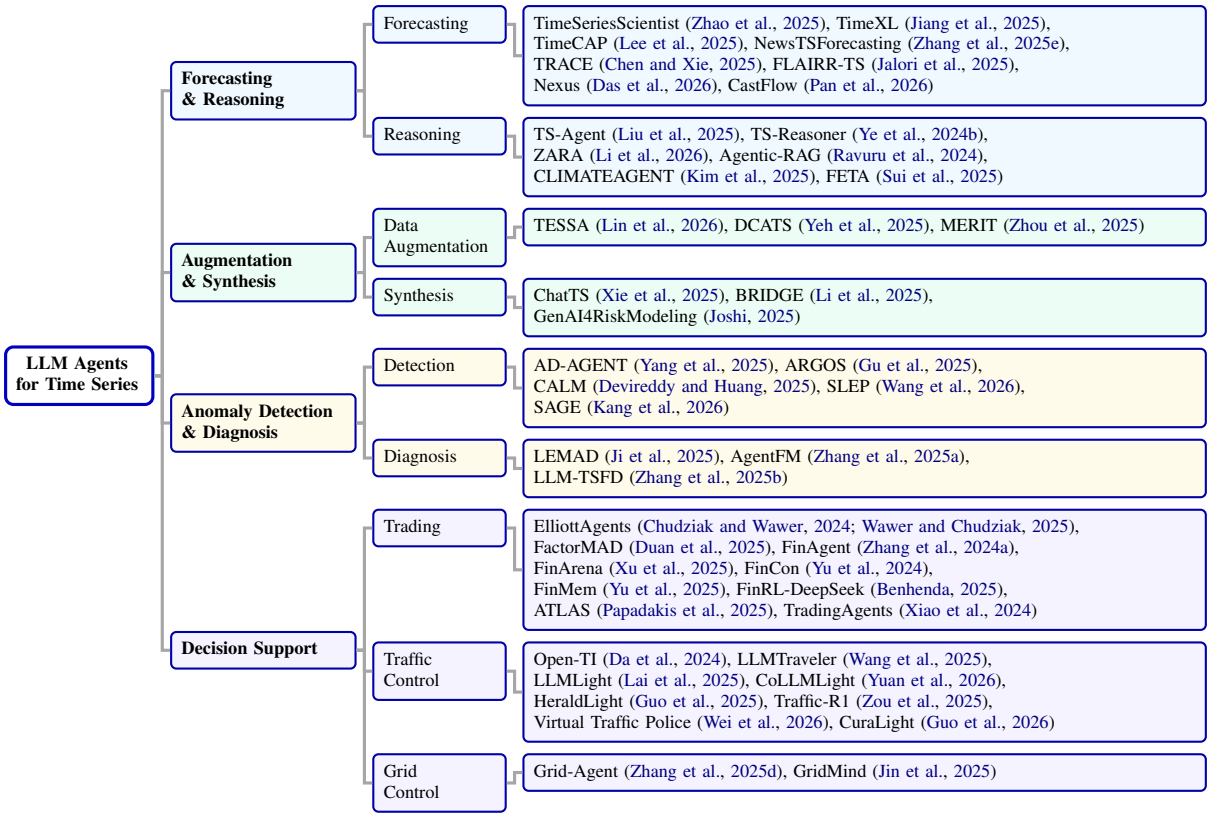



Figure 3: A taxonomy of LLM agents for time-series tasks.

a representative example: it uses statistical models and other tools to generate diagnostics, validation results, and configuration records, and stores them as evidence logs for review, provenance, and correction. Nexus (Das et al., 2026) uses contextualization, dual-resolution macro/micro outlook generation, and synthesis/calibration stages, while CastFlow (Pan et al., 2026) organizes forecasting as planning, action, prediction, and reflection with memory and multi-view diagnostic tools.

*Competing hypotheses* are another distinctive feature of forecasting. Different strategies may focus on different signals, temporal scales, or exogenous factors, so relying on a single reasoning path can be brittle. One design is to use competitive architectural patterns, where agents or candidate strategies represent forecasting views and are compared through explicit error feedback. NewsTS-Forecasting (Zhang et al., 2025e) follows this idea by using error-based scoring and reflection to control strategy updates. TRACE (Chen and Xie, 2025) similarly uses communication and multi-agent consistency refinement under sparse or missing observations, while also showing that consistency alone is not enough unless communication is checked against evidence.

**Time-Series Reasoning.** Time-series reasoning agents derive explanations, answers, or classifications from temporal evidence rather than directly predicting future values. In reasoning, two task-specific considerations are especially important: (i) domain knowledge and numerical support, and (ii) multi-step reasoning reliability.

*Domain knowledge and numerical support* are essential in reasoning because text-trained LLMs do not reliably encode temporal structure, domain mechanisms, or quantitative operations. A common design is therefore to separate planning from computation: the main agent handles coordination, while numerical analysis is delegated to specialized sub-agents or auditable tools and operators. Agentic-RAG (Ravuru et al., 2024) illustrates the sub-agent route, while TS-Agent (Liu et al., 2025) illustrates the tool-grounded route. Reasoning systems may also require domain knowledge beyond raw observations. ZARA (Li et al., 2026) mines discriminative features offline and stores domain feature-importance profiles as reusable guidance, while CLIMATEAGENT (Kim et al., 2025) uses specialized data agents to handle API conventions, metadata retrieval, and parameter validation. For classification-style reasoning, FETA (Sui et al.,

2025) decomposes multivariate series into channel-wise comparisons against retrieved exemplars and then aggregates confidence-weighted decisions.

*Multi-step reasoning reliability* is similar to multi-stage forecasting workflows: reasoning tasks also involve a sequence of dependent steps, and early errors can propagate across the chain. A common design is therefore to make the process explicit, so intermediate results can be checked and revised rather than passed forward implicitly. CLIMATEAGENT (Kim et al., 2025) follows the sub-agent route and records code, retrieved data, and results as evidence logs, so later stages can build on verified outputs. TS-Reasoner (Ye et al., 2024b) follows the operator route by compiling reasoning into an executable operator pipeline, where execution feedback can trigger plan revision and operator reselection. In both cases, evidence logs support downstream reasoning, reflection, and recovery. Verification may be handled by dedicated checking agents or by the same LLM in a critic role, as in TS-Agent (Liu et al., 2025).

**Discussion:** Forecasting and reasoning both need evidence beyond a fixed context window and are vulnerable to error accumulation in long workflows. These shared challenges motivate pattern libraries for retrieving precedents, evidence logs for preserving intermediate steps, and tool calls for quantitative support. The main difference is that forecasting more often compares competing hypotheses, whereas reasoning more often uses refinement loops to reduce long-chain errors.

### 4.2 Time-Series Augmentation & Synthesis

Time-series data augmentation and synthesis construct additional data while preserving time-series semantics. Their main failure mode is semantic drift: the constructed data may look plausible but violate the structures that matter for downstream tasks, such as trend, periodicity, local shapes, or correlations. The key difference is semantic source: augmentation mainly relies on the target series, whereas synthesis must align control text (e.g., scenario descriptions) with domain rules.

**Time-Series Data Augmentation.** Time-series data augmentation aims to construct additional data around a target sequence while preserving task-relevant semantics. It typically appears in two forms: generating numeric perturbations and generating textual annotations as an alternative representation of the same series. These two forms are shaped by different challenges: *(i) semantic preservation* is central for numeric augmentation, while *(ii) domain annotation understanding* is the main bottleneck for textual augmentation.

*Semantic preservation* is the core challenge in numeric augmentation. Augmented data should remain aligned with the target series rather than merely look realistic in isolation. A common design follows one of two routes: either construct target-specific training sets from neighboring series, as in DCATS (Yeh et al., 2025), or retrieve similar sequences and then apply classic augmentations such as jittering, scaling, and time warping, as in MERIT (Zhou et al., 2025). Verification is especially important here. LLM-as-Judge based only on pretrained knowledge can be brittle, so stronger designs usually rely on downstream validation signals (Yeh et al., 2025).

*Domain annotation understanding* is the main challenge in textual augmentation. In series-to-text semantic augmentation, the added data is not a new numeric sequence but a textual annotation, which can be viewed as another representation or semantic view of the same series. The difficulty is that pretrained LLMs do not reliably understand domain-specific time-series annotations. TESSA (Lin et al., 2026) derives domain-agnostic concepts (e.g., trend, periodicity, volatility) from cross-domain annotations and converts them into domain-specific annotations via a domain agent.

**Time-Series Synthesis.** Time-series synthesis aims to generate new sequences under specified controls or constraints rather than to expand a single target series. Under this view, it mainly includes two settings: text-guided synthesis, where the system generates sequences that satisfy both scenario descriptions and domain knowledge, and domain-attribute construction without external scenario text. These two settings face different bottlenecks: *(i) text–series alignment* is central in the former, while *(ii) domain constraint compliance* is the main challenge in the latter.

*Text–series alignment* is the core challenge in text-guided synthesis. In real-world settings, paired scenario-query and time-series data are usually scarce, so LLMs cannot reliably map free-form text directly to numerical sequences. A common design is therefore to rewrite descriptions of real time series into structured text queries and then map these queries into executable generation settings. BRIDGE (Li et al., 2025) extracts templates

(e.g., length, trend, periodicity, extrema, variance), refines them, and trains a controlled diffusion generator. GenAI4RiskModeling (Joshi, 2025) instead uses LLM-generated queries to tune GAN/VAE-based interest-rate scenario generation.

*Domain constraint compliance* is the main challenge in domain-attribute construction without external scenario text. In this setting, the difficulty is not text alignment but ensuring that generated data still conforms to implicit domain rules and constraints. ChatTS (Xie et al., 2025) constructs synthetic time-series/text supervision by sampling domain-relevant attributes, generating rule-consistent series, and filtering Q&A pairs for attribute consistency.

**Discussion:** For augmentation and synthesis, the central issue is semantic drift: generated outputs may look plausible while violating task-relevant structures. Current systems therefore favor construction and verification pipelines, using data processing, generation modules, and downstream validation signals. Memory is less central here; when used, it mainly stores reusable templates or semantic patterns with domain rules.

### 4.3 Time-Series Anomaly Detection & Diagnosis

Time-series anomaly detection and diagnosis are closely related: detection asks whether and when abnormal patterns occur, while diagnosis asks why and where they occur and how to respond. Their main difference lies in task emphasis.

**Time-Series Anomaly Detection.** In our scope, time-series anomaly detection focuses on producing reliable alarms, such as point-wise labels, anomalous windows, or alert events. In practice, three challenges are especially important: *(i) evidence quality*, *(ii) non-degradation* relative to the base detector, and *(iii) streaming monitoring* under continual updates.

*Evidence quality* matters because reliable alarms often require more than raw series values alone. Here, decision evidence refers to the information directly supporting the alarm decision, such as summary statistics, learned features, or retrieved historical snippets. A common design is to strengthen this evidence by injecting it into prompts or downstream modules. SLEP (Wang et al., 2026) follows this direction by enriching detector inputs with additional evidence rather than relying only on the raw sequence. SAGE (Kang et al., 2026) makes this evidence construction more explicit by assigning specialized analyzers to point, structural, seasonal, and pattern anomalies, then consolidating their tool-grounded outputs into confidence-scored anomaly records.

*Non-degradation* is critical because the agent is often layered on top of an existing deployed detector, so it should improve alarm quality without underperforming the base system. A common design is to treat the deployed detector as a base model and let the agent learn complementary corrections for its typical errors. ARGOS (Gu et al., 2025) exemplifies this idea by constructing deterministic rules to correct base-detector failures and fusing rule outputs with detector outputs at inference time.

