LLM Agents for Time-Series: A Survey

Anonymous ACL submission

Abstract

Recent advances in large language models (LLMs) have accelerated the development of agentic systems for time-series analysis. While traditional time-series methods typically make predictions or decisions based on a given set of evidence, many real-world applications require agentic systems that can autonomously plan workflows, reflect on intermediate results, and leverage external tools and memory. This survey presents a systematic review of LLM-based agents for time-series tasks. We adopt a problem-driven taxonomy, organizing existing systems into four problem categories: Forecasting & Reasoning, Data Augmentation & Synthesis, Anomaly Detection & Diagnosis, and Decision Support. For each category, we analyze how agent behaviors are implemented through architectural design, external tool integration, and memory mechanisms. We further discuss datasets, environments, and evaluation protocols, and outline open challenges and future directions for LLM-based time-series agents. Overall, our goal is to provide researchers with a structured view of how LLM-based agents can be designed to address different time-series problems.

1 Introduction

Time-series analysis (Hamilton, 2020) plays an important role in many real-world domains, including finance (Tsay, 2005; Taylor, 2008; Franses and Van Dijk, 2000), healthcare (Crabtree et al., 1990; Morid et al., 2023), energy systems (Deb et al., 2017), and transportation (Xu et al., 2016; Ghosh et al., 2009). 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 et al., 2001) and Transformers (Vaswani et al., 2017).

However, applying these methods in practice often relies heavily on expert knowledge to design analysis pipelines and interpret results. Recent advances in large language models (LLMs) (Zhao et al., 2023; Chang et al., 2024) offer an alternative by leveraging general-purpose language reasoning to assist time-series analysis, for example as modules for encoding and explaining temporal patterns. Beyond this prompt-based use, LLMs can also be deployed as *agents* (Wang et al., 2024a) that iteratively plan actions, call external tools, and maintain memory across multiple steps. Such agentic systems are particularly suitable for time-series applications because many time-series tasks require iterative refinement as new observations arrive, adaptive planning 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., 2025). Motivated by this emerging paradigm, this survey reviews recent progress on LLM-based agent systems for time-series tasks.

Positioning. Existing LLM-for-time-series surveys (Chang et al., 2025; ZHANG et al., 2025) typically organize methods by individual LLM capabilities (e.g., planning, reasoning, memory, and tool use), rather than by how these capabilities are integrated to solve 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 challenges that are central to time-series settings, including streaming inputs, temporal dependence, semantic drift, and action–data coupling. Moreover, there is still a lack of surveys that focus specifically on LLM agents for time-series problems. To bridge this gap, we adopt a *problem-driven taxonomy* that groups meth-

ods by the tasks they target and then analyzes recurring design patterns in architecture, tool use, and memory mechanisms. A detailed comparison with closely related surveys is provided in Appendix A.

Contributions. This survey makes four contributions: (i) We propose a problem-driven taxonomy and summarize representative systems in each family. (ii) We analyze recurring agentic patterns and cross-cutting system components (architectures, tools, memory). (iii) We connect task properties (e.g., drift, delayed feedback, operational constraints) to design choices and evaluation protocols. (iv) We curate practical resources and discuss open problems and future directions.

2 Background and Foundations

Time-Series Data and Representations. A time series consists of temporally ordered observations indexed by $\{\tau_t\}_{t=1}^{T}$ with $\tau_1 < \cdots < \tau_T$. We denote a (possibly multivariate) sequence as

$$\mathbf{X}_{1:T} = (\mathbf{x}_1, \ldots, \mathbf{x}_T), \quad \mathbf{x}_t \in \mathbb{R}^d.$$

In practice, time-series data may be irregularly sampled, noisy, or partially observed. Systems therefore operate under partial observability and typically construct representations $\mathbf{z}_t = \phi(\mathbf{X}_{1:t})$ that summarize historical context.

Time-series Modeling. Time-series models can be viewed as inference modules that map observed evidence to beliefs or predictions: statistical models (e.g., ARIMA, ARCH/GARCH, HMMs) encode explicit assumptions and interpretable uncertainty (Shumway and Stoffer, 2017; Engle, 1982; Bollerslev, 1986; Rabiner, 1989), deep models (e.g., LSTM, TCN, DeepAR) learn nonlinear mappings via gradient-based optimization (Hochreiter and Schmidhuber, 1997; Bai et al., 2018; Salinas et al., 2020), and LLM-based models (e.g., Time-LLM, LSTPrompt) represent sequences in tokenized or multimodal forms to support reasoning and tool orchestration (Jin et al., 2023; Liu et al., 2024). Despite these differences, most remain one-shot, inference-only systems that estimate future or latent quantities from past data without acting to shape outcomes.

From Inference to Agentic Systems. Static prediction pipelines are often inadequate for interactive, decision-driven time-series tasks that require iterative evidence gathering, feedback integration, and adaptive action selection, which has motivated the emergence of LLM agents. Prior surveys (Wang et al., 2024a; ZHANG et al., 2025) characterize such systems as observe–act–update loops that combine reasoning, planning, tool use, and memory. We adopt this view and analyze how these capabilities are realized across problem settings.

3 Fundamental Design Dimensions

Before introducing the task taxonomy, we first characterize time-series agent systems along three fundamental design dimensions: architecture, tools, and memory. Together, these dimensions capture how an agent is organized, how it interacts with external resources, and how it retains information across multi-step workflows. This section provides a concise classification of these dimensions to support the task-specific analysis that follows.

