

Evaluating Large Language Models for Time Series Anomaly Detection in Aerospace Software

Yang Liu¹, Yixing Luo^{1*}, Xiaofeng Li¹, Xiaogang Dong^{1*}, Bin Gu¹, Zhi Jin²

¹Software Department, Beijing Institute of Control Engineering, Beijing, China

²School of Computer Science, Peking University, Beijing, China

Email: liuyang12339@163.com, {luoyi_xing, li_x_feng}@126.com

dongxiaogang@263.net, gubinbj@sina.com, zhjin@pku.edu.cn

Abstract—Time series anomaly detection (TSAD) is essential for ensuring the safety and reliability of aerospace software systems. Although large language models (LLMs) provide a promising training-free alternative to unsupervised approaches, their effectiveness in aerospace settings remains under-examined because of complex telemetry, misaligned evaluation metrics, and the absence of domain knowledge. To address this gap, we introduce ATSADBENCH, the first benchmark for aerospace TSAD. ATSADBENCH comprises nine tasks that combine three pattern-wise anomaly types, univariate and multivariate signals, and both in-loop and out-of-loop feedback scenarios, yielding 108,000 data points. Using this benchmark, we systematically evaluate state-of-the-art open-source LLMs under two paradigms: DIRECT, which labels anomalies within sliding windows, and PREDICTION-BASED, which detects anomalies from prediction errors. To reflect operational needs, we reformulate evaluation at the window level and propose three user-oriented metrics: Alarm Accuracy (AA), Alarm Latency (AL), and Alarm Contiguity (AC), which quantify alarm correctness, timeliness, and credibility. We further examine two enhancement strategies, few-shot learning and retrieval-augmented generation (RAG), to inject domain knowledge. The evaluation results show that (1) LLMs perform well on univariate tasks but struggle with multivariate telemetry, (2) their AA and AC on multivariate tasks approach random guessing, (3) few-shot learning provides modest gains whereas RAG offers no significant improvement, and (4) in practice LLMs can detect true anomaly onsets yet sometimes raise false alarms, which few-shot prompting mitigates but RAG exacerbates. These findings offer guidance for future LLM-based TSAD in aerospace software.

Index Terms—Time Series Anomaly Detection, Large Language Model, Aerospace Software

I. INTRODUCTION

With the rapid growth and increasing complexity of modern software systems, ensuring their safety and reliability has become a critical concern. Among various assurance techniques, time series anomaly detection (TSAD) plays a vital role in monitoring software system health by identifying abnormal behaviors in operational data. Effective anomaly detection enables timely responses to system faults, performance degradation, or potential security attacks. Conversely, failure to detect anomalies promptly can result in cascading failures, prolonged downtime, and severe economic losses as well as safety risks, particularly in large-scale, safety-critical scenarios.

Deep learning has become the dominant approach for TSAD. Supervised approaches, which require labeled data,

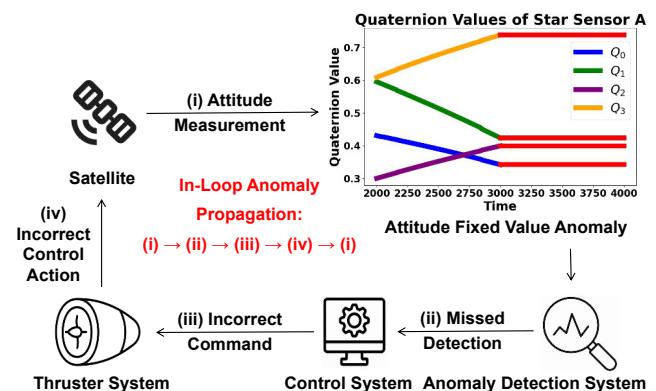


Fig. 1: A missed detection of a fixed-value anomaly in star sensor attitude measurements can propagate through the control loop, causing erroneous commands and continuous thruster firing, ultimately resulting in critical satellite damage.

have achieved significant success but are limited by the difficulty of obtaining sufficient labeled anomalous data in real-world systems [1]–[6]. This limitation has led to the rise of unsupervised approaches, which detect anomalies by learning normal patterns from unlabeled data [7]–[33]. However, these unsupervised approaches often struggle with generalization and transferability across diverse time series, requiring separate models to be trained for each individual series. This results in unaffordable maintenance costs in large-scale systems where numerous time series need to be monitored [34].

Recent advances have demonstrated the application of Large Language Models (LLMs) to time series anomaly detection, despite these models originally being designed for natural language processing tasks [23], [35], [36]. These models exhibit strong generalization capabilities, eliminating the need for training on specific time series and offering a promising, training-free alternative approach. Alnegheimish et al. [37] introduce the SIGLLM framework, which adapts LLMs through two paradigms: DIRECT and PREDICTION-BASED, showing that zero-shot LLMs can successfully find anomalies in univariate time series. To improve the anomaly detection performance and interpretability of LLMs, [38] proposed the LLMAID framework for univariate time series. This framework

*Corresponding authors.

enhances LLMs by leveraging domain knowledge injection and Chain-of-Thought (CoT). To the best of our knowledge, no work has explored the anomaly detection capabilities of LLMs on multivariate time series to date.

Despite their potential, applying LLMs to TSAD in aerospace is largely unexplored, facing three fundamental challenges: **(1) Complexity of Aerospace Data:** Aerospace telemetry exhibits highly diverse anomaly types and complex inter-variable relationships, governed by physical laws and feedback loops. Anomalies may propagate through the system (e.g., an undetected fixed-value anomaly in a star sensor leading to incorrect control actions and potential satellite damage, as illustrated in Fig. 1). Such complexity makes effective anomaly detection particularly challenging for generic LLMs. **(2) Misalignment of Evaluation Metrics:** Widely used point-based classification metrics, such as F1-score, do not reflect operational practice. In real scenarios, LLMs analyze windows of telemetry data, and users are concerned with whether a window contains actionable anomalies, rather than individual point-wise labels. Thus, current metrics do not adequately measure practical detection performance. **(3) Lack of Domain Knowledge:** LLMs, trained primarily on general-purpose data, lack the domain-specific understanding necessary for aerospace applications, such as physical control laws, sensor failure modes, and operational procedures, which limits their ability to detect and interpret critical anomalies.

To assess the performance of LLMs on diverse anomaly types in aerospace, we construct ATSADBENCH, the first comprehensive benchmark specifically designed for aerospace TSAD. ATSADBENCH encompasses nine representative tasks covering fixed-value, constant deviation, and time-varying deviation anomalies, across both univariate (U) and multivariate (M) signals, and includes both in-loop (M-IL) and out-of-loop (M-OL) feedback scenarios, totaling 108,000 data points. Using this benchmark, we systematically evaluate state-of-the-art open-source LLMs under two principal paradigms: DIRECT, which involves identifying anomalies within windows, and PREDICTION-BASED, which detects anomalies based on prediction errors. To more closely align evaluation with practical user needs, we recast the assessment as a window-based task and introduce three novel user-oriented metrics: Alarm Accuracy (AA), Alarm Latency (AL), and Alarm Contiguity (AC). These metrics collectively capture the correctness, timeliness, and reliability of alarms from an operational standpoint. Additionally, we explore enhancement strategies for LLMs, including few-shot learning and retrieval-augmented generation (RAG), evaluating their impact on anomaly detection performance with conventional and proposed metrics.

Our evaluation yields four main findings: First, LLMs perform notably well in univariate tasks, though they still lag behind SOTA unsupervised approaches, and they show a slight advantage in M-IL over M-OL; the PREDICTION-BASED paradigm consistently outperforms the DIRECT paradigm. Second, standard point-wise metrics fail to reflect operational requirements. Our proposed window-based metrics (AA, AL, AC) enable a more realistic and actionable assessment of

anomaly detection for aerospace applications. Third, Few-shot learning can enhance LLM performance under the DIRECT paradigm, while RAG provides only marginal or inconsistent improvements. Finally, a case study on a real in-orbit anomaly was conducted, showing that LLMs are capable of providing timely anomaly alarms, but may generate false positives, and few-shot learning can help mitigate false alarms.

The contributions of this paper are as follows:

- We present the first systematic evaluation of large language models for time series anomaly detection in aerospace applications. To facilitate this, we introduce ATSADBENCH, a comprehensive benchmark comprising nine tasks that encompass three pattern-wise anomaly types and three control-loop scenarios, enabling rigorous comparison with SOTA unsupervised approaches.
- We propose three user-oriented window-level metrics (AA, AL, AC) for practical evaluation to assess the correctness, timeliness, and credibility of alarms from an operational perspective.
- We investigate enhancement strategies, including few-shot learning and RAG, and validate their effectiveness through experiments and a real-world anomaly case study, revealing their practical benefits and limitations for LLM-based detection in aerospace contexts.

