A Hybrid Framework for Time Series Anomaly Detection with LLMs

Hakan Akdag
German Aerospace Center (DLR)
Cologne University of Applied Sciences
Cologne, Germany
hakan.akdag@dlr.de

Anh Thu Bui*
Cologne University of Applied Sciences
Cologne, Germany
anhthu.m.bui@gmail.com

Natalie Hußfeldt*
Cologne University of Applied Sciences
Cologne, Germany
n.hussfeldt@gmail.com

Gernot Heisenberg
Cologne University of Applied Sciences
Cologne, Germany
gernot.heisenberg@th-koeln.de

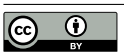
Abstract

Large Language Models (LLMs) are increasingly explored for time series anomaly detection (TSAD), yet their direct use often remains inferior to methods designed specifically for TSAD. In this work, we study whether this limitation arises primarily from weak forecasting or from insufficient exploitation of forecast residuals. Across LLaMA-3.1-8B-Instruct, Mistral-7B, and DeepSeek-R1-Distill-Qwen-7B, we find that parameter-efficient fine-tuning can improve forecasting accuracy, but does not consistently improve anomaly detection under fixed residual thresholding. The primary driver of detection performance is not forecasting quality, but how the resulting residuals are analyzed and refined in subsequent stages. Motivated by this finding, we propose a Forecaster–Detector–Imputer pipeline that restricts the LLM to forecasting normal temporal behavior, applies classical detectors to the resulting residuals using adaptive windowing and contamination-aware thresholding, and refines candidate detections through imputation via a time-series foundation model followed by a second detection pass. The proposed framework reaches F1 = 0.789 with LLaMA and F1 = 0.785 with Mistral, roughly doubling the zero-shot baselines, with results confirmed on an independent NAB benchmark. These results suggest that LLMs are most effective for TSAD when used as forecasting components within a hybrid residual-analysis pipeline, and that the primary bottleneck lies in post-hoc residual exploitation rather than forecasting quality alone.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**; **Temporal reasoning**; • **Information systems** → *Data mining*.

*Both authors contributed equally to this work.



This work is licensed under a Creative Commons Attribution 4.0 International License.
CIKM '26, Rome, Italy
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2539-5/2026/11
https://doi.org/10.1145/3799682.3840786

Keywords

Time Series Anomaly Detection; Large Language Models; Hybrid Models; Residual-Based Detection; Parameter-Efficient Fine-Tuning; Imputation; Few-Shot Forecasting

ACM Reference Format:
Hakan Akdag, Anh Thu Bui, Natalie Hußfeldt, and Gernot Heisenberg. 2026. A Hybrid Framework for Time Series Anomaly Detection with LLMs. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 07–11, 2026, Rome, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3799682.3840786

1 Introduction

Time Series Anomaly Detection (TSAD) is important in many data-driven systems, including industrial fault detection, fraud prevention, healthcare monitoring, cybersecurity and remote sensing [20, 23, 29, 34]. Its central challenge is to detect deviations from expected temporal behavior automatically, at scale, and with little labeled data. In practice, this remains difficult: labeled anomalies are scarce, class imbalance is often severe, and models that perform well in one setting may fail once operating conditions change [19, 25]. Recent benchmark studies also show that detection performance depends strongly on the anomaly type and that more complex models do not consistently outperform simpler baselines [25, 29].

Large Language Models (LLMs) have recently opened up a different way of thinking about TSAD. Although they are not trained on anomaly detection data and lack explicit numerical grounding, pretraining still gives them a strong prior over structured and compressible sequences [23], what may be referred to as a *simplicity bias* [11]. Many time series share this property through seasonality, trends, and recurring local patterns, which makes LLMs plausible zero-shot forecasters of normal temporal behavior. Approaches such as SigLLM [1] build on this idea by forecasting the next values of a series and deriving anomaly scores from the resulting residuals. This is a more natural use of LLMs than asking them to detect anomalies directly from prompts, where hallucinations and inconsistent judgments remain a persistent problem. Still, the overall detection performance of LLM-based TSAD pipelines can often not compete with expert models specialized for TSAD, cf. for instance Refs. [29, 31, 32].