3.1 Architecture

We identify the following architectural patterns for time-series agent systems; a visual summary is provided in Appendix B.

Adaptive Single-Agent. The most basic agentic form: one LLM dynamically selects tools and reasoning paths without a fixed workflow, interleaving reasoning and action as evidence arrives (Yao et al., 2023; Schick et al., 2023; Wang et al., 2024a). This distinguishes it from non-agentic LLM use.

Iterative Single-Agent. These systems use explicit self-refine/reflect loops: a single agent generates an output, critiques it, and revises across structured iterations (Madaan et al., 2023; Shinn et al., 2023; Chang et al., 2025). Compared to adaptive agents, the loop is fixed and each step keeps evaluation traces for the next round.

Lifelong / Continual Single-Agent. These systems learn continuously from streaming experience through explicit self-updating mechanisms: a single agent changes over time via training-based adaptation, rather than merely retaining memory, modifying prompts, or retrieving additional context (Parisi et al., 2019; Zheng et al., 2026).

Sequential Pipelines. Multi-stage workflows allow later-stage validation to trigger partial re-execution of earlier stages, blending structure with feedback adaptivity (Chang et al., 2025).

Committee / Debate Systems. Multiple agents generate diverse hypotheses in parallel and aggregate through debate or voting, emphasizing diversity and robustness (Liang et al., 2024).

Planner-Executor Systems. Planner-Executor structures decompose tasks into subtasks and coor-

dinate specialized components to solve them, improving modularity and scalability (Wooldridge, 2009; Wang et al., 2024a).

3.2 Tools

Tools are callable interfaces to external programs invoked by the LLM agent (Wang et al., 2024b). 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*, which transform raw inputs into more useful representations, such as through alignment or 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., 2024c). We summarize it into three categories: *(i) Evidence logs*, which record what happened during 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) Analytical 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 that performs multi-step decision-making over time-indexed evidence, where the LLM is centrally responsible for action selection, memory updating, or hypothesis revision under current observations. We include methods for time-series tasks in which the LLM selects and sequences the workflow and adapts it based on intermediate results or external evidence. 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* (forecasting, analytical QA, explanation), *Data Augmentation & Synthesis* (data quality, new sequences/annotations), *Anomaly Detection & Diagnosis* (detect–then–explain), and *Decision Support* (actionable recommendations under constraints, often via simulation/backtesting). Within each category, we further distinguish several sub-problems. Table 2 summarizes the categories. Figure 1 shows the full hierarchical 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 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., 2025b). As a result, effective systems usually do not treat context as a static input. Instead, they actively construct evidence beyond the context window during inference, for example by retrieving relevant slices on demand (Jalori et al., 2025) or querying prototype memories (Jiang et al., 2025). We discuss them separately below.

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. A full workflow often involves data diagnosis, preprocessing, model or hyperparameter selection, and validation, and errors in early stages can easily

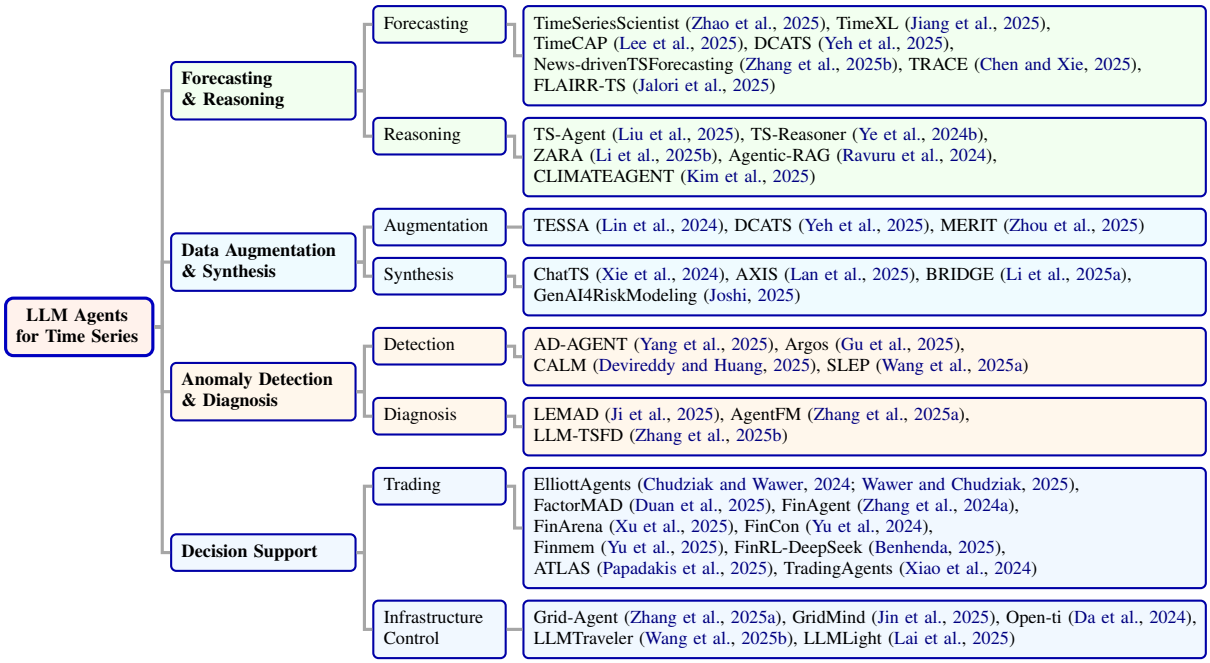

Figure 1: A taxonomy of LLM agents for time-Series.