II. BACKGROUND AND RELATED WORK

A. Time Series Anomaly Detection

Time series anomaly detection refers to the task of identifying data points or subsequences within a temporal sequence that deviate significantly from the expected normal behavior. In this context, an anomaly is a data point or interval whose characteristics differ markedly from the majority of the time series. This task is essential in software reliability engineering and has been dominated by deep learning approaches, which leverage sophisticated models to capture temporal dependencies and inter-variable correlations. Approaches can generally be categorized as supervised or unsupervised, depending on the availability of labeled data.

1) Supervised Anomaly Detection: Supervised approaches cast anomaly detection as a classification problem, requiring a labeled training dataset \mathcal{T} where each data point $\mathbf{x}_t \in \mathcal{T}$ is annotated as normal or anomalous. The model is trained to distinguish between these classes and subsequently labels each test data point during inference. While effective, this approach is often impractical for real-world software systems due to the scarcity of anomalies and the high cost of manual labeling.

2) Unsupervised Anomaly Detection: Unsupervised approaches do not rely on labeled data. These approaches typically assume anomalies are rare within the training set $\mathcal{T} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ and treat most data as normal. Both training and test data are segmented into overlapping windows via a sliding window strategy. Models are trained to learn normal behavior and, during inference, assign an anomaly score to each data point, reflecting its deviation from expected patterns. Data points with scores exceeding a predefined

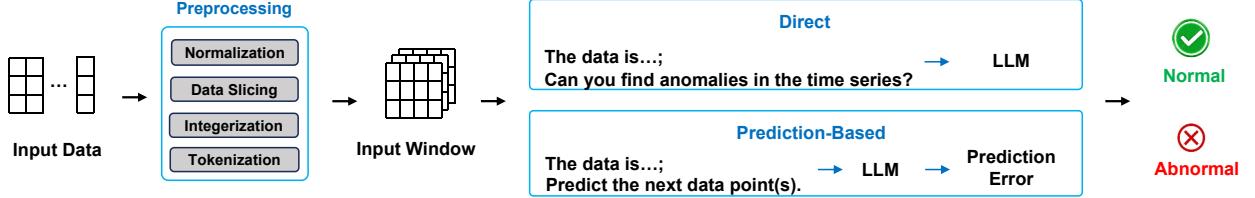


Fig. 2: Pipeline of time series anomaly detection with zero-shot LLMs

threshold are flagged as anomalous. Unsupervised anomaly detection approaches can be further categorized according to how anomaly scores are generated:

- **DIRECT paradigm:** These approaches detect anomalies by directly assigning an anomaly score to each data point within a window, based on its conformity to learned normal patterns. Representative approaches include:
 - **Reconstruction-based approach:** It encodes input windows into low-dimensional representations and reconstructs them. High reconstruction errors indicate anomalies [11], [12], [14], [17], [19], [22], [39].
 - **Contrastive learning-based approach:** It learns representations where normal and anomalous data points are well-separated by using contrastive objectives during training. Samples that deviate from learned normal patterns receive higher anomaly scores [40], [41].
- **PREDICTION-BASED paradigm:** These approaches forecast future data points based on the current window and classify points as anomalies if the prediction error is significant [15], [16], [18], [42], [43].

3) *Zero-Shot Anomaly Detection with LLMs:* LLMs with extensive pre-training and remarkable generalization capabilities have recently enabled innovative approaches to time series anomaly detection. Leveraging zero-shot learning, LLMs can identify anomalies in sequential data without requiring task-specific training or labeled examples. As shown in Fig. 2, the zero-shot workflow comprises two primary steps: preprocessing and anomaly detection:

- **Preprocessing** includes normalization, slicing the time series into windows, integerization (rescaling to integer values), and tokenization. For LLMs such as GPT [44] and DeepSeek [45], digits are separated by spaces for accurate tokenization [46], [47]; in contrast, LLaMA [48] and Qwen [49] support direct numerical tokenization.
- **Anomaly Detection** is performed via two paradigms as follows [37]:
 - **DIRECT paradigm:** The LLM is prompted to identify anomalies directly within each window, returning indices of anomalous data points.
 - **PREDICTION-BASED paradigm:** The LLM predicts the next data point(s) according to the input window, and deviations between predictions and actual values are used to detect anomalies.

B. Related Work

Recent years have seen extensive empirical research on time series anomaly detection. Audibert et al. [50] provided a comparative study encompassing traditional, machine learning, and deep neural network approaches, while Li et al. [51] conducted a comprehensive evaluation of unsupervised deep learning algorithms. Sarfraz et al. [52] critically examined state-of-the-art deep learning techniques, benchmarking them against simpler baselines. Additionally, Freeman et al. [53] offered a broad survey of twelve algorithms, ranging from statistical models to early neural network approaches, assessed across diverse time series characteristics.

With the advent of LLMs, research attention has shifted toward their potential in anomaly detection [23], [35], [36]. Recent work [37], [38] have demonstrated that LLMs can detect anomalies with minimal training, but primarily evaluated univariate time series. In contrast, our work is the first to systematically investigate LLMs for multivariate time series anomaly detection, particularly in complex industrial settings such as multi-sensor aerospace telemetry. This fills a critical gap by assessing whether LLMs can model the intricate inter-variable dependencies inherent in real-world systems.

Beyond detection algorithms, evaluation methodology has received increasing attention. Si et al. [34] proposed the “All-in-One” paradigm for unified model assessment, while Puccetti et al. [54] stressed the importance of detection latency. Kim et al. [55] cautioned that the widely used “point adjustment” protocol can artificially inflate performance scores. In response, we introduce novel, user-oriented evaluation metrics specifically designed for aerospace operational contexts, capturing not only detection accuracy but also the timeliness and credibility of anomaly alerts, which are key considerations in safety-critical applications.

Finally, the choice of evaluation datasets is essential for rigorous benchmarking. Wu et al. [56] identified limitations in many mainstream datasets and released the UCR Time Series Anomaly Archive as a more robust alternative, while Si et al. [34] provided the industrial-scale TimeSeriesBench dataset to reflect real-world complexity. In contrast, we construct a new benchmark dataset derived from authentic aerospace telemetry, encompassing diverse anomaly types and both out-of-loop and in-loop scenarios. This application-driven design ensures that our evaluation aligns with the actual challenges faced in industrial practice, setting our work apart from previous studies on generic datasets.

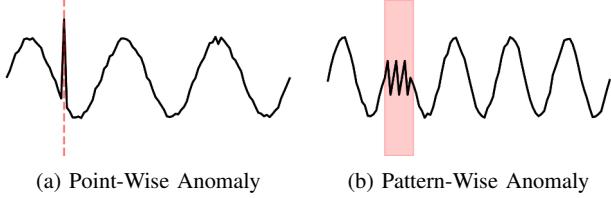


Fig. 3: Overview of time series anomaly types.

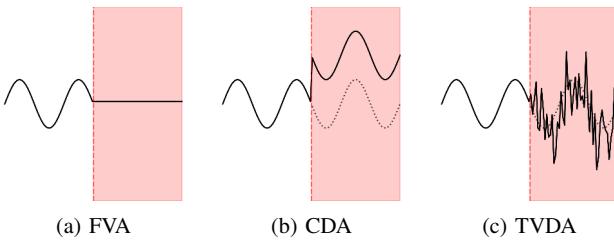


Fig. 4: Visualization of three types of pattern-wise anomaly.

III. ATSADBENCH CONSTRUCTION

A. Types of Time Series Anomalies in Aerospace

Following the behavior-driven taxonomy of [57], time series anomalies in aerospace systems are broadly classified as **point-wise** or **pattern-wise** (see Fig.3). Point-wise anomalies are isolated events, such as spikes or glitches, occurring at single time steps. In contrast, pattern-wise anomalies span contiguous intervals, characterized by sustained deviations from normal trends [34]. While point-wise anomalies are typically addressed using conventional out-of-limits (OOL) checks or basic statistical approaches [58]–[60], pattern-wise anomalies pose greater detection challenges in aerospace applications. This study thus focuses on the evaluation of LLM-based approaches for complex pattern-wise anomalies.