To address this limitation, in this work we do not consider the LLM as a standalone anomaly detector. Instead, we use it only to

model normal temporal dynamics and delegate anomaly scoring to a downstream residual-analysis module. This design choice is motivated by our empirical observation that, under the zero-shot SigLLM baseline, range-aware scores such as VUS-ROC remain relatively strong even when point-wise F_1 is low, indicating that anomaly information is already present in the residuals but is not extracted effectively by the final scoring step. We therefore propose a Forecaster–Detector–Imputer (FDI) pipeline that separates sequence modeling, residual-based anomaly scoring, and residual stabilization through imputation. On 13 labeled real-world TAMA segments [36] spanning five structural anomaly categories, FDI reaches $F_1 = 0.789$ with LLaMA and $F_1 = 0.785$ with Mistral using only open-source models in the 7–8B range, substantially improving the respective SigLLM baseline values. These findings suggest that, in this benchmark setting, LLMs are more effective as forecasting components in a hybrid TSAD pipeline than as standalone detectors.

The remainder of this work is structured as follows: Sections 2 and 3 briefly review the state of the art in TSAD and recent LLM-based approaches for TSAD, and position our method within this research landscape. Subsequently, we introduce a hybrid framework based on a Forecaster–Detector–Imputer pipeline in Sect. 4. The experimental setup, datasets, and evaluation metrics are described in Sect. 5. Finally, we present our results in Sect. 6, and conclude in Sect. 7 with a discussion of findings and future directions.

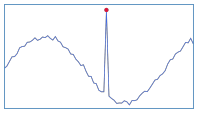
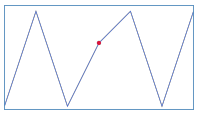
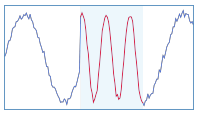
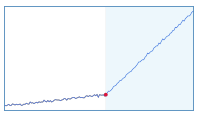
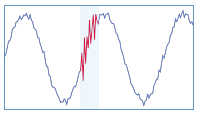
2 Time Series Anomaly Detection

Time series anomaly detection (TSAD) studies how to identify observations or subsequences that deviate from the expected temporal behavior of a signal. Unlike standard forecasting, the goal is not only to predict future values accurately, but also to distinguish normal variation from behavior that is unlikely under the underlying data-generating process. This is difficult in practice because anomalies are rare, heterogeneous, and often defined by context rather than by absolute magnitude alone [19, 25, 29].

Within TSAD one may differentiate between *point-wise* and *range-wise* anomalies. Point anomalies affect isolated observations in time, whereas range anomalies extend over contiguous subsequences and often alter local temporal structure rather than producing a single extreme value. This distinction matters because different anomaly types induce different residual signatures and therefore favor different detection strategies [25, 29, 36]. Henceforth, we follow the taxonomy for anomalies in time series as summarized in Tab. 1. The anomaly type has direct consequences for detection. Global and contextual point anomalies often appear as localized residual spikes and can, in favorable cases, be recovered by scalar scoring rules. Shapelet and seasonal anomalies, by contrast, are expressed through changes in local pattern structure and are therefore more naturally analyzed over subsequences. Trend anomalies are especially challenging because they may unfold gradually, producing residuals that remain individually unremarkable while accumulating into a meaningful deviation over time [25, 36]. As recent benchmark studies show, this type dependence is one reason why no single TSAD method dominates across all settings [25, 29].

Classical TSAD methods span a broad range of modeling assumptions. Statistical approaches such as ARIMA model temporal dependence explicitly and are effective for linear and stationary regimes, but they struggle with more complex nonlinear dynamics.

Table 1: Taxonomy for point-wise (global or contextual) and are range-wise (frequency, trend, or shapelet) anomalies following Refs. [21, 36].

Name	Description	Illustration
Global	Isolated observations deviating from the global distribution	
Contextual	Observations anomalous in local context but not globally	
Frequency	Subsequences with a shape distinct from all others	
Trend	Disruption of periodic structure while overall shape is preserved	
Shapelet	Sustained shift in mean level or series direction	

Density-based methods such as LOF [4], boundary-based methods such as One-Class SVM [30], and isolation-based methods such as Robust Random Cut Forest [12] remain competitive, especially when anomalies can be identified according to local or geometric patterns in the data. On the other hand, deep learning approaches, including recurrent models [15], temporal convolutional networks [2], and Transformers [32], can capture richer nonlinear dependencies, but they often require more task-specific training data and careful tuning to generalize reliably across domains [19, 29, 34].

These considerations define the problem setting of this paper. We study forecasting-based TSAD on univariate time series with heterogeneous anomaly structure and evaluate methods in a setting where point-wise and range-wise anomalies require qualitatively different downstream analysis. This motivates a later separation between sequence modeling, which aims to forecast normal behavior, and anomaly scoring, which must interpret residuals in a type-sensitive manner.