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 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, rollback, and correction.

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. A common design is to use committee/debate-style architectures, where different agents represent different forecasting views and are compared through explicit error feedback. NewsTSForecasting (Zhang et al., 2025b) follows this idea by using error-based scoring and acceptance tests 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 intermediate communication is checked against evidence.

Time-Series Reasoning. Time-series reasoning agents aim to derive explanations, diagnoses, or decisions from temporal evidence rather than directly predict 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 naturally encode temporal structure, domain mechanisms, or reliable 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., 2025b) mines discriminative features offline and stores feature–domain relations as reusable guidance, while CLIMATEAGENT (Kim et al., 2025) uses specialized data agents to handle API conventions, metadata retrieval, and parameter validation.

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 are closely related, as both need evidence beyond a fixed context window and can suffer from error accumulation in long workflows. These shared challenges motivate similar memory mechanisms: patterns library helps retrieve relevant precedents, while evidence logs preserve intermediate steps for reflection and revision. Their tool use is also broadly aligned, with heavy reliance on data processing and frequent calls to operators or statistical/ML models for quantitative support. The main difference lies in architecture: forecasting more often compares competing hypotheses through committee/debate designs, whereas reasoning more often relies on iterative single-agent loops to reduce long-chain errors.

4.2 Time-Series Augmentation & Synthesis

Time-series augmentation and synthesis both aim to construct additional data while preserving time-series semantics. In both settings, the 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. We compare them separately below.

Time-Series Augmentation. Time-series 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. A common remedy is to first learn general semantic descriptions from cross-domain annotations by decontextualizing them into domain-agnostic concepts (e.g., trend, periodicity, volatility), and then convert the general semantic annotations into domain-specific annotations via a domain agent (Lin et al., 2024).

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-data construction without any text guidance. 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., 2025a) extracts templates (e.g., length, trend, periodicity, extrema, variance), refines them, and trains a controlled diffusion generator. GenAI4RiskModeling (Joshi, 2025) similarly rewrites narratives into structured scenario–

query forms and lets the agent translate them into hyperparameter settings for GAN/VAE-based generation.

Domain constraint compliance is the main challenge in domain-data construction without text control. In this setting, the difficulty is not text alignment but ensuring that generated data still conforms to implicit domain rules and constraints. A common design is therefore to let the agent first choose a set of controllable domain attributes, then translate them into executable generation rules, and finally filter inconsistent outputs through downstream verification, keeping only aligned supervision signals (Xie et al., 2024).

Discussion: For augmentation and synthesis, the central issue is that generated outputs may look plausible while drifting away from task-relevant semantics. As a result, current systems are often organized around explicit construction–verification pipelines rather than unconstrained LLM generation. This emphasis is also reflected in tool use, which centers on data processing, augmentation or generation modules, and downstream validation signals. In existing work, memory is less central here than in forecasting or reasoning. And when it appears, it mainly appears as 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, and we discuss them separately below.

Time-Series Anomaly Detection. Under this survey's definition, time-series anomaly detection focuses on producing reliable alarms, such as pointwise 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., 2025a) follows this direction by enriching detector inputs with additional evidence rather than relying only on the raw sequence.

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 agent filters training data by distinguishing transient noise from sustained distribution shift, and only the latter is used for short-horizon adaptation of the forecasting-based detector.

Time-Series Diagnosis. Under this survey's definition, 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 by retrieving labeled historical faults through RAG as few-shot references, which guides the prompt and reduces 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 by introducing a human-in-the-loop data preparation stage, where users specify labeling or cleaning intent in natural language and the system gener-

ates executable processing code that is iteratively refined through feedback.

Discussion: Anomaly detection and diagnosis are both concerned with building reliable and actionable monitoring pipelines from heterogeneous temporal evidence. This makes tool support especially important: both settings rely heavily on data processing, existing detectors or analytical modules, and external evidence sources. Sequential pipelines therefore remain the dominant architectural pattern, although streaming monitoring scenarios further motivate continual agent systems that can update over time. Memory serves a supporting but important role: evidence logs help keep intermediate artifacts auditable, while pattern-library memory allows retrieval of historical incidents or labeled examples to stabilize decisions.

4.4 Time-Series Decision Making

Time-series decision making 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. The key cross-domain difference is constraint type: trading usually tolerates some risk, whereas infrastructure control (e.g., traffic or power systems) often impose hard feasibility and safety constraints. We therefore examine them separately below.

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 design is a planner–executor architecture, 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; a common design is to inject user risk preferences and feedback into prompts in a human-in-the-loop manner, so that they directly influence the final recommendation (Xu et al., 2025). In other settings, the system does not explicitly incorporate user feedback, but instead constructs internal role-based variation to induce different risk styles, again typically through prompt injection (Xiao et al., 2024; Yu et al., 2025).

Infrastructure Control. These agents mainly target control tasks in infrastructure systems, especially traffic and power systems. Compared with trading, infrastructure control is shaped mainly by two features: (i) hard constraints, and (ii) human-specified policies.