Within the pattern-wise category, we propose an aerospace-specific taxonomy consisting of two principal anomaly types, defined as follows (illustrated in Fig.4). Let $x(t)$ denote the expected output at time t and $y(t)$ the observed anomalous output for $t \in [t_1, t_2]$:

- **Fixed-Value Anomaly (FVA):** The output is frozen at its initial anomalous value: $y(t) = x(t_1), \forall t \in [t_1, t_2]$ (Fig. 4a).
- **Deviation Anomaly:** The output is offset from the normal trajectory: $y(t) = x(t) + \Delta$. This is subdivided as:
 - **Constant Deviation Anomaly (CDA):** $\Delta = b$, a constant offset (Fig.4b).
 - **Time-Varying Deviation Anomaly (TVDA):** $\Delta = \delta(t)$, a dynamic, time-dependent deviation (Fig. 4c).

B. Evaluation Task Definition

To comprehensively assess anomaly detection approaches in aerospace scenarios, our benchmark comprises nine tasks along two axes: signal dimensionality and anomaly propagation. The tasks are defined as follows:

TABLE I: The statistics of the constructed ATSADBENCH

Signal Dimensions	Tasks	Data Points	Anomaly Ratio
Univariate	U-FVA	8000	25%
	U-CDA	8000	25%
	U-TVDA	8000	25%
Multivariate	M-IL-FVA	8000	25%
	M-IL-CDA	8000	25%
	M-IL-TVDA	8000	25%
	M-OL-FVA	8000	25%
	M-OL-CDA	8000	25%
	M-OL-TVDA	8000	25%
Total	Training Set	72000	0%
	Test Set	36000	50%

- **Univariate (U) Tasks:** Evaluates detection based solely on temporal patterns in a single variable.
- **Multivariate (M) Tasks:** Assesses reasoning over dependencies among multiple variables, which are intrinsic to aerospace systems. To capture the diverse propagation effects of anomalies within such interconnected environments, this category is further subdivided based on the anomaly's influence on system-wide behavior:

- **In-Loop (IL) Tasks:** Anomalies affect sensors engaged in closed-loop control, with fault effects propagating system-wide (see Fig. 1).
- **Out-of-Loop (OL) Tasks:** Anomalies occur in backup or inactive sensors, remaining isolated without affecting system-level behavior.

Each of these three settings, i.e., Univariate, Multivariate In-Loop, and Multivariate Out-of-Loop, is combined with the three previously defined pattern-wise anomaly types (FVA, CDA, and TVDA), resulting in nine evaluation tasks.

C. ATSADBENCH Construction

We present ATSADBENCH, a comprehensive benchmark for time series anomaly detection in aerospace software. The construction pipeline comprises scenario design (including anomaly injection), data processing, and dataset formulation.

1) *Scenario Design and Anomaly Injection.:* The nine evaluation tasks span both univariate and multivariate domains, with the latter further divided into in-loop and out-of-loop contexts depending on sensor involvement in feedback control. For multivariate in-loop tasks, anomalies are injected into active sensors (i.e., Star Sensor 1, Gyroscope 1) whose outputs impact control actions. In out-of-loop tasks, faults affect redundant sensors with no influence on control, ensuring anomaly isolation. Each task is instantiated with two simulation runs (one each for star sensor and gyroscope faults), yielding 12 experiments in total. Each experiment lasts 2000 seconds: 1500 seconds normal operation followed by 500 seconds with injected faults. Anomalies are introduced as follows:

- **FVA:** sensor output is frozen at a constant value.
- **CDA:** a fixed offset is added to the true signal.
- **TVDA:** the output gradually diverges using a time-dependent transformation (e.g., misalignment matrix for

star sensors or drifting noise for gyroscopes).

All injected anomalies were validated by domain experts and reproduce realistic on-orbit fault behaviors.

2) *Dataset Preprocessing*: All raw telemetry data were recorded at 2 Hz (one data point every 0.5 seconds). To ensure high data quality and validity of the benchmark, two primary preprocessing steps were applied: (i) Correction of sign inversions in quaternion-based star sensor outputs, addressing inherent sign ambiguity that can cause artificial discontinuities and be misinterpreted as anomalies; (ii) Removal of the initial stabilization phase following simulation startup, thereby excluding transient behaviors unrelated to fault dynamics and focusing the analysis on steady-state operations.

3) *ATSADBENCH Formulation*: Table I summarizes the overall statistics of the constructed ATSADBENCH, which covers both univariate and multivariate anomaly detection tasks, each further subdivided by anomaly type and system context. Each data point in the dataset contains:

- **Sensor Telemetry Data**: Measurements or parameters collected from system sensors, sampled at 2 Hz.
- **Task Label**: The specific anomaly detection task, indicating both the anomaly pattern and the system configuration.
- **Ground Truth Annotation**: A binary label indicating whether the data point is normal or anomalous.

For each task, the dataset contains a fixed number of data points (8,000 per task), as detailed in Table I. This uniformity across tasks is a result of the controlled simulation settings, which generate consistent sequence lengths and facilitate systematic comparison between different scenarios. Such standardization ensures that evaluation results are directly comparable and not confounded by differences in data volume or segment length. The benchmark is divided into a training set and a test set. The training set includes only normal data, with no injected anomalies, providing a clean baseline for model training. In contrast, the test set comprises both normal and anomalous segments, with an overall anomaly ratio of 50%, enabling rigorous and reproducible assessment of detection performance under challenging conditions. Further details and access to the full dataset are provided online¹). In total, the benchmark comprises 108,000 data points.

IV. STUDY DESIGN

To systematically evaluate LLMs for time series anomaly detection in aerospace applications, we designed a study guided by four research questions, with a focus on industrial relevance and practical deployment.

A. Research Questions

RQ1: How do LLMs perform in anomaly detection on ATSADBENCH compared to SOTA unsupervised approaches? We evaluate LLMs across diverse tasks using standard metrics (F1, AUROC, AUPRC) to benchmark their anomaly detection capabilities in aerospace contexts.

¹<https://github.com/liuyang2001/ATSADBench>

RQ2: Can LLMs meet practical operational requirements for alarm accuracy, timeliness, and credibility in mission-critical aerospace scenarios? We assess whether LLMs can generate alarms that are both timely and trustworthy, as measured by user-oriented, window-level metrics essential for real-world deployment.

RQ3: How much can LLM performance be improved through enhancement strategies such as few-shot learning and RAG? We examine the value of integrating domain knowledge via few-shot prompting and reference retrieval for boosting anomaly detection effectiveness, as these simple and widely used methods provide a clear starting point for exploring enhancement strategies.

RQ4: How do LLMs perform on real-world aerospace anomaly detection tasks? We validate the practical applicability of LLM-based detection through a case study on an actual in-orbit anomaly event.

B. Evaluation Metrics

Standard Classification Metrics. To benchmark the anomaly detection capabilities of LLMs and unsupervised approaches, we adopt the F1-score, a widely used metric for time series anomaly detection [24], [61], [62]. Anomalies are detected by treating each data point as either *normal* or *abnormal*, allowing direct use of binary classification metrics. Additionally, to mitigate the impact of threshold selection inherent in PREDICTION-BASED approaches, we further use two threshold-independent metrics: the Area Under the Receiver Operating Characteristic Curve (AUROC) and the Area Under the Precision-Recall Curve (AUPRC), both calculated on the continuous prediction error [19], [63].

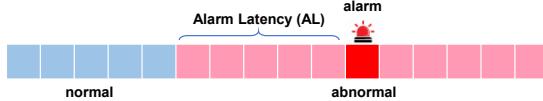
User-oriented Metrics. As discussed in § I, standard point-level metrics often fail to capture the operational needs of aerospace systems users. To address this, we aggregate model outputs at the window level. For each input window, the LLM produces binary labels for all data points; we count the number of points flagged as anomalous. If this count exceeds a set threshold, the window is labeled as anomalous, which directly reflects the user's concern: determining if a window of time contains an anomaly requiring attention. We then introduce three metrics that align with industrial requirements:

Alarm Accuracy (AA). The primary goal of this metric is to measure the correctness of window-level alarm decisions:

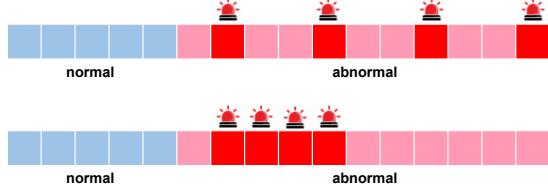
$$AA = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP, TN, FP, FN are the counts of true/false positives/negatives at the window level.