*Streaming monitoring* is a special but practically important setting because concept drift may require continual adaptation. The main risk is contamination control: short-lived anomalies may be absorbed as the new normal during online updates. CALM (Devireddy and Huang, 2025) addresses this with a continual design in which an LLM judge filters training data by distinguishing transient noise from sustained distribution shift, and only the latter is used to fine-tune the forecasting-based detector.

**Time-Series Diagnosis.** In our scope, time-series diagnosis focuses on explaining, localizing, and responding to abnormal events. In practice, two challenges are especially important: *(i) evidence integration*, and *(ii) limited diagnostic supervision*.

*Evidence integration* is important for diagnosis for reasons similar to anomaly detection, but diagnosis places more emphasis on explanation and localization. In the papers we survey, this is often handled by explicitly incorporating textual evidence, such as logs, alerts, traces, and topology context, as part of the model input or generated report. AgentFM (Zhang et al., 2025a) follows this pattern and further improves stability through a RAG+CoT design that retrieves labeled historical examples as task-specific references, reducing free-form drift.

*Limited diagnostic supervision* is another recurring bottleneck because labeled fault cases and high-quality incident narratives are often scarce. LLM-TSFD (Zhang et al., 2025b) addresses this with a human-in-the-loop data preparation stage, where users specify labeling or cleaning intent and the system generates executable code refined through feedback.

**Discussion:** Anomaly detection and diagnosis both build actionable monitoring pipelines from heterogeneous temporal evidence. This makes tool support central, especially data processing, detectors, analytical modules, and external evidence sources. Sequential cooperative pipelines remain common, while streaming scenarios emphasize memory updates and deployment-time adaptation. Memory mainly supports auditability and retrieval through evidence logs and historical pattern libraries.

### 4.4 Time-Series Decision Support

Time-series decision support differs from forecasting or diagnosis because the output is an executable action, plan, or allocation rather than a descriptive judgment. Decisions must be grounded in a predefined action space and satisfy domain constraints. Existing agentic work that satisfies our time-series scope is concentrated in trading, traffic control, and grid control; because the latter two share closed-loop control constraints, we discuss them together. We exclude decision-support agents in domains such as healthcare treatment and supply-chain operations when they do not directly use time-series evidence to produce executable decisions.

**Trading.** Trading agents focus on sequential financial decisions (e.g., buy, sell, hold) under evolving market conditions. Three considerations are central: (i) heterogeneous external evidence, (ii) historical experience, and (iii) risk preference.

*Heterogeneous external evidence*, such as news, financial statements, social media, earnings calls, visual charts, and technical indicators, often affects trading decisions, but it comes in different forms and operates at different temporal scales. Processing all of it with a single agent can easily lead to context overload and mixed signals. A common cooperative pattern is a planner–executor structure, where specialized subagents or modules handle different evidence sources and their outputs are then aggregated. TradingAgents (Xiao et al., 2024), FinCon (Yu et al., 2024), and FinArena (Xu et al., 2025) all follow this pattern.

Trading decisions often rely heavily on *historical experience*. In agent systems, this is usually supported by memory in the form of *pattern library* and *analytical strategies*, which abstract past situations into retrievable patterns or reusable decision experience. FinAgent (Zhang et al., 2024a) exemplifies this design: each stage produces a retrieval-oriented query, allowing market situations, price-driving explanations, and trading lessons to be stored separately. FinCon (Yu et al., 2024) further updates *manager-level investment beliefs*, which serve as evolving analytical strategies.

*Risk preference* is another special concern in trading. In some settings, the system needs to account for human preference alignment; FinArena (Xu et al., 2025) injects user risk preferences and feedback into prompts so that they directly influence the final recommendation. In other settings, the system does not explicitly incorporate user feedback, but instead constructs internal role-based variation to induce different risk styles, as in TradingAgents (Xiao et al., 2024) and FinMem (Yu et al., 2025). Recent evaluation work further emphasizes that static financial QA is insufficient for evaluating trading agents: StockBench (Chen et al., 2025) evaluates agents in multi-month markets where daily prices, fundamentals, and news lead to sequential buy–sell–hold decisions.

**Traffic & Grid Control.** Traffic and grid control agents both choose executable control actions from evolving infrastructure states. Traffic systems focus on signal actions under real-time flow constraints and network-level coordination, while grid systems target mitigation plans under changing loads, violations, contingencies, and uncertainty.

*Hard constraints* are central because infrastructure actions must be evaluated in closed-loop environments. In traffic control, queue length, waiting time, throughput, and travel time evolve after each signal action. LLMLight (Lai et al., 2025) operates in a fixed control space and is evaluated in a traffic simulator. More recent systems add stronger coordination and validation mechanisms: CoLLMLight (Yuan et al., 2026) constructs a spatiotemporal graph for network-wide coordination, HeraldLight (Guo et al., 2025) uses a dual-LLM design with herald-guided prompts for fine-grained signal control, and Traffic-R1 (Zou et al., 2025) trains a lightweight reasoning model for real-time signal control. In grid control, candidate actions such as switching, curtailment, or dispatch must be checked by power-flow solvers or safety validators. Grid-Agent (Zhang et al., 2025d) is a representative example: the LLM proposes structured mitigation plans, while power-flow solvers and validation modules determine their performance. GridMind (Jin et al., 2025) similarly uses LLM agents for power-system analysis, with solvers providing domain-grounded feedback.

*Human-specified policies* are another distinctive feature. In some systems, humans specify policies or optimization settings, and the LLM agent mainly orchestrates downstream execution. Open-TI (Da et al., 2024) illustrates this pattern well: some tasks translate human-described policies into signal actions, while others let the user specify simulation settings or optimization techniques and have the LLM route the request to the appropriate tool chain. Virtual Traffic Police (Wei et al., 2026) follows a related augmentation strategy by using an LLM agent to adjust parameters of existing traffic controllers under unforeseen incidents, while CuraLight (Guo et al., 2026) uses debate-guided data curation to improve an LLM-centered signal controller.

**Discussion:** Different decision targets lead to different design principles across trading, traffic control, and grid control. Trading systems more often use cooperative planner–executor designs to decompose multimodal evidence, whereas traffic and grid control systems rely more on simulator-grounded execution–validation pipelines. The same contrast appears in memory and tool use: trading emphasizes historical experience and heterogeneous data, while traffic and grid control depend more on simulation and feasibility checks.

## 5 Resources

This section summarizes key resources for implementing and evaluating LLM agents for time-series tasks. We organize them into four categories and provide representative examples (Table 5).

**Datasets and Repositories.** These resources provide the raw observations used for training and offline evaluation. Representative forecasting datasets include Electricity (Lai et al., 2018), METR-LA (Li et al., 2017), and ETT (Zhou et al., 2021), while common anomaly datasets include SWaT (Mathur and Tippenhauer, 2016) and SMAP/MSL (Hundman et al., 2018). Monash TSF (Godahewa et al., 2021) serves as a multi-domain repository.

**Benchmarks.** Benchmarks define comparable tasks and metrics across methods. Forecasting benchmarks include M4/M5 (Makridakis et al., 2018, 2022) and MIRAI (Ye et al., 2024a), while anomaly suites include NAB (Lavin and Ahmad, 2015) and TSB-UAD (Paparrizos et al., 2022). TimeSeriesExam (Cai et al., 2024) extends evaluation toward reasoning.

**Interactive Environments.** These platforms enable closed-loop experiments with explicit state, action, and reward signals. Typical examples include Grid2Op (Marot et al., 2021) for power systems, SUMO (Behrisch et al., 2011) for traffic control, and SocioDojo (Cheng and Chin, 2024) for trading.

**Toolkits.** Toolkits provide reusable pipelines, APIs, and baselines for reproducible development. Common options include StatsForecast (Garza et al., 2022), NeuralForecast (Challu et al., 2023), Sktime (Löning et al., 2019), and Stable-Baselines3 (Raffin et al., 2021).

## 6 Future Work

This survey reviews LLM-based agentic systems for time-series problems through a problem-driven taxonomy of forecasting and reasoning, augmentation and synthesis, anomaly detection and diagnosis, and decision support. Building on this taxonomy, we highlight four open directions.

**Numerical Understanding and Domain Knowledge.** LLMs remain limited in understanding numerical signals and specialized domain knowledge (Hung et al., 2023; Ye et al., 2024b). Even with external tools, agents must still interpret tool outputs and connect them to domain-specific reasoning.

**Causal and Counterfactual Temporal Reasoning.** Many time-series agents still rely mainly on correlations, while causal discovery from time series remains assumption-sensitive (Assaad et al., 2022). Diagnosis and decision support require evidence-grounded reasoning about interventions, delayed effects, and counterfactual outcomes; otherwise agents may hallucinate causal or temporal claims (Wang et al., 2023).

**Online Adaptation and Continual Improvement.** In anomaly detection and decision support, agents often need to improve after deployment. Yet most systems remain static or rely on memory updates, while parameter adaptation remains rare (Jaglan and Barnes, 2025; Zheng et al., 2026).

**Benchmarking Agent Workflows.** Current studies evaluate time-series agents mainly with task metrics, leaving workflow quality undermeasured. Future benchmarks should evaluate intermediate decisions, tool use, validation, and reflection alongside downstream performance (Weng et al., 2026; Cheng et al., 2026).

## Limitations

While this survey provides a comprehensive overview of LLM-based agentic systems for time-series tasks, it has several limitations:

**Scope and coverage.** Due to the rapid pace of advancements in LLM-based agents, some recent developments and emerging directions may not be fully captured in this survey.

**Lack of quantitative comparison.** The broad range of time-series tasks and heterogeneous evaluation settings make it difficult to establish a unified and fair empirical comparison across all systems.

**Design guidance.** The design insights are derived from recurring patterns observed in the literature rather than controlled experimental validation, and thus may not generalize to all practical settings.

Despite these limitations, we hope this survey provides a useful and structured reference for understanding and designing LLM-based time-series agents.

## References


Sridhar Adepu, Venkata Reddy Palleti, Gyanendra Mishra, and Aditya Mathur. 2020. Investigation of cyber attacks on a water distribution system. In *International Conference on Applied Cryptography and Network Security*, pages 274–291. Springer.

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

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang. 2020. Gluonts: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116):1–6.

Charles K. Assaad, Emilie Devijver, and Eric Gaussier. 2022. Survey and evaluation of causal discovery methods for time series. *Journal of Artificial Intelligence Research*, 73:767–819.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271.

Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. 2011. Sumo–simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, the third international conference on advances in system simulation*. ThinkMind.

Mostapha Benhenda. 2025. FinRL-DeepSeek: LLM-infused risk-sensitive reinforcement learning for trading agents. *arXiv preprint arXiv:2502.07393*.

Aadyot Bhatnagar, Paul Kassianik, Chenghao Liu, Tian Lan, Wenzhuo Yang, Rowan Cassius, Doyen Sahoo, Devansh Arpit, Sri Subramanian, Gerald Woo, Amrita Saha, Arun Kumar Jagota, Gokulakrishnan Gopalakrishnan, Manpreet Singh, K C Krithika, Sukumar Maddineni, Daeki Cho, Bo Zong, Yingbo Zhou, and 4 others. 2021. Merlion: A machine learning library for time series. *arXiv preprint arXiv:2109.09265*.

Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. 2021. A review on outlier/anomaly detection in time series data. *ACM computing surveys (CSUR)*, 54(3):1–33.

Tim Bollerslev. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104.

Yifu Cai, Arjun Choudhry, Mononito Goswami, and Artur Dubrawski. 2024. Timeseriesexam: A time series understanding exam. *arXiv preprint arXiv:2410.14752*.