Hard constraints are central in infrastructure control because decisions usually operate in strict environments, where feasibility and safety must be validated by the environment or external tools. As a result, simulators or solvers are often essential as verification tools. A common design is a tool-grounded sequential pipeline, in which the LLM proposes candidate actions or plans, while simulators determine feasibility and effect. GridAgent (Zhang et al., 2025a) is a representative example: the LLM proposes structured mitigation plans, while power-flow solvers and validation modules determine their performance. Similarly, LLMLight (Lai et al., 2025) operates in a fixed control space and is evaluated in a traffic simulator.

Human-specified policies are another distinctive feature of infrastructure control. 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.

Discussion: Different decision targets lead to different design principles for trading and infrastructure-control agents. Trading systems more often use planner–executor designs to decompose multimodal evidence, whereas infrastructure-control systems more often rely on sequential execution–validation pipelines. The same contrast appears in memory and tool use: trading emphasizes historical experience (pattern libraries, analytical strategies) and heterogeneous-data tooling, while infrastructure control depends more on simulators or solvers for feasibility and safety checks.

5 Resources for Implementation and Evaluation

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 3).

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 standardized 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 Conclusion and Future Work

This survey presents the first comprehensive review of LLM-based agentic systems for time-series problems. We propose a novel problem-driven taxonomy that organizes existing methods by the problems they address and highlights recurring design patterns across tasks. While this taxonomy provides a practical lens for understanding and designing time-series agents, several important challenges remain open for future research.

Trustworthiness and Validation. Although recent agent systems improve flexibility and efficiency, their trustworthiness remains limited (Jin et al., 2024; Raza et al., 2025). In many time-series settings, reliable validation methods such as simulation and backtesting are unavailable, so evaluation still relies on LLM-as-a-judge or weak heuristic checks (Tomašević et al., 2025). More reliable validation methods are therefore needed.

Numerical Understanding and Domain Knowledge. LLMs are general language models pretrained mainly on text, and remain limited in understanding numerical signals and specialized domain knowledge (Hung et al., 2023; Ye et al., 2024b). Many current agent designs implicitly assume that pretrained reasoning can transfer directly to time-series settings. Yet even when external tools provide quantitative results, agents still need the ability to interpret them, judge their reliability, and connect them to domain-specific reasoning. Improving this capability remains an important direction.

Online Adaptation and Continual Improvement. In real-world applications such as anomaly detection and decision making, agents often need to improve continuously after deployment. Yet most current systems remain largely static and rely mainly on memory updates, while parameter adaptation remains rare (Jaglan and Barnes, 2025; Zheng et al., 2026). A key next step is to enable stronger online adaptation in deployed agents.

Benchmarking Agent Workflows. Current studies still evaluate time-series agents mainly with standard task metrics. This leaves an important question: how should we evaluate agent workflows themselves, beyond downstream results—for example, in terms of the quality of intermediate decisions, the appropriateness of tool use, and the effectiveness of reflection? Recent work has begun to expose this gap (Weng et al., 2026; Cheng et al., 2026). Future work should therefore benchmark agent workflows more systematically.

7 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.

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

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, and 1 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 J"org 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 1 others. 2025. A survey of reasoning and agentic systems in time series with large language models. *arXiv preprint arXiv:2509.11575*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

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

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.

Benjamin F Crabtree, Subhash C Ray, Priscilla M Schmidt, Patrick T O'Connor, and David D Schmidt. 1990. The individual over time: time series applications in health care research. *Journal of clinical epidemiology*, 43(3):241–260.

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.

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

Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah. 2017. A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74:902–924.

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*.

Philip Hans Franses and Dick Van Dijk. 2000. *Nonlinear time series models in empirical finance*. Cambridge university press.

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.

Bidisha Ghosh, Biswajit Basu, and Margaret O'Mahony. 2009. Multivariate short-term traffic flow forecasting using time-series analysis. *IEEE transactions on intelligent transportation systems*, 10(2):246–254.

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*.

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 (2025). *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, and 1 others. 2020. Rl unplugged: A suite of benchmarks for offline reinforcement learning. *Advances in neural information processing systems*, 33:7248–7259.

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 Ar ik. 2025. Flairr-ts–forecasting llm-agents with iterative refinement and retrieval for time series. *arXiv preprint arXiv:2508.19279*.

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, Yifan Li, Shirui Pan, and Qingsong Wen. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*. Accepted at ICLR 2024.

Ming Jin, Yifan Zhang, Wei Chen, Kexin Zhang, Yuxuan Liang, Bin Yang, Jindong Wang, Shirui Pan, and Qingsong Wen. 2024. Position: What can large language models tell us about time series analysis. In *Forty-first International Conference on Machine Learning*.

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

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.

Tian Lan, Hao Duong Le, Jinbo Li, Wenjun He, Meng Wang, Chenghao Liu, and Chen Zhang. 2025. Axis: Explainable time series anomaly detection with large language models. *arXiv preprint arXiv:2509.24378*.

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.

Hao Li, Yuhao Huang, Chang Xu, Viktor Schlegel, Renhe Jiang, Riza Batista-Navarro, Goran Nenadic, and Jiang Bian. 2025a. Bridge: Bootstrapping text to control time-series generation via multi-agent iterative optimization and diffusion modelling. *arXiv preprint arXiv:2503.02445*.

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. 2025b. Zara: Zero-shot motion time-series analysis via knowledge and retrieval driven llm agents. *arXiv preprint arXiv:2508.04038*.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. *Preprint*, arXiv:2305.19118.