Alarm Latency (AL). We propose the AL metric to quantify the average delay, in number of windows, from the true onset of an anomaly to the first triggered alarm. As shown in Fig. 5a, each input window is visualized as a colored block: blue indicates a normal input window, pink denotes an abnormal window where the detector fails to raise an alarm, and red signifies an abnormal window where the detector successfully raises an alarm. AL quantifies the delay between the start of



(a) AL measures the delay in terms of the number of input windows from anomaly onset to the initial alarm



(b) AC measures the degree of alarm concentration, distinguishing between dispersed (top) and contiguous (bottom) alarms.

Fig. 5: Conceptual illustration of the AL and AC metrics.

an anomalous event and the first window in which an alarm is successfully triggered. It is measured by the number of windows. Lower AL indicates more timely detection.

Alarm Contiguity (AC). As illustrated in Fig. 5b, the AC metric quantifies how concentrated or continuous alarms are within an anomalous period, directly reflecting the trustworthiness and usability of the alarm system for users. In aerospace scenarios, highly contiguous alarms, those sustained throughout an anomaly, are much more actionable and reassuring than sporadic or scattered alarms. Contiguous alarms reliably signal ongoing risk, minimizing the chance that users will miss critical events, while dispersed alarms may cause confusion or erode user trust.

For each anomalous segment, defined by consecutive input windows $[s_n, e_n]$ for $n \in [1, N]$ (where N is the total number of anomalous segments), we identify L_n contiguous alarm runs, with $[\alpha_i, \beta_i]$ denoting the i -th alarm run and $i \in [1, L_n]$. The length of the longest alarm run is normalized by the segment length. The overall AC score is then given by:

$$AC = \frac{1}{N} \sum_{n=1}^N \frac{\max_{i \in [1, L_n]} (\beta_i - \alpha_i)}{e_n - s_n}$$

A higher AC indicates more sustained and reliable alarms, which are crucial for user trust and timely response in aerospace applications.

C. Evaluated Approaches

Selected Unsupervised learning Approaches. Guided by recent empirical studies and surveys [18], [22], [34], [41], [50]–[56], [64], we selected state-of-the-art unsupervised anomaly detection approaches representative of two main paradigms as mentioned in § II-A2: Sub-Adjacent [22] and TFMAE [41] for the DIRECT paradigm, and GCAD [18] for the PREDICTION-BASED paradigm.

Studied LLMs. For anomaly detection with large language models, we evaluated two open-source SOTA LLMs:

TABLE II: Window-related hyperparameters Configurations

Paradigm	Approach	Signal Dimensions	Window Size	Step Size	Prediction Horizon
Directly	Zero-shot LLMs	U	500	100	–
		M	10	10	–
	Sub-Adjacent	U&M	100	1	–
Prediction-based	TFMAE	U&M	100	1	–
		U	1000	20	20
	GCAD	U&M	30	1	1

*Note that all hyperparameters refer to the number of data points, with an interval of 0.5s between each data point.

DeepSeek-V3-0324 [45] (DeepSeek-V3) and Qwen3-235B-A22B [49] (Qwen3). Both models were obtained from the Hugging Face model hub² and run locally via API. We did not include commercial or closed-source models (such as the GPT series) in our evaluation, as they cannot be deployed for local inference. This limitation is particularly important given the data confidentiality requirements in the aerospace domain, which prohibit uploading proprietary telemetry data to third-party cloud services.

D. Experiments Setups

1) Hyperparameter Configurations.: Both unsupervised learning approaches and LLMs rely on the sliding window strategy as discussed in § II-A2 and § II-A3, making window-related hyperparameters critical to performance. For unsupervised learning approaches, we followed the recommended configurations from their original publications and reported results determined according to the best F1-score. For fair comparison with LLMs, we conducted preliminary experiments to identify optimal window and step sizes (details in the online supplementary materials¹), and adopted the configuration yielding the best results. All window settings are summarized in Table II. All LLMs used greedy decoding (temperature = 0) to ensure stable, reproducible outputs.

2) Enhancement Strategies for LLMs: To further assess LLM performance in anomaly detection, we explored two strategies: few-shot learning and RAG, both designed to better integrate domain knowledge into the detection process. The detailed prompt structures are provided in the online supplementary materials¹.

Few-shot Learning Setting. The prompt design was tailored to each detection paradigm. For the DIRECT paradigm, the prompt included one positive example (a normal temporal sequence manually selected from the training set) and three negative examples, each representing a synthetic anomaly type with accompanying explanations. For the PREDICTION-BASED paradigm, only the positive example was included in the prompt to avoid biasing the model’s predictions.

RAG Setting. In the RAG setting, reference windows were retrieved from the training set based on Euclidean

²<https://huggingface.co/models>

TABLE III: RQ1—Performance of zero-shot LLMs and SOTA unsupervised learning approaches on the ATSADBENCH. The best performance within each paradigm (DIRECT and PREDICTION-BASED) is highlighted in **bold**, and the overall best performance in each column is underlined.

Approach	Univariate				Multivariate-Out-of-Loop				Multivariate-In-Loop				Overall	
	U-FVA	U-CDA	U-TVDA	Avg.	MO-FVA	MO-CDA	MO-TVDA	Avg.	MI-FVA	MI-CDA	MI-TVDA	Avg.	Avg.	Avg.
DIRECT														
DeepSeek-V3	0.4315	0.4278	0.6199	0.4931	0.5007	0.4197	0.4355	0.4520	0.4514	0.4466	0.4912	0.4631	0.4694	
Qwen3	0.7760	0.3341	0.3450	0.4850	0.2507	0.2010	0.2847	0.2455	0.2982	0.2583	0.2597	0.2721	0.3342	
Sub-Adjacent	0.5070	0.4957	0.5050	0.5026	0.4835	0.3905	0.4690	0.4477	0.4770	0.4298	0.4681	0.4583	0.4695	
TFMAE	0.5000	0.5850	0.5000	0.5283	0.7210	0.8170	0.5145	0.6842	0.7240	0.3283	0.4760	0.5094	0.5740	
PREDICTION-BASED														
DeepSeek-V3	0.5366	0.7797	0.8192	0.7118	0.4941	0.5846	0.5956	0.5581	0.5041	0.6512	0.5491	0.5681	0.6127	
Qwen3	0.5066	0.6517	0.9722	0.7102	0.3931	0.4436	0.4226	0.4198	0.4471	0.4386	0.5246	0.4701	0.5334	
GCAD	0.0000	0.2578	0.6981	0.3187	0.9475	0.9190	0.9560	0.9408	0.8590	0.9300	0.9470	0.9120	0.7238	

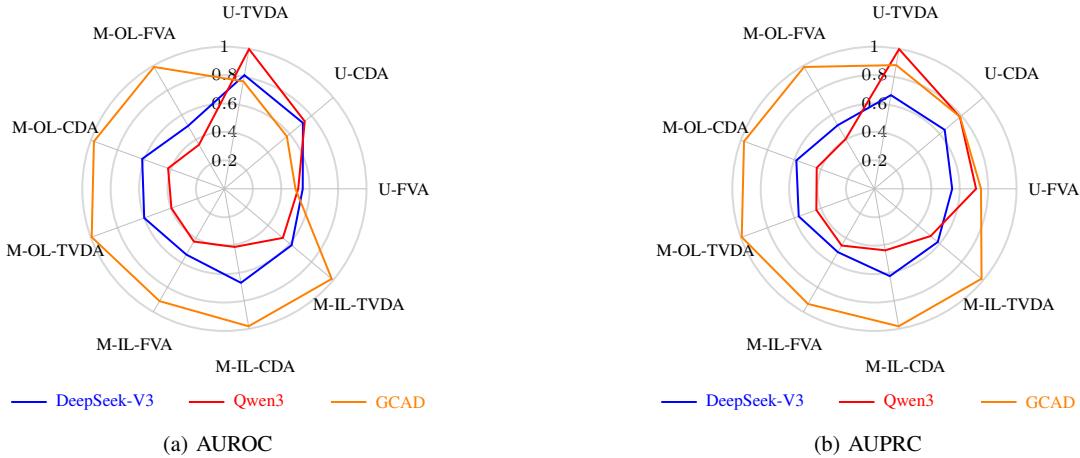


Fig. 6: RQ1—Radar plots of AUROC and AUPRC scores across nine tasks for PREDICTION-BASED approaches

distance [65]. The most similar window was incorporated into the LLM prompt as additional context.

All experiments were conducted on a workstation equipped with an Intel Xeon Platinum 8358P (128 cores), 503 GB RAM, and NVIDIA A100 (80 GB) GPUs.