Yifu Cai, Xinyu Li, Mononito Goswami, Michał Wiliński, Gus Welter, and Artur Dubrawski. 2025. Timeseriesgym: A scalable benchmark for (time series) machine learning engineering agents. *arXiv preprint arXiv:2505.13291*.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. 2023. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 6989–6997.

Ching Chang, Yidan Shi, Defu Cao, Wei Yang, Jeehyun Hwang, Haixin Wang, Jiacheng Pang, Wei Wang, Yan Liu, Wen-Chih Peng, and Tien-Fu Chen. 2026. A survey of reasoning and agentic systems in time series with large language models. *Transactions on Machine Learning Research*.

Chris Chatfield. 2000. *Time-series forecasting*. Chapman and Hall/CRC.

Si-An Chen, Chun-Liang Li, Sercan O. Arik, Nathanael Christian Yoder, and Tomas Pfister. 2023. Tsmixer: An all-mlp architecture for time series forecasting. *Transactions on Machine Learning Research*.

Yanxu Chen, Zijun Yao, Yantao Liu, Amy Xin, Jin Ye, Jianing Yu, Lei Hou, and Juanzi Li. 2025. Stockbench: Can llm agents trade stocks profitably in real-world markets? *arXiv preprint arXiv:2510.02209*.

Yuxuan Chen and Haipeng Xie. 2025. Trace: Unlocking the potential of llms in time series forecasting for distributed energy resources. *IEEE Transactions on Artificial Intelligence*.

Junyan Cheng and Peter Chin. 2024. Sociodojo: Building lifelong analytical agents with real-world text and time series. In *The Twelfth International Conference on Learning Representations*.

Mingyue Cheng, Xiaoyu Tao, Qi Liu, Ze Guo, and Enhong Chen. 2026. Position: Beyond model-centric prediction – agentic time series forecasting. *arXiv preprint arXiv:2602.01776*.

Jaroslaw A Chudziak and Michal Wawer. 2024. Elliottagents: A natural language-driven multi-agent system for stock market analysis and prediction. In *Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation*, pages 961–970.

Longchao Da, Kuanru Liou, Tiejin Chen, Xuesong Zhou, Xiangyong Luo, Yezhou Yang, and Hua Wei. 2024. Open-TI: Open traffic intelligence with augmented language model. *International Journal of Machine Learning and Cybernetics*, 15(10):4761–4786.

Sarkar Snigdha Sarathi Das, Palash Goyal, Mihir Parmar, Nanyun Peng, Vishy Tirumalashetty, Chun-Liang Li, Rui Zhang, Jinsung Yoon, and Tomas Pfister. 2026. Nexus: An agentic framework for time series forecasting. *arXiv preprint arXiv:2605.14389*.

Jan G De Gooijer and Rob J Hyndman. 2006. 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473.

Ashok Devireddy and Shunping Huang. 2025. Calm: A framework for continuous, adaptive, and llm-mediated anomaly detection in time-series streams. *arXiv preprint arXiv:2508.21273*.

Yitong Duan, Chuheng Zhang, and Jian Li. 2025. FactorMAD: A multi-agent debate framework based on large language models for interpretable stock alpha factor mining. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pages 605–613.

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

Robert F Engle. 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1008.

Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. 2025. From llm reasoning to autonomous ai agents: A comprehensive review. *arXiv preprint arXiv:2504.19678*.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*.

Federico Garza, Max Mergenthaler Canseco, Cristian Challú, and Kin G Olivares. 2022. Statsforecast: Lightning fast forecasting with statistical and econometric models. *PyCon Salt Lake City, Utah, US*, 2022:6.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

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

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*, volume 36, pages 24013–24034.

Yile Gu, Yifan Xiong, Jonathan Mace, Yuting Jiang, Yigong Hu, Baris Kasikci, and Peng Cheng. 2025. ARGOS: Agentic time-series anomaly detection with autonomous rule generation via large language models. *arXiv preprint arXiv:2501.14170*.

Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Thomas Paine, Sergio Gómez, Konrad Zolna, Rishabh Agarwal, Josh S Merel, Daniel J Mankowitz, Cosmin Paduraru, Gabriel Dulac-Arnold, Jerry Li, Mohammad Norouzi, Matthew Hoffman, Nicolas Heess, and Nando de Freitas. 2020. Rl unplugged: A suite of benchmarks for offline reinforcement learning. *Advances in neural information processing systems*, 33:7248–7259.

Qing Guo, Xinhang Li, Junyu Chen, Zheng Guo, Xiaocong Li, Lin Zhang, and Lei Li. 2025. A dual large language models architecture with herald guided prompts for parallel fine grained traffic signal control. *arXiv preprint arXiv:2511.00136*.

Qing Guo, Xinhang Li, Junyu Chen, Zheng Guo, Shengzhe Xu, Lin Zhang, and Lei Li. 2026. CuraLight: Debate-guided data curation for LLM-centered traffic signal control. *arXiv preprint arXiv:2604.05663*.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

James D Hamilton. 2020. *Time series analysis*. Princeton university press.

Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. 2022. Adbench: Anomaly detection benchmark. *Advances in neural information processing systems*, 35:32142–32159.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 387–395.

Chia-Chien Hung, Wiem Ben Rim, Lindsay Frost, Lars Bruckner, and Carolin Lawrence. 2023. Walking a tightrope–evaluating large language models in high-risk domains. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pages 99–111.

Aman Jaglan and Jarrod Barnes. 2025. Continual learning, not training: Online adaptation for agents. *arXiv preprint arXiv:2511.01093*.

Gunjan Jalori, Preetika Verma, and Sercan Ö Arık. 2025. FLAIRR-TS–forecasting LLM-agents with iterative refinement and retrieval for time series. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15427–15437. Association for Computational Linguistics.

Sijie Ji, Xinzhe Zheng, and Chenshu Wu. 2024. Hargpt: Are llms zero-shot human activity recognizers? *arXiv preprint arXiv:2403.02727*.

Xin Ji, Le Zhang, Wenya Zhang, Fang Peng, Yifan Mao, Xingchuang Liao, and Kui Zhang. 2025. LEMAD: LLM-empowered multi-agent system for anomaly detection in power grid services. *Electronics*, 14(15):3008.

Yushan Jiang, Zijie Pan, Xikun Zhang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. 2024. Empowering time series analysis with large language models: A survey. *arXiv preprint arXiv:2402.03182*.

Yushan Jiang, Wenchao Yu, Geon Lee, Dongjin Song, Kijung Shin, Wei Cheng, Yanchi Liu, and Haifeng Chen. 2025. TimeXL: Explainable multi-modal time series prediction with LLM-in-the-loop. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Hongwei Jin, Kibaek Kim, and Jonghwan Kwon. 2025. GridMind: LLMs-powered agents for power system analysis and operations. In *Proceedings of the SC'25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 560–568.

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. 2024. Time-llm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*.

Satyadhar Joshi. 2025. Using gen ai agents with gae and vae to enhance resilience of us markets. *Available at SSRN 5123068*.

Hyeongwon Kang, Jeongseob Kim, Jinwoo Park, and Pilsung Kang. 2026. Detecting time series anomalies like an expert: A multi-agent llm framework with specialized analyzers. *arXiv preprint arXiv:2605.05725*.

Hyeonjae Kim, Chenyue Li, Wen Deng, Mengxi Jin, Wen Huang, Mengqian Lu, and Binhang Yuan. 2025. Climateagent: Multi-agent orchestration for complex climate data science workflows. *arXiv preprint arXiv:2511.20109*.

Kenneth R Knapp, Michael C Kruk, David H Levinson, Howard J Diamond, and Charles J Neumann. 2010. The international best track archive for climate stewardship (ibtracs) unifying tropical cyclone data. *Bulletin of the American Meteorological Society*, 91(3):363–376.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104.

Siqi Lai, Zhao Xu, Weijia Zhang, Hao Liu, and Hui Xiong. 2025. LLMLight: Large language models as traffic signal control agents. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2335–2346.

Nikolay Laptev. 2015. Yahoo s5 dataset. Yahoo Webscope Dataset S5. Time-series anomaly detection benchmark.

Alexander Lavin and Subutai Ahmad. 2015. Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pages 38–44. IEEE.

Geon Lee, Wenchao Yu, Kijung Shin, Wei Cheng, and Haifeng Chen. 2025. TimeCAP: Learning to contextualize, augment, and predict time series events with large language model agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18082–18090.

Zikang Leng, Hyeokhyen Kwon, and Thomas Ploetz. 2023. Generating virtual on-body accelerometer data from virtual textual descriptions for human activity recognition. In *Proceedings of the 2023 ACM International Symposium on Wearable Computers*.

Hao Li, Yu-Hao Huang, Chang Xu, Viktor Schlegel, Renhe Jiang, Riza Batista-Navarro, Goran Nenadic, and Jiang Bian. 2025. BRIDGE: Bootstrapping text to control time-series generation via multi-agent iterative optimization and diffusion modeling. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 34742–34773. PMLR.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. 2024. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

Zechen Li, Baiyu Chen, Hao Xue, and Flora D Salim. 2026. ZARA: Training-free motion time-series reasoning via evidence-grounded LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14986–15008. Association for Computational Linguistics.

Minhua Lin, Zhengzhang Chen, Yanchi Liu, Xujiang Zhao, Zongyu Wu, Junxiang Wang, Xiang Zhang, Suhang Wang, and Haifeng Chen. 2026. Decoding time series with llms: A multi-agent framework for cross-domain annotation. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 6244–6281.

Haoxin Liu, Zhiyuan Zhao, Jindong Wang, Harshavardhan Kamarthi, and B. Aditya Prakash. 2024a. Lst-prompt: Large language models as zero-shot time series forecasters by long-short-term prompting. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7832–7840, Bangkok, Thailand. Association for Computational Linguistics.

Jiexi Liu and Songcan Chen. 2024. Timesurl: Self-supervised contrastive learning for universal time series representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13918–13926.

Jun Liu, Chaoyun Zhang, Jiaxu Qian, Minghua Ma, Si Qin, Chetan Bansal, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2024b. Large language models can deliver accurate and interpretable time series anomaly detection. *arXiv preprint arXiv:2405.15370*.

Penghang Liu, Elizabeth Fons, Annita Vapsi, Mohsen Ghassemi, Svitlana Vyetrenko, Daniel Borrajo, Vamsi K. Potluru, and Manuela Veloso. 2025. TS-Agent: Understanding and reasoning over raw time series via iterative insight gathering. *arXiv preprint arXiv:2510.07432*. NeurIPS 2025 Workshop on Foundations of Reasoning in Language Models.

Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. 2023. Largest: A benchmark dataset for large-scale traffic forecasting. *Advances in Neural Information Processing Systems*, 36:75354–75371.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024c. itransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*.

Markus Löning, Anthony Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J Király. 2019. sktime: A unified interface for machine learning with time series. *arXiv preprint arXiv:1909.07872*.

Yunfei Luo, Yuliang Chen, Asif Salekin, and Tauhidur Rahman. 2024. Toward foundation model for multivariate wearable sensing of physiological signals. *arXiv preprint arXiv:2412.09758*.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2018. The m4 competition: Results, findings, conclusion and way forward. *International Journal of forecasting*, 34(4):802–808.

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

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long short term memory networks for anomaly detection in time series. *ESANN*.

Antoine Marot, Benjamin Donnot, Gabriel Dulac-Arnold, Adrian Kelly, Aidan O'Sullivan, Jan Viebahn, Mariette Awad, Isabelle Guyon, Patrick Panciatici, and Camilo Romero. 2021. Learning to run a power network challenge: a retrospective analysis. In *NeurIPS 2020 competition and demonstration track*, pages 112–132. PMLR.