Minhua Lin, Zhengzhang Chen, Yanchi Liu, Xujiang Zhao, Zongyu Wu, Junxiang Wang, Xiang Zhang, Suhang Wang, and Haifeng Chen. 2024. Decoding time series with llms: A multi-agent framework for cross-domain annotation. *arXiv preprint arXiv:2410.17462*.

Haoxin Liu, Zhiyuan Zhao, Jindong Wang, Harshavardhan Kamarthi, and B. Aditya Prakash. 2024. 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.

Penghang Liu, Elizabeth Fons, Svitlana Vyetrenko, Daniel Borrajo, Vamsi Potluru, and Manuela Veloso. 2025. Ts-agent: A time series reasoning agent with iterative statistical insight gathering. *arXiv preprint arXiv:2510.07432*.

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.

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*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Swabha Swayamdipta, Yiming Yang, and Hannaneh Hajishirzi. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

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.

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, Lakhmi Jain, and 1 others. 2001. Recurrent neural networks. *Design and applications*, 5(64-67):2.

Mohammad Amin Morid, Olivia R Liu Sheng, and Joseph Dunbar. 2023. Time series prediction using deep learning methods in healthcare. *ACM Transactions on Management Information Systems*, 14(1):1–29.

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.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.

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*.

Shaina Raza, Ranjan Sapkota, Manoj Karkee, and Christos Emmanouilidis. 2025. Trism for agentic ai: A review of trust, risk, and security management in llm-based agentic multi-agent systems. *arXiv preprint arXiv:2506.04133*.

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.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Preprint*, arXiv:2302.04761.

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.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

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

Xiaoyu Tao and 1 others. 2026. Cast-r1: Learning tool-augmented sequential decision policies for time series forecasting. *arXiv preprint arXiv:2602.13802*.

Stephen J Taylor. 2008. *Modelling financial time series*. world scientific.

Aleksandar Tomašević, Darja Cvetković, Sara Major, Slobodan Maletić, Miroslav Anđelković, Ana Vranić, Boris Stupovski, Dušan Vudragović, Aleksandar Bogojević, and Marija Mitrović Dankulov. 2025. Towards operational validation of llm-agent social simulations: A replicated study of a reddit-like technology forum. *arXiv preprint arXiv:2508.21740*.

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. 2025a. Large-model-based smart agent for time series anomaly detection in power systems. *Expert Systems with Applications*, page 128917.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 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. 2025b. Agentic large language models for day-to-day route choices. *Transportation Research Part C: Emerging Technologies*, 180:105307.

Zhiruo Wang, Zhoujun Cheng, Hao Zhu, Daniel Fried, and Graham Neubig. 2024b. 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*.

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.

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. 2024. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *arXiv preprint arXiv:2412.03104*.

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.

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. 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*.

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, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, and 1 others. 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.

Zhiyuan Zeng, Jiashuo Liu, Siyuan Chen, Tianci He, Yali Liao, Yixiao Tian, Jinpeng Wang, Zaiyuan Wang, Yang Yang, Lingyue Yin, and 1 others. 2025. Futurex: An advanced live benchmark for llm agents in future prediction. *arXiv preprint arXiv:2508.11987*.

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, SIQI GOH, XINGSHENG CHEN, ZONGRU LI, HONGGANG WEN, JIALE GUO, MENGLIN YANG, HONGZHI YIN, QIANG YANG, SIU-MING YIU, and 1 others. 2025. From prompts to agents: A comprehensive survey of llm-driven time series analysis. *arXiv preprint*.

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, and 1 others. 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*.

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

Yuxuan Zhang, Yangyang Feng, Daifeng Li, Kexin Zhang, Junlan Chen, and Bowen Deng. 2025b. 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. 2024c. 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. Timeseries-scientist: 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, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

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.

A Survey Comparison

This section positions our survey against closely related surveys on LLMs for time-series analysis. Table 1 compares them along five dimensions:

(i) *TS-specific*, which indicates whether the survey is explicitly designed for time-series problems rather than general LLM or agent settings;

(ii) *LLM Agents*, which indicates whether the survey focuses on agentic systems in which the LLM plays an active role in multi-step decision-making, where "Partial" means that agentic systems are included but are not the main focus;

(iii) The four task columns, which indicate whether the survey clearly covers each major problem family in our taxonomy—*Forecasting & Reasoning*, *Augmentation & Synthesis*, *Anomaly Detection & Diagnosis*, and *Decision Support*—where ✓ denotes clear coverage, ∼ partial coverage, and × that the topic is not a main focus;

(iv) *Taxonomy*, which describes the main organizing principle of the survey;

(v) *Design Guidance*, which indicates whether the survey provides practical guidance for selecting or understanding system designs in concrete settings. Under *Taxonomy*, "Problem-driven" groups methods by time-series problems, "Capability-driven" by agent capabilities such as planning, memory, or tool use, "Method-driven" by methodological categories such as prompting, fine-tuning, tokenization, or model integration, and "Other" covers organizing principles that do not fit these categories cleanly. Under *Design Guidance*, "Limited" means that guidance is provided only indirectly or at a high level.

B Architecture Overview

Figure 2 provides a visual summary of the architectural patterns discussed in Section 3. It organizes time-series agent systems into single-agent and multi-agent settings and summarizes the main architectural patterns in our taxonomy, including adaptive, iterative, and continual single-agent designs, as well as sequential pipeline, planner–executor, and committee/debate systems.