V. RESULTS AND ANALYSIS

A. Evaluation of RQ1

We compare the performance of LLMs (DeepSeek-V3 and Qwen3) against SOTA unsupervised learning approaches (GCAD, TFMAE, Sub-Adjacent) across U, M-OL, and M-IL scenarios. Evaluation is conducted using F1-score, AUROC, and AUPRC as primary metrics. Table III and Fig. 6 present the experimental results on ATSADBENCH.

(1) Zero-shot LLMs vs. Unsupervised Learning Approaches: LLMs exhibit inferior average performance compared to unsupervised approaches. Under the DIRECT paradigm, the best LLM (DeepSeek-V3) achieves an average

F1-score of 0.4694, which is 10.46% lower than the top-performing unsupervised approach, TFMAE (See Table III). In the PREDICTION-BASED paradigm, LLMs achieve a maximum average F1-score of 0.6127 (DeepSeek-V3), trailing GCAD by 11.11% (See Table III). The AUROC and AUPRC scores of LLMs are also significantly lower than those of GCAD (See Fig. 6). This performance gap largely stems from the specialized domain knowledge required for aerospace TSAD, highlighting the limitations of generic, off-the-shelf LLMs relative to domain-trained models.

(2) FVA vs. CDA vs. TVDA: The performance of LLMs varies across the three anomaly types, with no single type proving consistently easier to detect. This variability is dependent on the chosen paradigm, signal dimensionality, and the specific model. For instance, in the univariate tasks, DeepSeek-V3 under the DIRECT paradigm achieves its best F1-score on TVDA (0.6199), whereas Qwen3 performs best on FVA (0.7760) (See Table III). Besides, Qwen3 performs

TABLE IV: RQ2–Performance on AA, AL, and AC. The best performance in each column is **bold**.

LLM	AA			AL			AC		
	U	M-OL	M-IL	U	M-OL	M-IL	U	M-OL	M-IL
DIRECT									
DeepSeek-V3	0.53	0.49	0.49	4.17	1.83	1.17	0.37	0.07	0.08
Qwen3	0.45	0.51	0.48	2.33	1.83	1.33	0.22	0.09	0.07
PREDICTION-BASED									
DeepSeek-V3	0.71	0.56	0.57	1.17	2.50	1.83	0.55	0.05	0.05
Qwen3	0.70	0.41	0.47	1.17	1.17	1.33	0.68	0.04	0.07

best on FVA in the DIRECT paradigm but on TVDA in the PREDICTION-BASED paradigm (See Table III). Therefore, we cannot draw a general conclusion about which anomaly type is inherently more or less challenging for LLMs, as the detection performance is highly context-dependent.

(3) Univariate vs. Multivariate (In-Loop and Out-of-Loop): LLMs perform best in univariate tasks, sometimes even surpassing SOTA unsupervised approaches, while their effectiveness declines as task complexity increases. In M tasks, their overall performance on M-IL tasks is slightly higher than on M-OL tasks. For univariate tasks, Qwen3 under the PREDICTION-BASED paradigm achieves the highest average F1-score (0.7118), outperforming M-IL and M-OL settings by 14.37% and 15.37%, respectively (See Table III). The AUROC and AUPRC scores show similar trends (See Fig. 6). By analyzing the outputs, we found that the performance drop mainly comes from mistaking normal fluctuations in other variables for anomalies, which weakens the visibility of true anomalies. The overall performance in M-IL tasks is better than in M-OL tasks (by 1–5% in F1-score (See Table III) and by 2–7% in both AUROC and AUPRC scores (See Fig. 6)). This may be attributed to in-loop propagation generating larger and more pronounced anomalous patterns in the data, thereby facilitating improved detection performance by the LLMs.

(4) DIRECT vs. PREDICTION-BASED Paradigm: LLMs under the PREDICTION-BASED paradigm achieve F1-scores 15–19% higher than under the DIRECT paradigm (see Table III). We attribute this to the inherent auto-regressive strengths of LLMs, which are better exploited in the PREDICTION-BASED setting, while the DIRECT paradigm requires more abstract, zero-shot classification, which is a more challenging task.

Answer to RQ1: LLMs perform notably well in U tasks, though they still lag behind SOTA unsupervised approaches. In M tasks, they show a slight advantage in M-IL over M-OL. The effectiveness of LLMs varies across anomaly types. Among the two paradigms, LLMs achieve better results under the PREDICTION-BASED paradigm.

B. Evaluation of RQ2

We evaluate the practical performance of the SOTA LLMs, DeepSeek-V3 and Qwen3, on three metrics: AA, AL, and AC, as described in § IV-B. Results are shown in Table IV, where higher AA/AC and lower AL indicate better performance.

(1) Performance in AA: LLMs perform optimally only in univariate tasks under the PREDICTION-BASED paradigm, achieving an accuracy of nearly 0.7. Conversely, in all other cases, their accuracy drops to a range between 0.41 and 0.57, which is almost indistinguishable from random guessing. This may be due to the inherent limitations of off-the-shelf LLMs in processing inter-variate relationships and their lack of domain-specific knowledge in aerospace.

(2) Performance in AL: Qwen3 under the PREDICTION-BASED paradigm exhibited the best overall alarm latency, raising an alarm within an average of 1.22 anomalous windows. In other cases, however, significant delays can occur, with alarms sometimes not being issued until the fourth window, which could result in immeasurable damage in aerospace software.

(3) Performance in AC: LLMs achieve significantly higher alarm contiguity in univariate tasks compared to multivariate tasks. In the univariate tasks, the best AC score reaches up to 0.68; however, in the multivariate tasks, AC drops sharply, with the highest score being only 0.09. This further highlights the limited ability of LLMs to analyze relationships among multiple variables.

Answer to RQ2: LLMs achieve higher AA and AC only in univariate tasks, whereas their performance in multivariate tasks is nearly equivalent to random guessing (AA close to 0.5, AC averaging 0.06). AL does not follow this trend; the best result is achieved by Qwen3 under the PREDICTION-BASED paradigm, with an average of 1.22 windows.

C. Evaluation of RQ3

We evaluate the performance of LLMs on ATSADBENCH using few-shot learning and retrieved normal temporal sequences. We use both standard classification metrics and practical metrics described in § IV-B. Table V presents the results. Green text indicates a performance improvement (higher F1/AA/AC, lower AL), and red indicates a degradation.

(1) Performance of Few-Shot Learning: Few-shot learning greatly enhances the performance of LLMs under the Direct paradigm, but shows only slight improvements or even substantial declines under the Prediction-Based paradigm. Under the Direct paradigm, Few-shot learning significantly improves DeepSeek-V3 by 11.25% in F1-score and 40.17% in AC, and Qwen3 by 28.23% in F1-score and 60.22% in AC. In contrast, the Prediction-Based paradigm shows only modest improvements (0.4% to 4%) for Qwen3, with increased AL, while DeepSeek-V3 sees nearly 10% declines in F1-score and AA. This may be due to the absence of negative data points in the Prediction-Based paradigm.

(2) Performance of RAG: LLMs with RAG do not show significant improvements. For the three metrics (F1-score, AA,

TABLE V: RQ3–Performance comparison of enhancement strategies of LLMs.

Strategy	F1-Score	AA	AL	AC
<i>DeepSeek-V3 with DIRECT paradigm</i>				
ZeroShot	0.4694	0.5024	2.4	0.1706
FewShot	+11.25%	-1.49%	-0.5	+40.17%
RAG	-1.65%	+0.60%	-0.2	-2.00%
<i>Qwen3 with DIRECT paradigm</i>				
ZeroShot	0.3342	0.4804	1.8	0.1244
FewShot	+28.23%	-0.17%	-0.2	+60.22%
RAG	+15.52%	-0.60%	0.0	-2.17%
<i>DeepSeek-V3 with PREDICTION-BASED paradigm</i>				
ZeroShot	0.6127	0.6131	1.8	0.2178
FewShot	-9.40%	-8.93%	+0.4	-1.47%
RAG	-4.80%	-4.50%	+0.4	-1.08%
<i>Qwen3 with PREDICTION-BASED paradigm</i>				
ZeroShot	0.5334	0.5254	1.2	0.2617
FewShot	+3.71%	+3.81%	+0.7	+1.28%
RAG	+2.27%	+0.47%	+0.1	+3.36%

and AC), changes remain within the -5% to +4% range, except for Qwen3 with the DIRECT paradigm, which shows a notable +15.52% improvement in F1-score. And the changes of AL fluctuate within -0.2 to +0.4. This may be due to the inability of LLMs to extract domain-specific knowledge from retrieved normal temporal sequences.