Aditya P Mathur and Nils Ole Tippenhauer. 2016. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, pages 31–36. IEEE.

Larry R Medsker and Lakhmi C Jain. 2000. *Recurrent Neural Networks: Design and Applications*. CRC Press.

Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Ayush Saraf, Amy Bearman, and Babak Damavandi. 2023. Imu2clip: Language-grounded motion sensor translation with multimodal contrastive learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13246–13253.

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

*International Conference on Learning Representations*.

Bokai Pan, Mingyue Cheng, Zhiding Liu, Shuo Yu, Xiaoyu Tao, Yuchong Wu, Qi Liu, Defu Lian, and Enhong Chen. 2026. Castflow: Learning role-specialized agentic workflows for time series forecasting. *arXiv preprint arXiv:2604.27840*.

Charidimos Papadakis, Angeliki Dimitriou, Giorgos Filandrianos, Maria Lymperaiou, Konstantinos Thomas, and Giorgos Stamou. 2025. ATLAS: Adaptive trading with LLM agents through dynamic prompt optimization and multi-agent coordination. *arXiv preprint arXiv:2510.15949*.

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*.

Lawrence R Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. 2021. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of machine learning research*, 22(268):1–8.

Chidaksh Ravuru, Sagar Srinivas Sakhinana, and Venkataramana Runkana. 2024. Agentic retrieval-augmented generation for time series analysis. *arXiv preprint arXiv:2408.14484*.

Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3009–3017.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191.

Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797.

Robert H Shumway and David S Stoffer. 2017. Arima models. In *Time series analysis and its applications: with R examples*, pages 75–163. Springer.

Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. 2017. Anomaly detection in streams with extreme value theory. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1067–1075.

Songyuan Sui, Zihang Xu, and Xia Hu. 2025. Training-free time series classification via in-context reasoning with llm agents. *arXiv preprint arXiv:2510.05950*.

Xiaoyu Tao, Mingyue Cheng, Chuang Jiang, Tian Gao, Huanjian Zhang, and Yaguo Liu. 2026. Cast-r1: Learning tool-augmented sequential decision policies for time series forecasting. *arXiv preprint arXiv:2602.13802*.

Ruey S Tsay. 2005. *Analysis of financial time series*. John wiley & sons.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Bingrui Wang, Yuan Zhou, Leijiao Ge, and Sun-Yuan Kung. 2026. Large-model-based smart agent for time series anomaly detection in power systems. *Expert Systems with Applications*, 296:128917.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. 2024a. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Leizhen Wang, Peibo Duan, Zhengbing He, Cheng Lyu, Xin Chen, Nan Zheng, Li Yao, and Zhenliang Ma. 2025. Agentic large language models for day-to-day route choices. *Transportation Research Part C: Emerging Technologies*, 180:105307.

Zexin Wang, Changhua Pei, Minghua Ma, Xin Wang, Zhihan Li, Dan Pei, Saravan Rajmohan, Dongmei Zhang, Qingwei Lin, and Haiming Zhang. 2024b. Revisiting vae for unsupervised time series anomaly detection: A frequency perspective. In *Proceedings of the ACM Web Conference*, pages 3096–3105.

Zhendong Wang, Ioanna Miliou, Isak Samsten, and Panagiotis Papapetrou. 2023. Counterfactual explanations for time series forecasting. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 1391–1396.

Zhiruo Wang, Zhoujun Cheng, Hao Zhu, Daniel Fried, and Graham Neubig. 2024c. What are tools anyway? a survey from the language model perspective. *arXiv preprint arXiv:2403.15452*.

Michał Wawer and Jarosław A Chudziak. 2025. Integrating traditional technical analysis with ai: A multi-agent llm-based approach to stock market forecasting. *arXiv preprint arXiv:2506.16813*.

Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash V. Gayah, Kai Xu, and Zhenhui Li. 2019a. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1290–1298.

Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. 2019b. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1913–1922.

Shiqi Wei, Qiqing Wang, and Kaidi Yang. 2026. Virtual traffic police: Large language model-augmented traffic signal control for unforeseen incidents. *arXiv preprint arXiv:2601.15816*.

Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. 2020. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*.

Muyan Weng, Defu Cao, Wei Yang, Yashaswi Sharma, and Yan Liu. 2026. Temporalbench: A benchmark for evaluating llm-based agents on contextual and event-informed time series tasks. *arXiv preprint arXiv:2602.13272*.

Michael Wooldridge. 2009. *An Introduction to MultiAgent Systems*. Wiley, Chichester, UK.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pages 22419–22430.

Renjie Wu and Eamonn J Keogh. 2021. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE transactions on knowledge and data engineering*, 35(3):2421–2429.

Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. 2024. TradingAgents: Multi-agents LLM financial trading framework. *arXiv preprint arXiv:2412.20138*.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. 2025. ChatTS: Aligning time series with LLMs via synthetic data for enhanced understanding and reasoning. *Proceedings of the VLDB Endowment*, 18(8):2385–2398.

Congluo Xu, Zhaobin Liu, and Ziyang Li. 2025. FinArena: A human-agent collaboration framework for financial market analysis and forecasting. *arXiv preprint arXiv:2503.02692*.

Fengli Xu, Yuyun Lin, Jiaxin Huang, Di Wu, Hongzhi Shi, Jeungeun Song, and Yong Li. 2016. Big data driven mobile traffic understanding and forecasting: A time series approach. *IEEE transactions on services computing*, 9(5):796–805.

Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, and Yang Feng. 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pages 187–196.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2021. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*.

Hao Xue and Flora D. Salim. 2023. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*.

Tiankai Yang, Junjun Liu, Michael Siu, Jiahang Wang, Zhuangzhuang Qian, Chanjuan Song, Cheng Cheng, Xiyang Hu, and Yue Zhao. 2025. Ad-agent: A multi-agent framework for end-to-end anomaly detection. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 191–205.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Chenchen Ye, Ziniu Hu, Yihe Deng, Zijie Huang, Mingyu Derek Ma, Yanqiao Zhu, and Wei Wang. 2024a. Mirai: Evaluating llm agents for event forecasting. *arXiv preprint arXiv:2407.01231*.

Wen Ye, Wei Yang, Defu Cao, Yizhou Zhang, Lumingyuan Tang, Jie Cai, and Yan Liu. 2024b. TS-Reasoner: Domain-oriented time series inference agents for reasoning and automated analysis. *arXiv preprint arXiv:2410.04047*.

Chin-Chia Michael Yeh, Vivian Lai, Uday Singh Saini, Xiran Fan, Yujie Fan, Junpeng Wang, Xin Dai, and Yan Zheng. 2025. Empowering time series forecasting with llm-agents. *arXiv preprint arXiv:2508.04231*.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. 2023. Frequency-domain mlps

are more effective learners in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 36, pages 76656–76679.

Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Jordan W Suchow, Denghui Zhang, and Khaldoun Khashanah. 2025. FinMem: A performance-enhanced LLM trading agent with layered memory and character design. *IEEE Transactions on Big Data*.

Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yupeng Cao, Zhi Chen, Jordan W Suchow, Rong Liu, Zhenyu Cui, Zhaozhuo Xu, Denghui Zhang, Koduvayur Subbalakshmi, Guojun Xiong, Yueru He, Jimin Huang, Dong Li, and Qianqian Xie. 2024. FinCon: A synthesized LLM multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems*, 37:137010–137045.

Zirui Yuan, Siqi Lai, and Hao Liu. 2026. CoLLM-Light: Cooperative large language model agents for network-wide traffic signal control. In *International Conference on Learning Representations*.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8980–8987.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11121–11128.

Zhiyuan Zeng, Jiashuo Liu, Siyuan Chen, Tianci He, Yali Liao, Yixiao Tian, Jinpeng Wang, Zaiyuan Wang, Yang Yang, Lingyue Yin, Mingren Yin, Zhenwei Zhu, Tianle Cai, Zehui Chen, Jiecao Chen, Yantao Du, Xiang Gao, Jiacheng Guo, Liang Hu, and 12 others. 2025. Futurex: An advanced live benchmark for llm agents in future prediction. *arXiv preprint arXiv:2508.11987*.

Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. 2022a. Tfad: A decomposition time series anomaly detection architecture with time-frequency analysis. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, pages 2497–2507.

Liang Zhang, Qiang Wu, Jun Shen, Linyuan Lu, Bo Du, and Jianqing Wu. 2022b. Expression might be enough: Representing pressure and demand for reinforcement learning based traffic signal control. In *Proceedings of the 39th International Conference on Machine Learning*, pages 26645–26654.

Lingzhe Zhang, Yunpeng Zhai, Tong Jia, Xiaosong Huang, Chiming Duan, and Ying Li. 2025a. AgentFM: Role-aware failure management for distributed databases with LLM-driven multi-agents. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, pages 525–529.

Qi Zhang, Chao Xu, Jie Li, Yicheng Sun, Jinsong Bao, and Dan Zhang. 2025b. Llm-tsfd: An industrial time series human-in-the-loop fault diagnosis method based on a large language model. *Expert Systems with Applications*, 264:125861.

Regina Zhang, Si Qi Goh, Xingsheng Chen, Zongru Li, Honggang Wen, Jiale Guo, Menglin Yang, Hongzhi Yin, Qiang Yang, Siu-Ming Yiu, and Kwok-Yan Lam. 2025c. From prompts to agents: A comprehensive survey of llm-driven time series analysis. *Available at SSRN 6614598*.

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, Longtao Zheng, Xinrun Wang, and Bo An. 2024a. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In *Proceedings of the 30th acm sigkdd conference on knowledge discovery and data mining*, pages 4314–4325.

Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K Gupta, and Jingbo Shang. 2024b. Large language models for time series: A survey. *arXiv preprint arXiv:2402.01801*.

Xiyuan Zhang, Dongjin Teng, Ranak Roy Chowdhury, Siyang Li, Dong Hong, Rajesh K. Gupta, and Jingbo Shang. 2024c. Unimts: Unified pre-training for motion time series. In *Advances in Neural Information Processing Systems*, volume 37, pages 107469–107493.

Yan Zhang, Ahmad Mohammad Saber, Amr Youssef, and Deepa Kundur. 2025d. Grid-Agent: An LLM-powered multi-agent system for power grid control. *arXiv preprint arXiv:2508.05702*.

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

Yuxuan Zhang, Yangyang Feng, Daifeng Li, Kexin Zhang, Junlan Chen, and Bowen Deng. 2025e. Can competition enhance the proficiency of agents powered by large language models in the realm of news-driven time series forecasting? *arXiv preprint arXiv:2504.10210*.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024d. A survey on the memory mechanism of large language model based agents. *Preprint*, arXiv:2404.13501.

Haokun Zhao, Xiang Zhang, Jiaqi Wei, Yiwei Xu, Yuting He, Siqi Sun, and Chenyu You. 2025. TimeSeriesScientist: A general-purpose AI agent for time series analysis. *arXiv preprint arXiv:2510.01538*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2026. A survey of large language models. *Frontiers of Computer Science*, 20(12):2012627.

Junhao Zheng, Chengming Shi, Xidi Cai, Qiuke Li, Duzhen Zhang, Chenxing Li, Dong Yu, and Qianli Ma. 2026. Lifelong learning of large language model based agents: A roadmap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115.

Shu Zhou, Yunyang Xuan, Yuxuan Ao, Xin Wang, Tao Fan, and Hao Wang. 2025. Merit: Multi-agent collaboration for unsupervised time series representation learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24011–24028.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, pages 27268–27286.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. 2023. One fits all: Power general time series analysis by pretrained lm. In *Advances in Neural Information Processing Systems*, volume 36.