C Table 2: Method Summary

Table 2 summarizes the representative methods covered in this survey. The methods are organized primarily by *Problem Type*, following the problem-driven taxonomy adopted in this work. For each method, the table reports its publication *Year* and *Venue*, and summarizes its classification along three agent-oriented dimensions: *Architecture*, *Tools*, and *Memory*. Detailed discussions are provided in Section 4.

D Table 3: Resources Summary

Table 3 provides a supplementary summary of the resources discussed in Section 5. The resources are organized by *Problem Type*, following the problem-driven taxonomy adopted in this work, and include benchmarks, datasets, environments, and toolkits. For each listed resource, the table reports the following practical attributes: (i) *Category*, the category of resource, such as benchmark, dataset, environment, toolkit, or dataset repository; (ii) *Problem Type*, the primary time-series problem or application domain the resource is associated with; (iii) *Interactive*, whether the resource supports interactive or closed-loop evaluation, where agent actions can affect subsequent observations or outcomes; (iv) *Timesteps*, the approximate temporal length of the data, when such information is available; (v) *Series*, the approximate number of variables or channels in the resource, when applicable; and (vi) *Last Update*, the most recent reported release, update year, or maintenance status. This table is intended as a compact reference to the practical characteristics of commonly used resources.

E Typical Design Patterns Across Time-Series Tasks

Figure 3 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.

In the figure, (i) dashed boxes denote agent modules together with their tools, while solid boxes group entities with similar roles; (ii) solid arrows show the main flow, while dashed arrows indicate conditional or fallback paths; and (iii) background colors are used to group panels from the same higher-level problem type.

Table 1: Comparison with related 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., 2025)	2025	Yes	Partial	✓	✓	✓	✓	Other	Limited
(ZHANG et al., 2025)	2025	Yes	Partial	✓	✓	✓	∼	Capability-driven	Limited
Ours	2026	Yes	Yes	✓	✓	✓	✓	Problem-driven	Yes

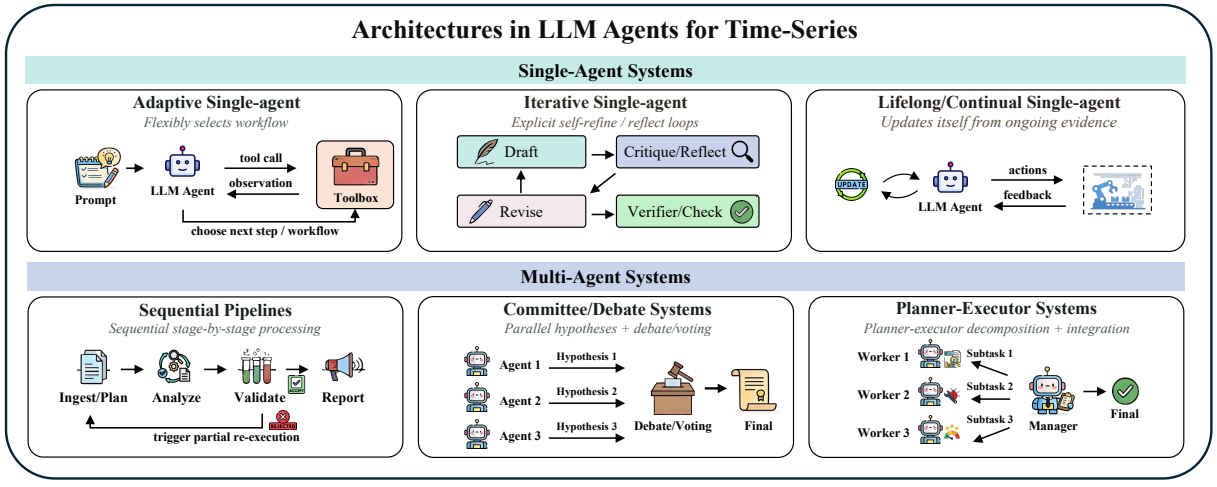

Figure 2: An illustration of the main architectural patterns for time-series agent systems.

Table 2: 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, architecture, tools, and memory.