Answer to RQ3: Few-shot learning enhances the performance of LLMs under the DIRECT paradigm but exhibits limited capacity under the Prediction-Based paradigm. RAG, on the other hand, does not demonstrate significant improvements across the tasks.

D. Evaluation of RQ4

To validate the practical applicability of LLMs, we conducted a case study on a historical in-orbit anomaly event. Based on our findings in § V-C, we selected the Qwen3 model under the PREDICTION-BASED paradigm and examined its performance under zero-shot, few-shot, and RAG settings.

Case Description. We utilized 45-dimensional telemetry data from a significant satellite anomaly on July 12, 2022. The event was initiated by a gyroscope sensor fault that, due to being in-loop, propagated through the attitude control system, ultimately causing structural damage. For our test, we extracted a 2000-second data segment, where the anomaly onset occurs at the 1000-second mark, as illustrated in Fig. 7.

Results and Analysis. The results for the zero-shot, few-shot, and RAG settings are presented in Fig 7a, 7b and 7c. In each subfigure, the green line represents the original gyroscope

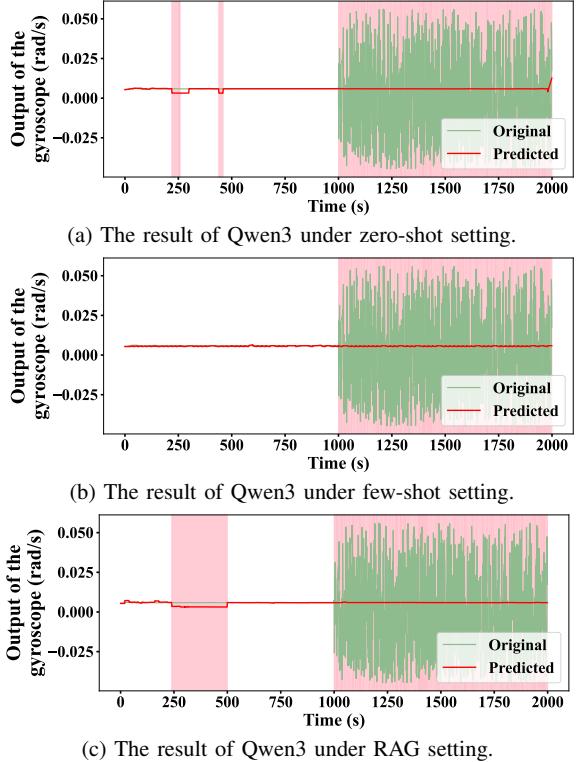


Fig. 7: RQ4—Case study results of Qwen3 on the gyroscope anomaly under different settings.

output, the red line shows the Qwen3 prediction, and the pink shaded areas indicate data points that the model flagged as anomalous. A key observation across all three settings is that they successfully generate timely and continuous alarms for the actual anomaly that begins at the 1000s mark. In the zero-shot setting (Fig. 7a), Qwen3 also produces false alarms in the intervals [220s, 257s] and [440s, 459s]. By contrast, the few-shot setting (Fig. 7b) eliminates these false alarms and achieves an F1-score of 0.9379, AA of 0.95, AL of 1, and AC of 0.52. The RAG setting (Fig. 7c), however, introduces a longer false alarm period from 240s to 499s.

Answer to RQ4: In the zero-shot setting, Qwen3 successfully detects the true anomaly onset but produces false alarms prior to the event. Few-shot learning eliminates these false alarms, while RAG leads to more false alarms.

E. Threats to Validity

The threats to internal validity lie in the construction of dataset, as ATSADBENCH is synthetically generated and may not fully capture the complexity of real aerospace anomalies. To address this, we employed high-fidelity simulation, incorporated a wide range of anomaly types and scenarios, and applied rigorous preprocessing to enhance data quality. For unsupervised learning approaches, we adopted recommended hyperparameter configurations and reported the best results.

The threats to external validity are the generalizability of our findings beyond the constructed benchmark and selected models. Our evaluation is limited to the benchmark we constructed and a subset of open-source LLMs, and results may not generalize to other architectures, proprietary models or datasets. To encourage broader validation, we have released our benchmark and code¹, and invite future studies to evaluate additional models and more diverse real-world datasets.

VI. DISCUSSION

A. Implications on Practice

The results from this study offer meaningful insights into the practical deployment of LLM-based anomaly detection within the aerospace sector, for both ground and onboard telemetry systems. Aerospace users can leverage the generalizable nature of LLMs to implement flexible, training-free anomaly detectors, enabling rapid deployment and seamless adaptation to evolving operational telemetry streams.

B. Limitations

We acknowledge several limitations. First, our benchmark is built from simulation data, and the injected anomalies may not fully represent the complexity and diversity of faults encountered in real-world aerospace operations. To enhance authenticity, we plan to develop future benchmarks based on actual satellite telemetry. Second, this work focuses exclusively on the aerospace domain. Applying our evaluation framework to other fields would help assess the generalizability and comparative performance of LLMs for TSAD across different domains. Third, the scope of model selection and enhancement strategies in this work remains limited. We only experimented with a small set of open-source models (DeepSeek and Qwen) and employed relatively simple strategies such as few-shot learning and RAG. Future work will explore a broader range of foundation models, including LLaMA and Mistral, and incorporate more advanced techniques such as reinforcement learning from human feedback (RLHF) and advanced contextual prompts [66] to further improve detection performance.

C. Future Directions

Enhancing Inter-Variable Reasoning in LLMs. A major challenge is that the natural fluctuations of other normal variables introduce noise, leading LLMs to misinterpret them as anomalies. Our results show that, without domain adaptation, LLM performance drops sharply on multivariate data as additional sensor streams introduce noise and obscure true anomalies. Future work should develop structured prompting techniques that explicitly encode inter-variable relationships such as causal links or physical constraints [67]–[70], so the model can better distinguish meaningful patterns from spurious correlations in complex telemetry.

Developing Practical Metrics. Our work underscores that traditional metrics like F1 score are insufficient for assessing an anomaly detector’s utility in safety-critical aerospace contexts. In such settings, when and how alarms are raised can be as important as overall accuracy. Industrial deployment

demands a comprehensive evaluation framework that accounts for alarm timeliness, credibility, false-alarm rates, and operational constraints like model size and inference latency. Future work should focus on designing and standardizing domain-specific metrics that capture these practical requirements and trade-offs [54], supporting more informed model selection and deployment in industrial settings.

Integrating Domain Knowledge into LLMs. Our results indicate that prompt-based enhancement strategies alone are insufficient for LLMs to effectively handle complex aerospace telemetry, particularly in PREDICTION-BASED anomaly detection. Achieving robust domain adaptation requires more fundamental approaches, such as domain-specific pretraining and fine-tuning [71]–[73]. By training LLMs on large corpora of in-domain time-series data and embedding relevant engineering knowledge, these approaches enable models to internalize essential physical relationships and operational patterns. This integration, which goes beyond zero-shot learning, is essential for narrowing the performance gap with specialized unsupervised learning approaches and enhancing detection accuracy and credibility in real-world aerospace applications.

Generality to Other Domains. Current TSAD research predominantly evaluates approaches on generic datasets spanning multiple domains, which may mask domain-specific challenges and limit the practical relevance of reported results. In contrast, our study demonstrates the benefits of a focused, in-depth evaluation in aerospace, leveraging domain-specific data and requirements. We encourage researchers in other critical application areas to conduct similarly targeted investigations, developing dedicated benchmarks, capturing the distinct characteristics of their domains, and designing practical evaluation metrics that align with real-world user needs.

VII. CONCLUSION

In this paper, we conducted the first systematic evaluation of LLMs for TSAD in the aerospace domain. To support this study, we introduce ATSADBENCH, a benchmark constructed from realistic aerospace scenarios with 108,000 data points, and propose three user-oriented, window-based metrics: Alarm Accuracy, Alarm Latency, and Alarm Contiguity. These metrics are designed to provide a practical and comprehensive assessment of model performance in operational contexts.

Our experimental results show that LLMs achieve promising results in univariate TSAD tasks but encounter significant difficulties with multivariate telemetry, where their performance is often close to random guessing. Additionally, our study finds that enhancement strategies offer only limited benefits; few-shot learning brings moderate improvements under the DIRECT paradigm, while RAG does not lead to substantial gains. We hope that the introduction of ATSADBENCH and user-oriented metrics will inspire further research into LLMs for TSAD and support the development of more effective approaches for aerospace and other safety-critical domains.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (Grant Nos. 62402038, U21B2015, 62192730 and 62192735).