Changxi Zhu, Mehdi Dastani, and Shihan Wang. 2024. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, 38(1):4.

Xingchen Zou, Yuhao Yang, Zheng Chen, Xixuan Hao, Yiqi Chen, Chao Huang, and Yuxuan Liang. 2025. Traffic-R1: Reinforced LLMs bring human-like reasoning to traffic signal control systems. *arXiv preprint arXiv:2508.02344*.

# A Survey Scope and Related Surveys

This appendix describes how we selected papers and how our survey differs from related surveys on LLMs for time-series analysis.

**Paper selection.** We searched arXiv, ACL Anthology, ACM Digital Library, IEEE Xplore, DBLP, Semantic Scholar, Google Scholar, and references from related surveys, with the last update in May 2026. Search terms paired agent concepts, including *LLM agent*, *multi-agent*, *tool use*, *memory*, and *reflection*, with time-series tasks and domains such as forecasting, anomaly detection, diagnosis, trading, traffic control, and power grids. We included papers in which an LLM controls at least one decision that shapes a multi-step time-series workflow, including action or tool selection, memory updating, hypothesis revision, or stage coordination. We excluded one-shot prompting, static LLM encoders, post-hoc explainers, and domain-general agents without time-series evaluation. The final corpus contains 47 representative systems; Table 3 codes their agent design, and Table 6 reports comparable metrics where available.

**Comparison with related surveys.** Table 1 compares survey scope, task coverage, taxonomy, and design guidance. For categorical labels, *Yes* denotes that the named dimension is a primary and sustained focus, *Partial* denotes agent coverage without a clear separation from prompt-only methods, and *Limited* denotes localized rather than category-wide design discussion. Method-, topology-, capability-, and problem-driven taxonomies organize methods by LLM adaptation, reasoning structure, agent capability, and time-series problem family, respectively.

Topology-driven surveys characterize reasoning structures (Chang et al., 2026); our problem-driven taxonomy complements them by linking task settings to architecture, tools, and memory. We cover 47 representative systems under the inclusion rule above and report per-system annotations in Table 3.

# B Representative Prompt Examples by Task Family

This appendix provides representative prompt examples for the major task families in our taxonomy. The examples are abstracted from recurring input–output needs in the surveyed systems. They are not verbatim prompts from any single paper, and they are not intended to define a separate taxonomy of prompt types.

## B.1 Forecasting and Reasoning

```
System:
You are a time-series analysis agent.
Use temporal evidence, retrieved context,
and tool outputs to produce a grounded
forecast, answer, or classification. Do
not rely on unsupported patterns.

User:
Task:
- Forecast the target horizon, or answer
  the temporal reasoning or classification
  question.

Inputs:
- Historical observations:
  <series or summarized windows>
- Target horizon or question:
  <forecast horizon / question>
- Retrieved analogs or external context:
  <optional>
- Candidate hypotheses or exemplars:
  <optional>
- Tool outputs:
  <trend, seasonality, anomaly,
   correlation, model scores>

Instructions:
1. Identify trend, seasonality, local
   changes, and unusual events.
2. Select evidence relevant to the
   horizon or question.
3. Compare candidate hypotheses or
   retrieved exemplars when provided.
4. Use tool outputs for quantitative
   claims.
5. Produce the forecast, answer, or label
   with a concise rationale.
6. State uncertainty and failure modes.

Output JSON:
{
          "evidence_summary": "...",
          "reasoning": "...",
          "prediction_answer_or_label": "...",
          "confidence": "...",
          "limitations": "..."
}
```

This example reflects common input–output needs in forecasting and reasoning agents such as TimeSeriesScientist (Zhao et al., 2025), TimeXL (Jiang et al., 2025), TS-Agent (Liu et al., 2025), and TS-Reasoner (Ye et al., 2024b).

## B.2 Augmentation and Synthesis

```
System:
You are assisting time-series data
construction. Augment a target series or
synthesize a controlled series only when
the requested temporal semantics and
domain constraints can be preserved.

User:
Task:
- Create an augmented example, synthetic
  series, or textual annotation.

Inputs:
- Construction mode:
  <target augmentation | text-guided
```

Table 1: Comparison with related time-series LLM surveys. ✓: clearly covered; ∼: partially covered; ×: not covered.

| Survey | Year | TS-specific | LLM Agents | Forecasting & Reasoning | Augmentation & Synthesis | Anomaly Detection & Diagnosis | Decision Support | Taxonomy | Design Guidance |
|---|---|---|---|---|---|---|---|---|---|
| (Jiang et al., 2024) | 2024 | Yes | No | ✓ | ∼ | ✓ | ∼ | Method-driven | Limited |
| (Zhang et al., 2024b) | 2024 | Yes | No | ✓ | ✓ | ✓ | × | Method-driven | Limited |
| (Chang et al., 2026) | 2026 | Yes | Partial | ✓ | ✓ | ✓ | ✓ | Topology-driven | Limited |
| (Zhang et al., 2025c) | 2025 | Yes | Partial | ✓ | ✓ | ✓ | ∼ | Capability-driven | Limited |
| **Ours** | 2026 | Yes | Yes | ✓ | ✓ | ✓ | ✓ | Problem-driven | Yes |

```
   synthesis | domain-attribute synthesis>
- Seed series or source examples:
  <values / windows / examples>
- Desired label, scenario, or query:
  <class, event, scenario, or description>
- Required temporal properties:
  <trend, periodicity, volatility, extrema>
- Domain constraints:
  <valid ranges, correlations,
   physical rules>
- Validation criteria:
  <downstream check or consistency rule>

Instructions:
1. Preserve task-relevant temporal
   semantics when augmenting a target
   series.
2. Align control text or domain attributes
   with the generated series when
   synthesizing new data.
3. Apply only transformations consistent
   with the label or scenario.
4. Avoid changing causal, seasonal, or
   domain-critical structure.
5. Explain which properties are preserved
   or intentionally changed.
6. Return a compact structured result for
   downstream validation.

Output JSON:
{
  "construction_mode": "...",
          "constructed_item": "...",
          "control_attributes": "...",
          "preserved_properties": "...",
  "changed_properties": "...",
  "validation_notes": "..."
}
```

This example reflects common input–output needs in augmentation and synthesis systems such as TESSA (Lin et al., 2026), MERIT (Zhou et al., 2025), BRIDGE (Li et al., 2025), and ChatTS (Xie et al., 2025).

### B.3 Anomaly Detection and Diagnosis

```
System:
You are a time-series monitoring agent.
Convert detector evidence, temporal
context, and auxiliary logs into a
supported alarm, point/window label, or
diagnostic explanation.

User:
Task:
- Decide whether the candidate window is
  anomalous, or explain a fault.

Inputs:
- Candidate window:
  <time range and observed values>
- Detector evidence:
  <scores, thresholds, labels, residuals>
- Temporal context:
  <recent history, seasonality,
   expected behavior>
- Auxiliary context:
  <logs, alerts, topology,
   related variables>
- Historical cases:
  <optional retrieved examples>

Instructions:
1. Compare the candidate window with
   expected temporal behavior.
2. Separate detector evidence from
   contextual or textual evidence.
3. Check whether context supports or
   contradicts the detector evidence.
4. Identify the most likely abnormal
   interval or root cause.
5. Avoid overriding the detector or
   overclaiming when evidence is
   weak or conflicting.
6. Return a decision, supporting evidence,
   and confidence.

Output JSON:
{
  "decision": "normal | anomalous | uncertain",
  "abnormal_interval": "...",
  "supporting_evidence": "...",
  "root_cause_hypothesis": "...",
  "confidence": "..."
}
```

This example reflects common input–output needs in anomaly detection and diagnosis systems such as SAGE (Kang et al., 2026), ARGOS (Gu et al., 2025), AgentFM (Zhang et al., 2025a), and LLM-TSFD (Zhang et al., 2025b).

### B.4 Decision Support

```
System:
You are a sequential decision-support
agent. Recommend only actions allowed by
the action space and supported by the
current temporal state, constraints, and
validation feedback.

User:
Task:
- Select the next trading, traffic-control,
  or grid-control action.

Inputs:
- Current state:
  <market / intersection / grid state>
- Recent history:
  <prices, flows, loads, events,
   or violations>
- Historical experience:
  <retrieved cases, lessons, or patterns>
- External evidence:
  <news, indicators, forecasts,
   alerts, optional>
```

```
- Action space:
  <allowed actions and parameter ranges>
- Constraints:
  <risk preference, safety limits,
   policies, timing, feasibility>
- Validation feedback:
  <simulator, solver, or previous outcome>

Instructions:
1. Summarize the state variables that
   matter for the next action.
2. Compare feasible actions under the
   provided constraints.
3. Use historical experience for
   market-like settings when available.
4. Use validation feedback to reject
   unsafe or dominated actions.
5. Recommend one action and explain the
   expected effect.
6. State the main risk and what should be
   monitored next.

Output JSON:
{
  "state_summary": "...",
  "recommended_action": "...",
  "justification": "...",
  "constraint_check": "...",
  "next_monitoring_target": "..."
}
```

This example reflects common input–output needs in decision-support systems such as TradingAgents (Xiao et al., 2024), FinCon (Yu et al., 2024), LLMLight (Lai et al., 2025), and GridAgent (Zhang et al., 2025d).

## C Typical Design Patterns

Figure 4 provides a supplementary visual summary of the typical design patterns discussed across different time-series tasks in Section 4. Each panel shows a common design pattern rather than the exact design of a specific system. Dashed boxes denote agent modules together with their tools, solid boxes group entities with similar roles, solid arrows show the main flow, dashed arrows indicate conditional or fallback paths, and background colors group panels from the same higher-level problem type.

## D Failure Modes and Design Patterns

The task sections discuss these failure modes in context. Table 2 brings them into one view and links each one to design choices reported in related work. The table is not a causal comparison; it is meant as a practical checklist for where a design choice helps and what should be inspected.

## E Resources Summary

Table 5 summarizes the resources discussed in Section 5, including benchmarks, datasets, environments, and toolkits. For each resource, we report its category, associated problem type, interactivity, temporal scale, number of series, and latest release when available.

## F Evaluation Evidence and Metrics

Table 6 reports representative comparable quantitative evidence across forecasting, reasoning, anomaly detection, trading, and traffic-control tasks. We only group results that share the same task, dataset, and metric, and we mark the surveyed agent methods in bold. The table is intended as comparable evidence rather than a universal leaderboard: scores may still depend on horizon, split, point-adjustment rule, transaction-cost assumption, or simulator configuration.

**Error metrics.** Regression-style tasks usually report point-wise errors such as mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE):

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

Here $y_i$ and $\hat{y}_i$ denote the target and prediction for the $i$-th evaluated point, and $n$ is the number of evaluated points. Lower values indicate better forecasts. Scale-free variants such as MAPE, sMAPE, or MASE are used when series with different magnitudes must be compared.

**Classification and detection metrics.** Reasoning, event prediction, anomaly detection, and some augmentation studies are commonly evaluated with accuracy, precision, recall, and F1. Let TP, FP, and FN denote true positives, false positives, and false negatives:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \quad \mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},$$
$$\mathrm{F1} = 2\frac{\mathrm{Precision}\cdot\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}}.$$

Macro-F1 averages class-wise F1 and is useful under class imbalance; micro-F1 aggregates counts before computing F1. In anomaly detection, F1 can be point-wise, point-adjusted, or event-based, so comparisons require the same adjustment rule. Score-based detectors may additionally use AUROC or AUPR.