Method	Year	Venue	Problem Type	Architecture	Tools	Memory
TimeSeriesScientist (Zhao et al., 2025)	2024	arXiv	Forecasting	Multi (Sequential Pipeline)	Statistical & ML Models	Evidence Logs
TimeXL (Jiang et al., 2025)	2025	arXiv	Forecasting	Multi (Sequential Pipeline)	None	Evidence Logs, Pattern Library
TimeCAP (Lee et al., 2025)	2025	AAAI	Forecasting	Multi (Sequential Pipeline)	Data Processing	Pattern Library
NewsTSForecasting (Zhang et al., 2025b)	2025	arXiv	Forecasting	Multi (Hybrid)	Search & Retrieval APIs	Evidence Logs, Analytical Strategies
TRACE (Chen and Xie, 2025)	2025	TAI	Forecasting	Multi (Committee / Debate)	None	Evidence Logs
FLAIRR-TS (Jalori et al., 2025)	2025	arXiv	Forecasting	Multi (Sequential Pipeline)	None	Evidence Logs, Analytical Strategies
TS-Agent (Liu et al., 2025)	2024	arXiv	Reasoning	Single (Iterative)	Data Processing, Statistical & ML Models	Evidence Logs
Agentic-RAG (Ravuru et al., 2024)	2024	arXiv	Reasoning	Multi (Planner–Executor)	None	Pattern Library
TS-Reasoner (Ye et al., 2024b)	2024	arXiv	Reasoning	Single (Iterative)	Database APIs, Data Processing, Statistical	Evidence Logs
ZARA (Li et al., 2025b)	2025	arXiv	Reasoning	Multi (Sequential Pipeline)	Database APIs; Data Processing; Statistical & ML Models	Pattern Library, Analytical Strategies
CLIMATEAGENT (Kim et al., 2025)	2025	arXiv	Reasoning	Multi (Planner–Executor)	Database APIs, Data Processing, Simulators & Solvers & Optimizers	Evidence Logs
TESSA (Lin et al., 2024)	2024	arXiv	Augmentation	Multi (Sequential Pipeline)	Data Processing	None
DCATS (Yeh et al., 2025)	2025	arXiv	Augmentation	Single (Iterative)	Statistical & ML Models	Evidence Logs
MERIT (Zhou et al., 2025)	2025	ACL	Augmentation	Multi (Sequential Pipeline)	Data Processing, Statistical & ML Models	None
ChatTS (Xie et al., 2024)	2024	ICSP	Synthesis	Multi (Sequential Pipeline)	None	None
AXIS (Lan et al., 2025)	2025	arXiv	Synthesis	Multi (Sequential Pipeline)	None	None
GenAI4RiskModeling (Joshi, 2025)	2025	SSRN	Synthesis	Multi (Sequential Pipeline)	Database APIs, Data Processing, Statistical & ML Models	None
BRIDGE (Li et al., 2025a)	2025	arXiv	Synthesis	Multi (Hybrid)	Search APIs, Statistical & ML Models	Pattern Library
AD-AGENT (Yang et al., 2025)	2025	arXiv	Detection	Multi (Sequential Pipeline)	Data Processing, Statistical & ML Models	Evidence Logs, Analytical Strategies
Argos (Gu et al., 2025)	2025	arXiv	Detection	Multi (Sequential Pipeline)	Data Processing, Statistical & ML Models	Evidence Logs
SLEP	2025	SSRN	Detection	Single (Adaptive)	Data Processing	Evidence Logs, Analytical Strategies
CALM	2025	arXiv	Detection	Single (Lifelong / Continual)	Data Processing, Statistical & ML Models	None
LEMAD (Zhang et al., 2025b)	2025	Electron.	Diagnosis	Multi (Sequential Pipeline)	Data Processing, Statistical & ML Models	Evidence Logs
AgentFM (Zhang et al., 2025a)	2025	FSE	Diagnosis	Multi (Planner–Executor)	Database APIs, Data Processing, Statistical & ML Models	None
LLM-TSFD (Zhang et al., 2025b)	2025	ESWA	Diagnosis	Single (Iterative)	Database APIs, Data Processing, Search & Retrieval APIs, Statistical & ML Models	Pattern Library
ElliottAgents (Chudziak and Wawer, 2024; Wawer and Chudziak, 2025)	2024	PACLIC	Trading	Multi (Planner–Executor)	Data Processing, Statistical & ML Models	Pattern Library
TradingAgents (Xiao et al., 2024)	2024	arXiv	Trading	Multi (Hybrid)	Data Processing, Search & Retrieval APIs	Evidence Logs
FinCon (Yu et al., 2024)	2024	NeurIPS	Trading	Multi (Planner–Executor)	Database APIs, Data Processing, Search & Retrieval APIs	Evidence Logs, Pattern Library, Analytical Strategies
FinAgent (Zhang et al., 2024a)	2024	WWW	Trading	Multi (Sequential Pipeline)	Data Processing	Pattern Library, Analytical Strategies
FactorMAD (Duan et al., 2025)	2024	arXiv	Trading	Multi (Committee / Debate)	Data Processing, Statistical & ML Models	Patterns Library
FinMem (Yu et al., 2025)	2025	AAAI	Trading	Single (Adaptive)	Database APIs, Data Processing	Evidence Logs, Patterns Library
FinArena (Xu et al., 2025)	2025	arXiv	Trading	Multi (Planner–Executor)	Data Processing, Search & Retrieval APIs	None
ATLAS (Papadakis et al., 2025)	2025	arXiv	Trading	Multi (Planner–Executor)	Search & Retrieval APIs, Simulators & Solvers & Optimizers	Analytical Strategies
FinRL-DeepSeek (Benhenda, 2025)	2025	arXiv	Trading	Single (Adaptive)	Statistical & ML Models	None
Open-TI (Da et al., 2024)	2024	arXiv	Infrastructure Control	Multi (Sequential Pipeline)	Database APIs, Data Processing, Statistical & ML Models	None
Grid-Agent (Zhang et al., 2025a)	2024	arXiv	Infrastructure Control	Multi (Sequential Pipeline)	Simulators & Solvers & Optimizers	Evidence Logs
GridMind (Jin et al., 2025)	2025	SC Workshops	Infrastructure Control	Multi (Planner–Executor)	Simulators & Solvers & Optimizers	Evidence Logs
LLMTraveler (Wang et al., 2025b)	2025	TRC	Infrastructure Control	Single (Iterative)	None	Evidence Logs
LLMLight (Lai et al., 2025)	2025	KDD	Infrastructure Control	Single (Iterative)	Simulators & Solvers & Optimizers	None

Table 3: 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 update status.