REFERENCES

- [1] J. Jang and H. Kwon, "TAIL-MIL: time-aware and instance-learnable multiple instance learning for multivariate time series anomaly detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 2025, pp. 17582–17589.
- [2] F. Liu, X. Zhou, J. Cao, Z. Wang, T. Wang, H. Wang, and Y. Zhang, "Anomaly detection in quasi-periodic time series based on automatic data segmentation and attentional LSTM-CNN (extended abstract)," in *Proceedings of the 39th IEEE International Conference on Data Engineering, ICDE*. IEEE, 2023, pp. 3777–3778.
- [3] Z. Zimmerman, K. Kamgar, N. S. Senobari, B. Crites, G. J. Funning, P. Brisk, and E. J. Keogh, "Matrix profile XIV: scaling time series motif discovery with gpus to break a quintillion pairwise comparisons a day and beyond," in *Proceedings of the ACM Symposium on Cloud Computing (SoCC)*. ACM, 2019, pp. 74–86.
- [4] G. Pang, C. Shen, and A. van den Hengel, "Deep anomaly detection with deviation networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. ACM, 2019, pp. 353–362.
- [5] J. Qiu, Q. Du, and C. Qian, "KPI-TSAD: A time-series anomaly detector for KPI monitoring in cloud applications," *Symmetry*, vol. 11, no. 11, p. 1350, 2019.
- [6] M. E. Villa-Pérez, M. Á. Álvarez-Carmona, O. Loyola-González, M. A. Medina-Pérez, J. C. Velasco-Rossell, and K. R. Choo, "Semi-supervised anomaly detection algorithms: A comparative summary and future research directions," *Knowl. Based Syst.*, vol. 218, p. 106878, 2021.
- [7] Z. Chen, J. Liu, Y. Su, H. Zhang, X. Ling, Y. Yang, and M. R. Lyu, "Adaptive performance anomaly detection for online service systems via pattern sketching," in *Proceedings of the 44th international conference on software engineering*, 2022, pp. 61–72.
- [8] Z. He, P. Chen, and T. Huang, "Share or not share? towards the practicability of deep models for unsupervised anomaly detection in modern online systems," in *Proc. IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2022, pp. 25–35.
- [9] X. Zhang, Y. Zhao, C. Liu, L. Wang, X. Yang, Y. Hou, Z. Lan, X. Hu, B. Miao, M. Yang *et al.*, "fkpiselect: Fault-injection based automated kpi selection for practical multivariate anomaly detection," in *IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*, 2023, pp. 183–194.
- [10] "Autokad: Empowering KPI anomaly detection with label-free deployment," in *IEEE International Symposium on Software Reliability Engineering, ISSRE*. IEEE, 2023, pp. 13–23.
- [11] Y. Jing, J. Wang, J. Qi, Q. Qi, B. He, Z. Zhuang, N. Wu, and J. Liao, "Diner: Interpretable anomaly detection for seasonal time series in web services," *IEEE Transactions on Services Computing*, 2024.
- [12] H. Liu, X. Huang, M. Jia, T. Jia, J. Han, Y. Li, and Z. Wu, "Uac-ad: Unsupervised adversarial contrastive learning for anomaly detection on multi-modal data in microservice systems," *IEEE Transactions on Services Computing*, 2024.
- [13] Q. Tang, C. Dai, Y. Wu, and H. Zhou, "Mlp-mixer based masked autoencoders are effective, explainable and robust for time series anomaly detection," *Proceedings of the VLDB Endowment*, vol. 18, no. 3, pp. 798–811, 2024.
- [14] Z. Wang, C. Pei, M. Ma, X. Wang, Z. Li, D. Pei, S. Rajmohan, D. Zhang, Q. Lin, H. Zhang *et al.*, "Revisiting vae for unsupervised time series anomaly detection: A frequency perspective," in *Proceedings of the ACM Web Conference*, 2024, pp. 3096–3105.
- [15] C. Lee, T. Yang, Z. Chen, Y. Su, and M. R. Lyu, "Maat: Performance metric anomaly anticipation for cloud services with conditional diffusion," in *Proceedings of the 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2023, pp. 116–128.
- [16] J. Liu, T. Yang, Z. Chen, Y. Su, C. Feng, Z. Yang, and M. R. Lyu, "Practical anomaly detection over multivariate monitoring metrics for online services," in *IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2023, pp. 36–45.
- [17] D. Kim, S. Park, and J. Choo, "When model meets new normals: test-time adaptation for unsupervised time-series anomaly detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 12, 2024, pp. 13113–13121.
- [18] Z. Liu, M. Gao, and P. Jiao, "GCAD: anomaly detection in multivariate time series from the perspective of granger causality," in *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, 2025, pp. 19041–19049.
- [19] X. Huang, W. Chen, B. Hu, and Z. Mao, "Graph mixture of experts and memory-augmented routers for multivariate time series anomaly detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 16, 2025, pp. 17476–17484.
- [20] W. Chen, L. Tian, B. Chen, L. Dai, Z. Duan, and M. Zhou, "Deep variational graph convolutional recurrent network for multivariate time series anomaly detection," in *International conference on machine learning*. PMLR, 2022, pp. 3621–3633.
- [21] W. Zhang, C. Zhang, and F. Tsung, "GRELEN: multivariate time series anomaly detection from the perspective of graph relational learning," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*. ijcai.org, 2022, pp. 2390–2397.
- [22] W. Yue, X. Ying, R. Guo, D. Chen, J. Shi, B. Xing, Y. Zhu, and T. Chen, "Sub-adjacent transformer: Improving time series anomaly detection with reconstruction error from sub-adjacent neighborhoods," *CoRR*, vol. abs/2404.18948, 2024.
- [23] C. Liu, S. He, Q. Zhou, S. Li, and W. Meng, "Large language model guided knowledge distillation for time series anomaly detection," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI*. ijcai.org, 2024, pp. 2162–2170.
- [24] C. Wang, Z. Zhuang, Q. Qi, J. Wang, X. Wang, H. Sun, and J. Liao, "Drift doesn't matter: Dynamic decomposition with diffusion reconstruction for unstable multivariate time series anomaly detection," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
- [25] J. Song, K. Kim, J. Oh, and S. Cho, "MEMTO: memory-guided transformer for multivariate time series anomaly detection," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [26] C. Lai, F. Sun, Z. Gao, J. H. Lang, and D. S. Boning, "Nominality score conditioned time series anomaly detection by point/sequential reconstruction," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2023.
- [27] L. Li, J. Yan, Q. Wen, Y. Jin, and X. Yang, "Learning robust deep state space for unsupervised anomaly detection in contaminated time-series," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6058–6072, 2023.
- [28] S. Liu, B. Zhou, Q. Ding, B. Hooi, Z. Zhang, H. Shen, and X. Cheng, "Time series anomaly detection with adversarial reconstruction networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 4293–4306, 2023.
- [29] Y. Zhang, Y. Chen, J. Wang, and Z. Pan, "Unsupervised deep anomaly detection for multi-sensor time-series signals," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 2118–2123, 2023.
- [30] B. Zheng, L. Ming, K. Zeng, M. Zhou, X. Zhang, T. Ye, B. Yang, X. Zhou, and C. S. Jensen, "Adversarial graph neural network for multivariate time series anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 12, pp. 7612–7626, 2024.
- [31] H. Xu, Y. Wang, S. Jian, Q. Liao, Y. Wang, and G. Pang, "Calibrated one-class classification for unsupervised time series anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 5723–5736, 2024.
- [32] H. Kim, S. Kim, S. Min, and B. Lee, "Contrastive time-series anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 10, pp. 5053–5065, 2024.
- [33] X. Zhang, S. Xu, H. Chen, Z. Chen, F. Zhuang, H. Xiong, and D. Yu, "Rethinking robust multivariate time series anomaly detection: A hierarchical spatio-temporal variational perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 12, pp. 9136–9149, 2024.
- [34] H. Si, J. Li, C. Pei, H. Cui, J. Yang, Y. Sun, S. Zhang, J. Li, H. Zhang, J. Han *et al.*, "Timeseriesbench: An industrial-grade benchmark for time series anomaly detection models," in *IEEE 35th International Symposium on Software Reliability Engineering (ISSRE)*, 2024, pp. 61–72.
- [35] S. Gao, T. Koker, O. Queen, T. Hartvigsen, T. Tsiligkaridis, and M. Zitnik, "Units: A unified multi-task time series model," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