Table 2: Summary of documented failure modes for time-series agent systems. Each failure mode is linked to related work, design choices, and implementation checks.

| Failure mode | Where it appears | Related work | Design choice | Implementation check |
|---|---|---|---|---|
| Error propagation across workflow stages | Forecasting and reasoning pipelines that pass outputs through diagnosis, model selection, context analysis, prediction, and validation | TimeSeriesScientist (Zhao et al., 2025); CLIMATEAGENT (Kim et al., 2025); CastFlow (Pan et al., 2026) | Keep stage outputs explicit. Log retrieved evidence, tool calls, predictions, and validation results so later stages do not inherit hidden errors | Can a reader inspect the artifact used by the next stage? |
| Numerical or domain reasoning errors | Tasks that require temporal structure, metadata, domain conventions, or quantitative operations | TS-Agent (Liu et al., 2025); TS-Reasoner (Ye et al., 2024b); CLIMATEAGENT (Kim et al., 2025) | Let the LLM plan and explain, but move calculation and domain-specific parsing into tools, operators, or specialist agents | Can the numerical claim be rerun outside the LLM? |
| Semantic drift in generated data | Augmentation and synthesis, where plausible-looking outputs can break trend, periodicity, local shape, correlation, or domain constraints | DCATS (Yeh et al., 2025); MERIT (Zhou et al., 2025); BRIDGE (Li et al., 2025) | Generate from target context and verify temporal properties before the synthetic data enters training or evaluation | Do checks cover trend, seasonality, local shape, and correlation rather than textual plausibility alone? |
| Brittle LLM-as-judge validation | Validation settings where a pretrained judge can approve an output without task evidence or executable checks | DCATS (Yeh et al., 2025); TS-Reasoner (Ye et al., 2024b) | Treat LLM critique as one signal. Require task metrics, execution feedback, or downstream validation before accepting an output | Would the output be rejected if another LLM approved it but the executable check failed? |
| Contamination during online updates | Streaming anomaly detection, where transient anomalies can be absorbed as the new normal | CALM (Devireddy and Huang, 2025) | Separate alarm decisions from update decisions; gate which observations can enter memory or fine-tuning data | Can short-lived anomalies be kept out of the update set? |
| Unsafe or infeasible actions in closed-loop control | Traffic and grid-control agents that issue executable actions under operational constraints | LLMLight (Lai et al., 2025); GridAgent (Zhang et al., 2025d); GridMind (Jin et al., 2025) | Check proposed actions with a simulator, solver, feasibility test, or safety validator before execution | Does every action pass an environment or constraint check first? |

**Generation and annotation metrics.** For augmentation and synthesis, evaluation is often indirect: generated samples are used to train or adapt a downstream model, and the downstream MAE, MSE, accuracy, or F1 is reported. Papers may also measure distributional fidelity with DTW, MMD, autocorrelation, spectral statistics, or nearest-neighbor analyses, and annotation quality with expert agreement or label accuracy.

**Decision and control metrics.** Trading agents are evaluated by both return and risk. Cumulative return (CR) measures portfolio growth, while the Sharpe ratio (SR) measures risk-adjusted return:

$$\mathrm{SR} = \frac{\mathbb{E}[R_t - R_f]}{\sigma(R_t - R_f)}.$$

Maximum drawdown, volatility, and turnover further capture downside risk and trading cost sensitivity. Recent trading-agent benchmarks such as StockBench (Chen et al., 2025) further stress multi-month sequential evaluation under daily market signals.

Traffic-control agents are usually evaluated in closed-loop simulators. Average travel time (ATT) is the mean travel duration across vehicles:

$$\mathrm{ATT} = \frac{1}{N}\sum_{i=1}^{N}(t_i^{\mathrm{exit}} - t_i^{\mathrm{entry}}),$$

where lower ATT indicates more efficient traffic flow. Related metrics include waiting time, queue length, throughput, cumulative reward, and constraint violations.

Table 3: Summary of representative LLM-based agentic methods for time-series tasks. Each method is grouped by **Problem Type** and further characterized by its publication year, venue or source, architecture, tools, and memory.

| Method | Year | Source | Problem Type | Architecture | Tools | Memory |
|---|---|---|---|---|---|---|
| TimeSeriesScientist (Zhao et al., 2025) | 2025 | arXiv | Forecasting | Multi (Cooperative) | Stat./ML Models | Evidence Logs |
| TimeXL (Jiang et al., 2025) | 2025 | NeurIPS | Forecasting | Multi (Cooperative) | None | Evidence Logs, Pattern Library |
| TimeCAP (Lee et al., 2025) | 2025 | AAAI | Forecasting | Multi (Cooperative) | Data Processing Tools | Pattern Library |
| NewsTSForecasting (Zhang et al., 2025e) | 2025 | arXiv | Forecasting | Multi (Competitive) | Search & Retrieval APIs | Evidence Logs, Analysis Strategies |
| TRACE (Chen and Xie, 2025) | 2025 | IEEE TAI | Forecasting | Multi (Competitive) | None | Evidence Logs |
| FLAIRR-TS (Jalori et al., 2025) | 2025 | EMNLP Findings | Forecasting | Multi (Cooperative) | None | Evidence Logs, Analysis Strategies |
| Nexus (Das et al., 2026) | 2026 | arXiv | Forecasting | Multi (Cooperative) | Data Processing Tools | Evidence Logs |
| CastFlow (Pan et al., 2026) | 2026 | arXiv | Forecasting | Multi (Mixed) | Data Processing Tools, Stat./ML Models | Evidence Logs, Pattern Library |
| TS-Agent (Liu et al., 2025) | 2025 | NeurIPS Wkshp. | Reasoning | Single | Data Processing Tools, Stat./ML Models | Evidence Logs |
| Agentic-RAG (Ravuru et al., 2024) | 2024 | KDD UC | Reasoning | Multi (Cooperative) | None | Pattern Library |
| TS-Reasoner (Ye et al., 2024b) | 2024 | arXiv | Reasoning | Single | Database APIs, Data Processing Tools, Stat./ML Models | Evidence Logs |
| ZARA (Li et al., 2026) | 2026 | ACL | Reasoning | Multi (Cooperative) | Database APIs, Data Processing Tools, Stat./ML Models | Pattern Library, Analysis Strategies |
| CLIMATEAGENT (Kim et al., 2025) | 2025 | arXiv | Reasoning | Multi (Cooperative) | Database APIs, Data Processing Tools, Simulators, Solvers & Optimizers | Evidence Logs |
| FETA (Sui et al., 2025) | 2025 | arXiv | Reasoning | Multi (Competitive) | Data Processing Tools, Search & Retrieval APIs | Pattern Library |
| TESSA (Lin et al., 2026) | 2026 | EACL | Augmentation | Multi (Cooperative) | Data Processing Tools | None |
| DCATS (Yeh et al., 2025) | 2025 | arXiv | Augmentation | Single | Stat./ML Models | Evidence Logs |
| MERIT (Zhou et al., 2025) | 2025 | ACL | Augmentation | Multi (Cooperative) | Data Processing Tools, Stat./ML Models | None |
| ChatTS (Xie et al., 2025) | 2025 | PVLDB | Synthesis | Multi (Cooperative) | None | None |
| GenAI4RiskModeling (Joshi, 2025) | 2025 | SSRN | Synthesis | Multi (Cooperative) | Database APIs, Data Processing Tools, Stat./ML Models | None |
| BRIDGE (Li et al., 2025) | 2025 | ICML | Synthesis | Multi (Mixed) | Search & Retrieval APIs, Stat./ML Models | Pattern Library |
| AD-AGENT (Yang et al., 2025) | 2025 | AACL | Detection | Multi (Cooperative) | Data Processing Tools, Stat./ML Models | Evidence Logs, Analysis Strategies |
| ARGOS (Gu et al., 2025) | 2025 | arXiv | Detection | Multi (Mixed) | Data Processing Tools, Stat./ML Models | Evidence Logs |
| SLEP (Wang et al., 2026) | 2026 | ESWA | Detection | Single | Data Processing Tools | Evidence Logs, Analysis Strategies |
| CALM (Devireddy and Huang, 2025) | 2025 | arXiv | Detection | Single | Data Processing Tools, Stat./ML Models | None |
| SAGE (Kang et al., 2026) | 2026 | arXiv | Detection | Multi (Cooperative) | Data Processing Tools, Stat./ML Models | Evidence Logs |
| LEMAD (Ji et al., 2025) | 2025 | Electron. | Diagnosis | Multi (Cooperative) | Data Processing Tools, Stat./ML Models | Evidence Logs |
| AgentFM (Zhang et al., 2025a) | 2025 | FSE | Diagnosis | Multi (Cooperative) | Database APIs, Data Processing Tools, Stat./ML Models | None |
| LLM-TSFD (Zhang et al., 2025b) | 2025 | ESWA | Diagnosis | Single | Database APIs, Data Processing Tools, Search & Retrieval APIs, Stat./ML Models | Pattern Library |
| ElliottAgents (Chudziak and Wawer, 2024; Wawer and Chudziak, 2025) | 2024 | PACLIC | Trading | Multi (Cooperative) | Data Processing Tools, Stat./ML Models | Pattern Library |
| TradingAgents (Xiao et al., 2024) | 2024 | arXiv | Trading | Multi (Mixed) | Data Processing Tools, Search & Retrieval APIs | Evidence Logs |
| FinCon (Yu et al., 2024) | 2024 | NeurIPS | Trading | Multi (Cooperative) | Database APIs, Data Processing Tools, Search & Retrieval APIs | Evidence Logs, Pattern Library, Analysis Strategies |
| FinAgent (Zhang et al., 2024a) | 2024 | KDD | Trading | Multi (Cooperative) | Data Processing Tools | Pattern Library, Analysis Strategies |
| FactorMAD (Duan et al., 2025) | 2025 | ICAIF | Trading | Multi (Competitive) | Data Processing Tools, Stat./ML Models | Pattern Library |
| FinMem (Yu et al., 2025) | 2025 | IEEE TBD | Trading | Single | Database APIs, Data Processing Tools | Evidence Logs, Pattern Library |
| FinArena (Xu et al., 2025) | 2025 | arXiv | Trading | Multi (Cooperative) | Data Processing Tools, Search & Retrieval APIs | None |
| ATLAS (Papadakis et al., 2025) | 2025 | arXiv | Trading | Multi (Cooperative) | Search & Retrieval APIs, Simulators, Solvers & Optimizers | Analysis Strategies |
| FinRL-DeepSeek (Benhenda, 2025) | 2025 | arXiv | Trading | Single | Stat./ML Models | None |
| Open-TI (Da et al., 2024) | 2024 | IJMLC | Traffic Control | Multi (Cooperative) | Database APIs, Data Processing Tools, Stat./ML Models | None |
| LLMTraveler (Wang et al., 2025) | 2025 | TR-C | Traffic Control | Single | None | Evidence Logs |
| LLMLight (Lai et al., 2025) | 2025 | KDD | Traffic Control | Single | Simulators, Solvers & Optimizers | None |
| CoLLMLight (Yuan et al., 2026) | 2026 | ICLR | Traffic Control | Multi (Mixed) | Data Processing Tools, Simulators, Solvers & Optimizers | Evidence Logs |
| HeraldLight (Guo et al., 2025) | 2025 | arXiv | Traffic Control | Multi (Mixed) | Stat./ML Models, Simulators, Solvers & Optimizers | Evidence Logs |
| Traffic-R1 (Zou et al., 2025) | 2025 | arXiv | Traffic Control | Single | Simulators, Solvers & Optimizers | Analysis Strategies |
| Virtual Traffic Police (Wei et al., 2026) | 2026 | arXiv | Traffic Control | Single | Search & Retrieval APIs, Simulators, Solvers & Optimizers | Evidence Logs |
| CuraLight (Guo et al., 2026) | 2026 | arXiv | Traffic Control | Multi (Competitive) | Simulators, Solvers & Optimizers | Evidence Logs |
| Grid-Agent (Zhang et al., 2025d) | 2025 | arXiv | Grid Control | Multi (Cooperative) | Simulators, Solvers & Optimizers | Evidence Logs |
| GridMind (Jin et al., 2025) | 2025 | SC Workshops | Grid Control | Multi (Cooperative) | Simulators, Solvers & Optimizers | Evidence Logs |