Resource	Category	Problem Type	Interactive	Timesteps	Series	Last Update
Forecasting						
M4 (Makridakis et al., 2018)	Benchmark	Forecasting	No	Up to 10k	∼100k	2020
M5 (Makridakis et al., 2022)	Benchmark	Forecasting	No	1.9k	42k	2022
MIRAI (Ye et al., 2024a)	Benchmark	Forecasting	No	∼ 1M	59k	2025
FutureX (Zeng et al., 2025)	Benchmark	Forecasting	No	∼ 1k	195	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	Ongoing
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	236 Datasets	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	Unclear	Unclear	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	781	2019
ADTK	Toolkit	Anomaly Detection	No	N/A	N/A	2020
Luminol	Toolkit	Anomaly Detection	No	N/A	N/A	2023
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	2021
SocioDojo (Cheng and Chin, 2024)	Environment	Trading	Yes	N/A	N/A	2024
CityLearn	Environment	Power System Control	Yes	N/A	N/A	2025
SUMO (Behrisch et al., 2011)	Environment	Traffic Control	Yes	N/A	N/A	Ongoing
Stable-Baselines3 (Raffin et al., 2021)	Toolkit	Reliable RL	No	N/A	N/A	2025
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

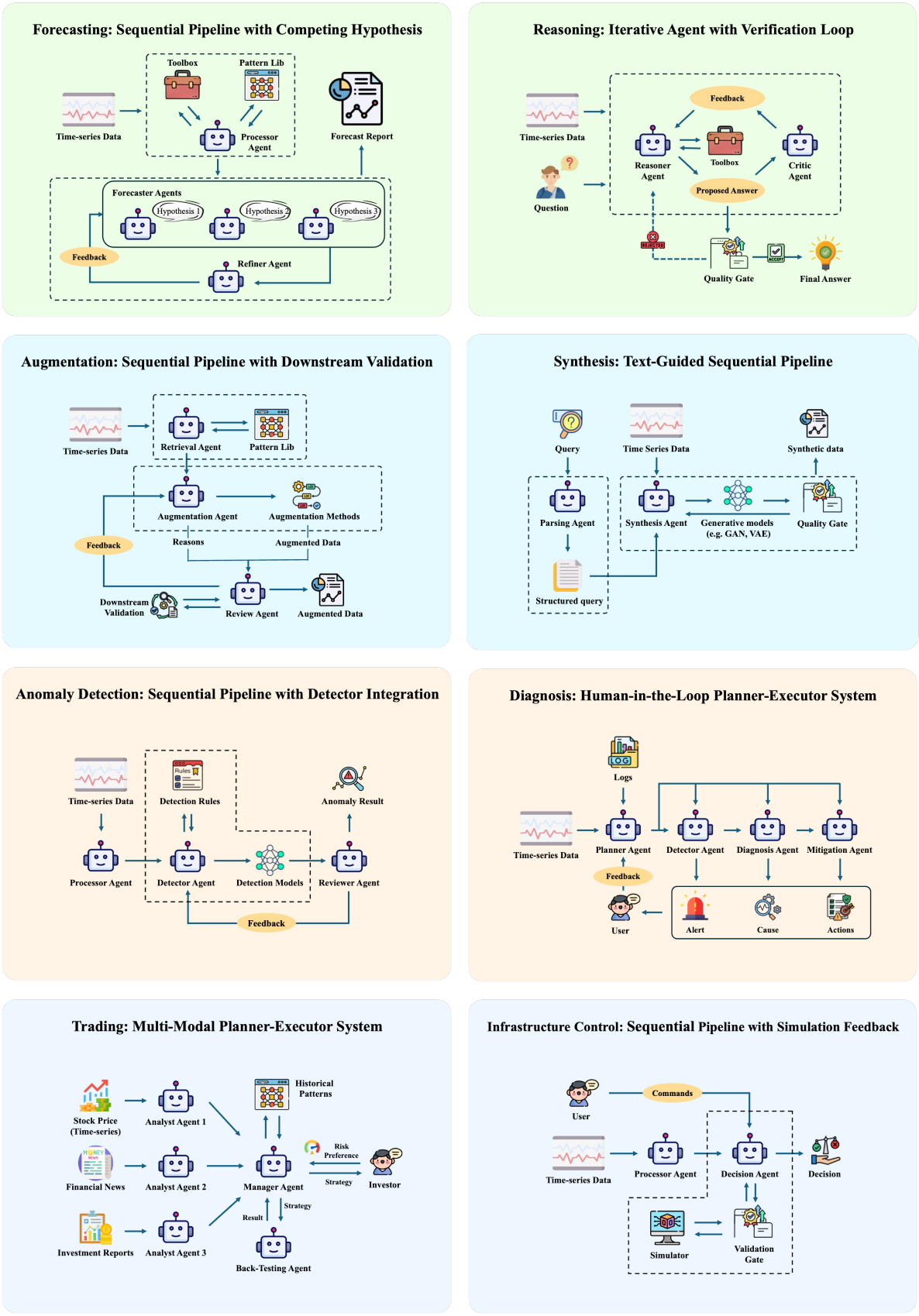

Figure 3: Typical design patterns of time-series agent systems across representative tasks. From top left to bottom right, the panels correspond to forecasting, reasoning, augmentation, synthesis, anomaly detection, diagnosis, trading, and infrastructure control.