- [36] T. Zhou, P. Niu, L. Sun, R. Jin *et al.*, “One fits all: Power general time series analysis by pretrained lm,” *Advances in neural information processing systems*, vol. 36, pp. 43 322–43 355, 2023.
- [37] S. Alnegheimish, L. Nguyen, L. Berti-Équille, and K. Veeramachaneni, “Large language models can be zero-shot anomaly detectors for time series?” *CoRR*, vol. abs/2405.14755, 2024.
- [38] J. Liu, C. Zhang, J. Qian, M. Ma, S. Qin, C. Bansal, Q. Lin, S. Rajmohan, and D. Zhang, “Large language models can deliver accurate and interpretable time series anomaly detection,” *CoRR*, vol. abs/2405.15370, 2024.
- [39] Y. Li, Y. Lu, J. Wang, Q. Qi, J. Wang, Y. Wang, and J. Liao, “Tadl: Fault localization with transformer-based anomaly detection for dynamic microservice systems,” in *IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2023, pp. 718–722.
- [40] X.-Y. Li, P.-N. Zhong, D. Chen, Z.-D. Zhang, and Y.-B. Yang, “Deepdiscord: Dual contrastive coding for transferable time series anomaly detection,” in *ECAI 2023*. IOS Press, 2023, pp. 1439–1446.
- [41] Y. Fang, J. Xie, Y. Zhao, L. Chen, Y. Gao, and K. Zheng, “Temporal-frequency masked autoencoders for time series anomaly detection,” in *IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 1228–1241.
- [42] M. Zhang, X. Shi, J. Huang, L. Su, and Y. Zhang, “Dyntackr: A robust two-stage framework with attribute enhancement for kpi anomaly detection,” in *IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 2024, pp. 169–176.
- [43] H. Si, C. Pei, Z. Li, Y. Zhao, J. Li, H. Zhang, Z. Diao, J. Li, G. Xie, and D. Pei, “Beyond sharing: Conflict-aware multivariate time series anomaly detection,” in *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2023, pp. 1635–1645.
- [44] K. K. Chang, M. Cramer, S. Soni, and D. Bamman, “Speak, memory: An archaeology of books known to chatgpt/gpt-4,” *CoRR*, vol. abs/2305.00118, 2023.
- [45] “Deepseek-v3 technical report,” *CoRR*, vol. abs/2412.19437, 2024.
- [46] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, “Large language models are zero-shot time series forecasters,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 19 622–19 635, 2023.
- [47] T. Liu and B. K. H. Low, “Goat: Fine-tuned llama outperforms GPT-4 on arithmetic tasks,” *CoRR*, vol. abs/2305.14201, 2023.
- [48] “Llama 2: Open foundation and fine-tuned chat models,” *CoRR*, vol. abs/2307.09288, 2023.
- [49] “Qwen3 technical report,” *CoRR*, vol. abs/2505.09388, 2025.
- [50] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, “Do deep neural networks contribute to multivariate time series anomaly detection?” *Pattern Recognition*, vol. 132, p. 108945, 2022.
- [51] D. Li, S. Zhang, Y. Sun, Y. Guo, Z. Che, S. Chen, Z. Zhong, M. Liang, M. Shao, M. Li *et al.*, “An empirical analysis of anomaly detection methods for multivariate time series,” in *IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2023, pp. 57–68.
- [52] M. S. Sarfraz, M.-Y. Chen, L. Layer, K. Peng, and M. Koulakis, “Position: quo vadis, unsupervised time series anomaly detection?” *arXiv preprint arXiv:2405.02678*, 2024.
- [53] C. Freeman, J. Merriman, I. Beaver, and A. Mueen, “Experimental comparison and survey of twelve time series anomaly detection algorithms (extended abstract),” in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, L. D. Raedt, Ed. ijcai.org, 2022, pp. 5737–5741.
- [54] T. Puccetti and A. Ceccarelli, “Detection latencies of anomaly detectors—an overlooked perspective?” in *IEEE 35th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2024, pp. 37–48.
- [55] S. Kim, K. Choi, H.-S. Choi, B. Lee, and S. Yoon, “Towards a rigorous evaluation of time-series anomaly detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022, pp. 7194–7201.
- [56] R. Wu and E. J. Keogh, “Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress,” *IEEE transactions on knowledge and data engineering*, vol. 35, no. 3, pp. 2421–2429, 2021.
- [57] K.-H. Lai, D. Zha, J. Xu, Y. Zhao, G. Wang, and X. Hu, “Revisiting time series outlier detection: Definitions and benchmarks,” in *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 1)*, 2021.
- [58] Q. Li, X. Zhou, P. Lin, and S. Li, “Anomaly detection and fault diagnosis technology of spacecraft based on telemetry-mining,” in *3rd International Symposium on Systems and Control in Aeronautics and Astronautics*, 2010, pp. 233–236.
- [59] J. M. Heras and A. Donati, “Enhanced telemetry monitoring with novelty detection,” *AI Magazine*, vol. 35, no. 4, pp. 37–46, 2014.
- [60] T. Yairi, Y. Kawahara, R. Fujimaki, Y. Sato, and K. Machida, “Telemetry-mining: a machine learning approach to anomaly detection and fault diagnosis for space systems,” in *Proceedings of the 2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT’06)*, 2006, pp. 8 pp.–476.
- [61] X. Li, P. Zhong, D. Chen, Z. Zhang, and Y. Yang, “Deepdiscord: Dual contrastive coding for transferable time series anomaly detection,” ser. *Frontiers in Artificial Intelligence and Applications*, vol. 372. IOS Press, 2023, pp. 1439–1446.
- [62] “Multivariate time series anomaly detection based on pre-trained models with dual-attention mechanism,” in *IEEE International Symposium on Software Reliability Engineering, ISSRE*. IEEE, 2024, pp. 73–78.
- [63] D. Kim, S. Park, and J. Choo, “When model meets new normals: test-time adaptation for unsupervised time-series anomaly detection.” AAAI Press, 2024.
- [64] M. Jin, H. Y. Koh, Q. Wen, D. Zambon, C. Alippi, G. I. Webb, I. King, and S. Pan, “A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [65] S. Yang, D. Wang, H. Zheng, and R. Jin, “Timerag: Boosting LLM time series forecasting via retrieval-augmented generation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE, 2025, pp. 1–5.
- [66] C. Li, S. Zhou, J. Kong, L. Qi, and H. Xue, “Kanoclip: Zero-shot anomaly detection through knowledge-driven prompt learning and enhanced cross-modal integration,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.
- [67] P. Shojaee, K. Meidani, S. Gupta, A. B. Farimani, and C. K. Reddy, “LLM-SR: scientific equation discovery via programming with large language models,” in *The Thirteenth International Conference on Learning Representations, (ICLR)*. OpenReview.net, 2025.
- [68] E. Meyerson, M. J. Nelson, H. Bradley, A. Gaier, A. M. Karkaj, A. K. Hoover, and J. Lehman, “Language model crossover: Variation through few-shot prompting,” *ACM Trans. Evol. Optim.*, vol. 4, no. 4, pp. 27:1–27:40, 2024.
- [69] B. Romera-Paredes, M. Barekatain, A. Novikov, M. Balog, M. P. Kumar, E. Dupont, F. J. R. Ruiz, J. S. Ellenberg, P. Wang, O. Fawzi, P. Kohli, and A. Fawzi, “Mathematical discoveries from program search with large language models,” *Nat.*, vol. 625, no. 7995, pp. 468–475, 2024.
- [70] R. Wang, B. Wang, K. Li, Y. Zhang, and J. Cheng, “Drsr: Llm based scientific equation discovery with dual reasoning from data and experience,” *arXiv preprint arXiv:2506.04282*, 2025.
- [71] C. Jeong, “Fine-tuning and utilization methods of domain-specific llms,” *CoRR*, vol. abs/2401.02981, 2024.
- [72] J. Zheng, H. Hong, X. Wang, J. Su, Y. Liang, and S. Wu, “Fine-tuning large language models for domain-specific machine translation,” *CoRR*, vol. abs/2402.15061, 2024.
- [73] J. Shen, N. A. Tenenholtz, J. B. Hall, D. Alvarez-Melis, and N. Fusi, “Tag-llm: Repurposing general-purpose llms for specialized domains,” in *Forty-first International Conference on Machine Learning (ICML)*. OpenReview.net, 2024.