Table 4: Within-family counts and percentages of architecture, tool, and memory choices among the 47 systems summarized in Table 3. Tool and memory categories are non-exclusive.

| Task Family | Architecture | Tools | Memory |
|---|---|---|---|
| Forecasting & Reasoning ($n = 14$) | Single: 2 (14.3%); Multi (Cooperative): 8 (57.1%); Multi (Competitive): 3 (21.4%); Multi (Mixed): 1 (7.1%) | Database APIs: 3 (21.4%); Search & Retrieval APIs: 2 (14.3%); Data Processing Tools: 8 (57.1%); Stat./ML Models: 5 (35.7%); Simulators, Solvers & Optimizers: 1 (7.1%); None: 4 (28.6%) | Evidence Logs: 10 (71.4%); Pattern Library: 6 (42.9%); Analysis Strategies: 3 (21.4%); None: 0 (0%) |
| Augmentation & Synthesis ($n = 6$) | Single: 1 (16.7%); Multi (Cooperative): 4 (66.7%); Multi (Competitive): 0 (0%); Multi (Mixed): 1 (16.7%) | Database APIs: 1 (16.7%); Search & Retrieval APIs: 1 (16.7%); Data Processing Tools: 3 (50.0%); Stat./ML Models: 4 (66.7%); Simulators, Solvers & Optimizers: 0 (0%); None: 1 (16.7%) | Evidence Logs: 1 (16.7%); Pattern Library: 1 (16.7%); Analysis Strategies: 0 (0%); None: 4 (66.7%) |
| Anomaly Detection & Diagnosis ($n = 8$) | Single: 3 (37.5%); Multi (Cooperative): 4 (50.0%); Multi (Competitive): 0 (0%); Multi (Mixed): 1 (12.5%) | Database APIs: 2 (25.0%); Search & Retrieval APIs: 1 (12.5%); Data Processing Tools: 8 (100%); Stat./ML Models: 7 (87.5%); Simulators, Solvers & Optimizers: 0 (0%); None: 0 (0%) | Evidence Logs: 5 (62.5%); Pattern Library: 1 (12.5%); Analysis Strategies: 2 (25.0%); None: 2 (25.0%) |
| Decision Support ($n = 19$) | Single: 6 (31.6%); Multi (Cooperative): 8 (42.1%); Multi (Competitive): 2 (10.5%); Multi (Mixed): 3 (15.8%) | Database APIs: 3 (15.8%); Search & Retrieval APIs: 5 (26.3%); Data Processing Tools: 9 (47.4%); Stat./ML Models: 5 (26.3%); Simulators, Solvers & Optimizers: 9 (47.4%); None: 1 (5.3%) | Evidence Logs: 10 (52.6%); Pattern Library: 5 (26.3%); Analysis Strategies: 4 (21.1%); None: 4 (21.1%) |

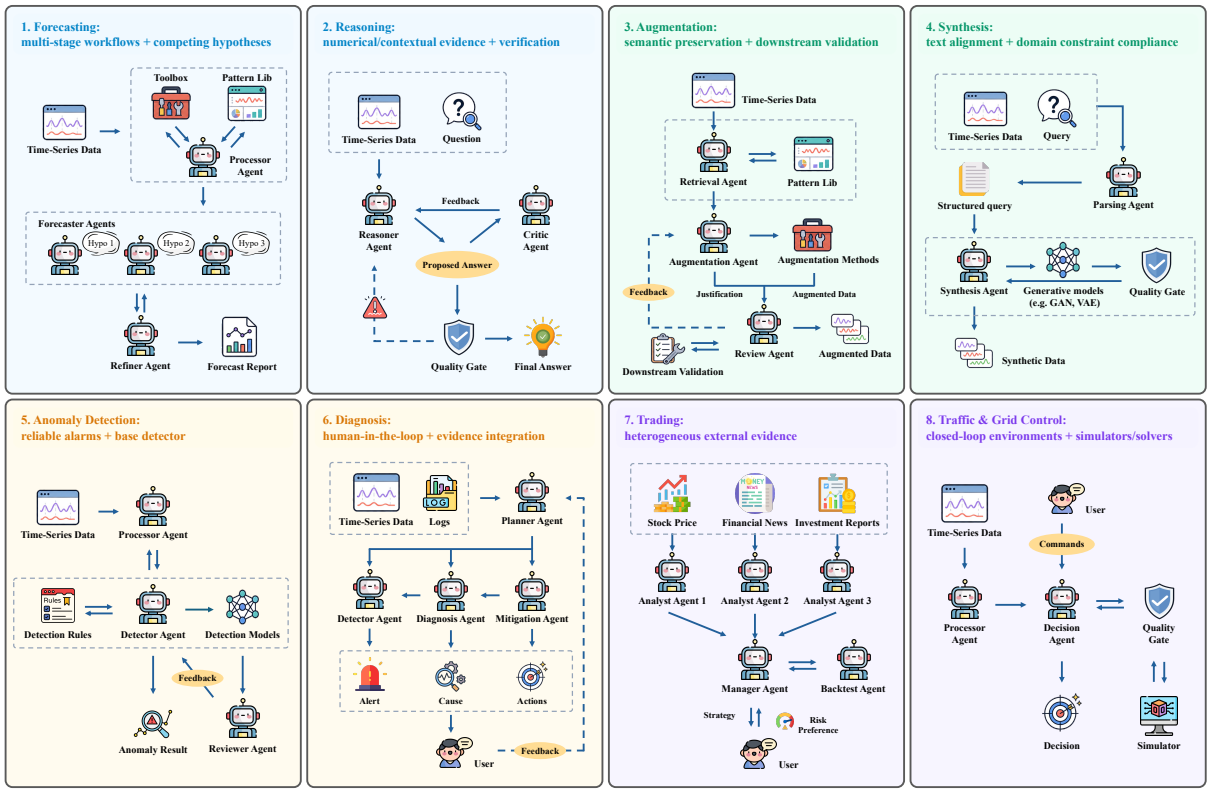



Figure 4: Typical design patterns of time-series agent systems across representative tasks. From top left to bottom right, the panels correspond to forecasting, reasoning, augmentation, synthesis, anomaly detection, diagnosis, trading, and traffic and grid control.

Table 5: Summary of representative resources for time-series tasks. Each resource is grouped by **Problem Type** and further characterized by its type, interactivity, temporal scale, number of series, and latest release.

| Resource | Category | Problem Type | Interactive | Timesteps | Series | Latest Release |
|---|---|---|---|---|---|---|
| *Forecasting* | | | | | | |
| M4 (Makridakis et al., 2018) | Benchmark | Forecasting | No | Up to 10k | ∼100k | 2018 |
| M5 (Makridakis et al., 2022) | Benchmark | Forecasting | No | 1.9k | 42k | 2020 |
| MIRAI (Ye et al., 2024a) | Benchmark | Forecasting | No | ∼ 1M | 59k | 2025 |
| FutureX (Zeng et al., 2025) | Benchmark | Forecasting | No | Live/Daily | 195 sources | 2025 |
| Monash TSF (Godahewa et al., 2021) | Dataset Repo | Multi-Domain Forecasting | No | Up to 527k | Up to 145k | 2021 |
| IHEPC | Dataset | Energy Forecasting | No | 2M | 9 | 2011 |
| Electricity (Lai et al., 2018) | Dataset | Energy Forecasting | No | 26k | 321 | 2015 |
| METR-LA (Li et al., 2017) | Dataset | Traffic Forecasting | No | 34k | 207 | 2018 |
| PEMS-BAY (Li et al., 2017) | Dataset | Traffic Forecasting | No | 52k | 325 | 2018 |
| ETT (Zhou et al., 2021) | Dataset | Energy Forecasting | No | Up to 69k | 7 | 2021 |
| Exchange (Lai et al., 2018) | Dataset | Finance Forecasting | No | 7.6k | 8 | 2021 |
| Traffic (Lai et al., 2018) | Dataset | Traffic Forecasting | No | 17k | 862 | 2023 |
| Weather | Dataset | Weather Forecasting | No | 53k | 21 | 2023 |
| LargeST (Liu et al., 2023) | Dataset | Traffic Forecasting | No | 526k | Up to 8.6k | 2023 |
| ILI | Dataset | Health Forecasting | No | Unclear | 7 | 2026 |
| StatsForecast (Garza et al., 2022) | Toolkit | Forecasting | No | N/A | N/A | 2025 |
| NeuralForecast (Challu et al., 2023) | Toolkit | Forecasting | No | N/A | N/A | 2026 |
| *Reasoning* | | | | | | |
| TimeSeriesExam (Cai et al., 2024) | Benchmark | Reasoning | No | N/A | >700 Tasks | 2024 |
| IBTrACS (Knapp et al., 2010) | Dataset | Reasoning | No | N/A | Unclear | 2025 |
| *Anomaly Detection* | | | | | | |
| TSB-UAD (Paparrizos et al., 2022) | Benchmark Suite | Anomaly Detection | No | Varies | 12.7k TS | 2022 |
| ADBench (Han et al., 2022) | Benchmark Suite | Anomaly Detection | No | Varies | 57 Datasets | 2022 |
| NAB (Lavin and Ahmad, 2015) | Benchmark | Anomaly Detection | No | Up to 22k | 58 | 2015 |
| UCR-AD (Wu and Keogh, 2021) | Benchmark | Anomaly Detection | No | Unclear | 250 | 2021 |
| SWaT (Mathur and Tippenhauer, 2016) | Dataset | Anomaly Detection | No | 11 days | 51 | 2016 |
| SMAP (Hundman et al., 2018) | Dataset | Anomaly Detection | No | 430k | 55 | 2018 |
| MSL (Hundman et al., 2018) | Dataset | Anomaly Detection | No | 67k | 27 | 2018 |
| WADI (Adepu et al., 2020) | Dataset | Anomaly Detection | No | Unclear | 103 | 2019 |
| KPI (Ren et al., 2019) | Dataset | Anomaly Detection | No | Unclear | ∼29 KPIs | 2019 |
| ADTK | Toolkit | Anomaly Detection | No | N/A | N/A | 2020 |
| Luminol | Toolkit | Anomaly Detection | No | N/A | N/A | 2017 |
| *Decision Support* | | | | | | |
| RL Unplugged (Gulcehre et al., 2020) | Dataset | Offline RL | No | N/A | N/A | 2020 |
| D4RL (Fu et al., 2020) | Dataset | Offline RL | No | N/A | N/A | 2021 |
| Grid2Op (Marot et al., 2021) | Environment | Power System Control | Yes | N/A | N/A | 2026 |
| SocioDojo (Cheng and Chin, 2024) | Environment | Trading | Yes | N/A | N/A | 2024 |
| CityLearn | Environment | Building Energy Control | Yes | N/A | N/A | 2025 |
| SUMO (Behrisch et al., 2011) | Environment | Traffic Control | Yes | N/A | N/A | 2026 |
| Stable-Baselines3 (Raffin et al., 2021) | Toolkit | Reliable RL | No | N/A | N/A | 2026 |
| *Others* | | | | | | |
| TimeSeriesGym (Cai et al., 2025) | Benchmark | Multi-Task | No | N/A | N/A | 2025 |
| Sktime (Löning et al., 2019) | Toolkit | Multi-Task | No | N/A | N/A | 2025 |
| GluonTS (Alexandrov et al., 2020) | Toolkit | Multi-Task | No | N/A | N/A | 2025 |
| Merlion (Bhatnagar et al., 2021) | Toolkit | Multi-Task | No | N/A | N/A | 2024 |

Table 6: Comparable quantitative evidence across time-series tasks (Part 1). Agent methods are highlighted in bold.

| Task | Dataset | Metric | Method | Score |
|---|---|---|---|---|
| *ETTh1 forecasting* (Zhou et al., 2021) | | | | |
| Forecasting | ETTh1 | MAE | **FLAIRR-TS** (Jalori et al., 2025) | 0.101 |
| Forecasting | ETTh1 | MAE | LSTP (Liu et al., 2024a) | 0.150 |
| Forecasting | ETTh1 | MAE | DLinear (Zeng et al., 2023) | 0.390 |
| Forecasting | ETTh1 | MAE | **Agentic-RAG (Llama3-8B)** (Ravuru et al., 2024) | 0.396 |
| Forecasting | ETTh1 | MAE | GPT4TS (Zhou et al., 2023) | 0.397 |
| Forecasting | ETTh1 | MAE | PatchTST (Nie et al., 2023) | 0.399 |
| Forecasting | ETTh1 | MAE | TimesNet (Wu et al., 2023) | 0.402 |
| Forecasting | ETTh1 | MAE | FEDFormer (Zhou et al., 2022) | 0.419 |
| Forecasting | ETTh1 | MAE | Time-LLM (Jin et al., 2024) | 0.460 |
| *Weather forecasting* (Lee et al., 2025) | | | | |
| Forecasting | Weather | F1 | **TimeXL (GPT-4o)** (Jiang et al., 2025) | 0.696 |
| Forecasting | Weather | F1 | **TimeCAP (GPT-4)** (Lee et al., 2025) | 0.668 |
| Forecasting | Weather | F1 | FreTS (Yi et al., 2023) | 0.623 |
| Forecasting | Weather | F1 | Time-LLM (Jin et al., 2024) | 0.613 |
| Forecasting | Weather | F1 | PatchTST (Nie et al., 2023) | 0.592 |
| Forecasting | Weather | F1 | LLMTime (Gruver et al., 2023) | 0.587 |
| Forecasting | Weather | F1 | Autoformer (Wu et al., 2021) | 0.546 |
| Forecasting | Weather | F1 | iTransformer (Liu et al., 2024c) | 0.541 |
| Forecasting | Weather | F1 | DLinear (Zeng et al., 2023) | 0.540 |
| Forecasting | Weather | F1 | Crossformer (Zhang and Yan, 2023) | 0.500 |
| Forecasting | Weather | F1 | PromptCast (Xue and Salim, 2023) | 0.499 |
| Forecasting | Weather | F1 | TimesNet (Wu et al., 2023) | 0.494 |
| Forecasting | Weather | F1 | TSMixer (Chen et al., 2023) | 0.488 |
| *TimeSeriesExam reasoning* (Cai et al., 2024) | | | | |
| Reasoning | TimeSeriesExam | Accuracy | **TS-Agent (GPT-4o-mini)** (Liu et al., 2025) | 0.55 |
| Reasoning | TimeSeriesExam | Accuracy | GPT-o1 | 0.37 |
| Reasoning | TimeSeriesExam | Accuracy | GPT-4o | 0.29 |
| Reasoning | TimeSeriesExam | Accuracy | DeepSeek | 0.28 |
| Reasoning | TimeSeriesExam | Accuracy | Phi-3.5 | 0.25 |
| *HAR average reasoning* (Li et al., 2026) | | | | |
| Reasoning | HAR average | Macro-F1 | **ZARA (Gemini-2.0-Flash)** (Li et al., 2026) | 0.814 |
| Reasoning | HAR average | Macro-F1 | UniMTS (Zhang et al., 2024c) | 0.321 |
| Reasoning | HAR average | Macro-F1 | IMU2CLIP (Moon et al., 2023) | 0.179 |
| Reasoning | HAR average | Macro-F1 | ImageBind (Girdhar et al., 2023) | 0.142 |
| Reasoning | HAR average | Macro-F1 | Gemini Plot | 0.135 |
| Reasoning | HAR average | Macro-F1 | Gemini Table | 0.130 |
| Reasoning | HAR average | Macro-F1 | Gemini Text | 0.129 |
| Reasoning | HAR average | Macro-F1 | HARGPT Text (Ji et al., 2024) | 0.104 |
| Reasoning | HAR average | Macro-F1 | IMUGPT (Leng et al., 2023) | 0.100 |
| Reasoning | HAR average | Macro-F1 | HARGPT Plot (Ji et al., 2024) | 0.099 |
| Reasoning | HAR average | Macro-F1 | NormWear (Luo et al., 2024) | 0.077 |
| *KPI anomaly detection* (Ren et al., 2019) | | | | |
| Anomaly detection | KPI | F1 | **ARGOS (GPT-4o)** (Gu et al., 2025) | 0.897 |
| Anomaly detection | KPI | F1 | LSTMAD (Malhotra et al., 2015) | 0.819 |
| Anomaly detection | KPI | F1 | FCVAE (Wang et al., 2024b) | 0.818 |
| Anomaly detection | KPI | F1 | **MERIT (Llama3.1-8B)** (Zhou et al., 2025) | 0.815 |
| Anomaly detection | KPI | F1 | TimesURL (Liu and Chen, 2024) | 0.803 |
| Anomaly detection | KPI | F1 | TS2Vec (Yue et al., 2022) | 0.791 |
| Anomaly detection | KPI | F1 | SR (Ren et al., 2019) | 0.785 |
| Anomaly detection | KPI | F1 | DONUT (Xu et al., 2018) | 0.779 |
| Anomaly detection | KPI | F1 | DSPOT (Siffer et al., 2017) | 0.768 |
| Anomaly detection | KPI | F1 | SPOT (Siffer et al., 2017) | 0.751 |
| Anomaly detection | KPI | F1 | AutoRegression | 0.668 |
| Anomaly detection | KPI | F1 | TFAD (Zhang et al., 2022a) | 0.564 |
| Anomaly detection | KPI | F1 | AnomalyTransformer (Xu et al., 2021) | 0.282 |

Table 7: Comparable quantitative evidence across time-series tasks (Part 2). Agent methods are highlighted in bold.

| Task | Dataset | Metric | Method | Score |
|---|---|---|---|---|
| *Yahoo anomaly detection* (Laptev, 2015) | | | | |
| Anomaly detection | Yahoo | F1 | **MERIT (Llama3.1-8B)** (Zhou et al., 2025) | 0.905 |
| Anomaly detection | Yahoo | F1 | TimesURL (Liu and Chen, 2024) | 0.892 |
| Anomaly detection | Yahoo | F1 | TS2Vec (Yue et al., 2022) | 0.885 |
| Anomaly detection | Yahoo | F1 | SR (Ren et al., 2019) | 0.879 |
| Anomaly detection | Yahoo | F1 | DONUT (Xu et al., 2018) | 0.873 |
| Anomaly detection | Yahoo | F1 | DSPOT (Siffer et al., 2017) | 0.861 |
| Anomaly detection | Yahoo | F1 | SPOT (Siffer et al., 2017) | 0.847 |
| Anomaly detection | Yahoo | F1 | **ARGOS (GPT-4o)** (Gu et al., 2025) | 0.810 |
| Anomaly detection | Yahoo | F1 | TFAD (Zhang et al., 2022a) | 0.773 |
| Anomaly detection | Yahoo | F1 | AutoRegression | 0.526 |
| Anomaly detection | Yahoo | F1 | FCVAE (Wang et al., 2024b) | 0.464 |
| Anomaly detection | Yahoo | F1 | LSTMAD (Malhotra et al., 2015) | 0.350 |
| Anomaly detection | Yahoo | F1 | LLMAD (GPT-4-32k) (Liu et al., 2024b) | 0.142 |
| *AAPL single-asset trading* (Yu et al., 2024) | | | | |
| Trading | AAPL | SR | **FinCon (GPT-4-Turbo)** (Yu et al., 2024) | 1.597 |
| Trading | AAPL | SR | FinGPT (GPT-4-Turbo) (Yang et al., 2023) | 1.161 |
| Trading | AAPL | SR | B&H | 1.107 |
| Trading | AAPL | SR | DQN | 1.048 |
| Trading | AAPL | SR | **FinAgent (GPT-4-Turbo)** (Zhang et al., 2024a) | 1.041 |
| Trading | AAPL | SR | **FinMem (GPT-4-Turbo)** (Yu et al., 2025) | 0.994 |
| Trading | AAPL | SR | PPO | 0.704 |
| Trading | AAPL | SR | A2C | 0.683 |
| Trading | AAPL | SR | GA (GPT-4-Turbo) (Park et al., 2023) | 0.372 |
| *AMZN single-asset trading* (Yu et al., 2024) | | | | |
| Trading | AMZN | SR | **FinCon (GPT-4-Turbo)** (Yu et al., 2024) | 0.904 |
| Trading | AMZN | SR | DQN | 0.398 |
| Trading | AMZN | SR | PPO | 0.138 |
| Trading | AMZN | SR | B&H | 0.072 |
| Trading | AMZN | SR | GA (GPT-4-Turbo) (Park et al., 2023) | -0.199 |
| Trading | AMZN | SR | A2C | -0.444 |
| Trading | AMZN | SR | **FinMem (GPT-4-Turbo)** (Yu et al., 2025) | -0.773 |
| Trading | AMZN | SR | **FinAgent (GPT-4-Turbo)** (Zhang et al., 2024a) | -1.493 |
| Trading | AMZN | SR | FinGPT (GPT-4-Turbo) (Yang et al., 2023) | -1.810 |
| *Jinan 1 traffic signal control* (Lai et al., 2025) | | | | |
| Traffic control | Jinan 1 | ATT | **LightGPT (Llama2-13B)** (Lai et al., 2025) | 274.03 |
| Traffic control | Jinan 1 | ATT | Advanced-CoLight (Zhang et al., 2022b) | 274.67 |
| Traffic control | Jinan 1 | ATT | **LightGPT (Llama3-8B)** (Lai et al., 2025) | 275.10 |
| Traffic control | Jinan 1 | ATT | GPT-4 | 275.26 |
| Traffic control | Jinan 1 | ATT | Efficient-CoLight (Zhang et al., 2022b) | 277.11 |
| Traffic control | Jinan 1 | ATT | CoLight (Wei et al., 2019b) | 279.60 |
| Traffic control | Jinan 1 | ATT | Maxpressure | 281.58 |
| Traffic control | Jinan 1 | ATT | FixedTime | 481.79 |
| Traffic control | Jinan 1 | ATT | Random | 597.62 |
| *Open-TI Config1 traffic signal control* (Da et al., 2024) | | | | |
| Traffic control | Open-TI Config1 | ATT | **Open-TI (GPT-4.0)** (Da et al., 2024) | 103.46 |
| Traffic control | Open-TI Config1 | ATT | PressLight (Wei et al., 2019a) | 107.39 |
| Traffic control | Open-TI Config1 | ATT | DQN | 162.19 |
| Traffic control | Open-TI Config1 | ATT | SOTL | 218.06 |
| Traffic control | Open-TI Config1 | ATT | FixedTime | 552.72 |