3 LLMs for TSAD

Large Language Models (LLMs) have recently emerged as a promising new paradigm for time-series modeling by treating numerical sequences as serialized token streams and applying autoregressive next-token prediction. Although originally pretrained on natural language, LLMs can transfer their sequence modeling capabilities

to structured temporal data, especially when the underlying signal exhibits regularity such as seasonality, repetition, or smooth trend development [11]. A key reason is that pretraining endows these models with a strong prior over structured and compressible sequences. As many time series exhibit exactly such regularities, prior work has shown that this simplicity bias can already support competitive zero-shot forecasting, which makes LLMs plausible candidates for forecasting-based anomaly detection even without task-specific pretraining [1, 11, 23].

Within TSAD, existing LLM-based approaches can be divided into two broad categories. In *direct prompting* approaches, the model is asked to identify anomalous observations or intervals directly from the input sequence. These methods are attractive because they appear simple and general, but in practice they remain limited by unstable numerical reasoning, prompt sensitivity, and high false-positive rates. A more robust alternative is *forecasting-based detection*, in which the LLM is used only to predict expected normal behavior and anomalies are inferred from deviations between forecast and observation. This formulation better matches the autoregressive strengths of LLMs and avoids relying on the model as a direct numerical classifier [1, 23].

The most relevant prior work for the present paper is SigLLM, which explicitly formalizes this distinction through its *Prompter* and *Detector* pipelines. Its results show that using an LLM as a forecaster is substantially more reliable than prompting it to label anomalies directly, but they also show that simple thresholding on forecast residuals remains insufficient to match specialized TSAD models [1]. In parallel, recent work on LLM adaptation for time-series forecasting, including TIME-LLM [18] and related parameter-efficient fine-tuning approaches, has demonstrated that methods such as LoRA and $(IA)^3$ can improve forecasting quality at low computational cost [14, 24]. However, better forecasting does not by itself resolve the downstream anomaly-scoring problem, which remains largely underexplored in the LLM-based TSAD literature.

A small number of recent studies have explored hybrid LLM-based TSAD systems by combining language models with external anomaly detection or verification modules [6, 33]. These works suggest that hybridization is promising, but they do not isolate whether the main limitation of forecasting-based LLM pipelines lies in the forecaster itself or in the downstream treatment of residuals. Our work focuses precisely on this question. We study whether improvements in LLM forecasting translate into better detection, and we show that they often do not under fixed residual thresholding. Based on this finding, we design a hybrid pipeline in which the LLM is restricted to forecasting normal behavior, while anomaly scoring and residual cleanup are handled explicitly by downstream modules.

4 Hybrid Framework

The central premise of our approach is that, in LLM-based TSAD, the limiting factor is not the existence of an anomaly signal in the forecast residuals, but the ability to extract that signal reliably. We therefore decouple the task into three stages: (1) forecasting expected normal behavior, (2) detecting anomalies from the resulting residuals, and (3) refining the residual signal through imputation before a final detection pass. In this framework, the LLM is used strictly as a forecaster of normal temporal structure rather than as a direct anomaly classifier. The full end-to-end pipeline is illustrated in Fig. 1.

Before describing the three stages in detail, we introduce two technical mechanisms that underpin the pipeline throughout:

- *Subsequence windowing* segments a time series into overlapping windows of length w and allows detectors to operate on local temporal structure rather than on isolated points. This window length is a critical hyperparameter in unsupervised TSAD: if w is too small, the detector may miss the shape of an anomaly; if it is too large, anomalous structure may be diluted by surrounding normal observations [7].
- *Imputation* describes the reconstruction of masked, missing, or corrupted regions from surrounding context. In anomaly detection, imputation is useful not only as a preprocessing tool but also as a post-processing mechanism for stabilizing a signal after initially detected anomalous regions have been removed or masked [9, 16].

Returning to our three-staged pipeline, let the observed univariate time series under consideration be denoted by

$$\mathbf{s} = (s_1, \ldots, s_m) \in \mathbb{R}^m. \tag{1}$$

Given the corresponding LLM forecast

$$\hat{\mathbf{s}} = (\hat{s}_1, \ldots, \hat{s}_m), \tag{2}$$

we define the pointwise absolute residual as

$$r_i = |s_i - \hat{s}_i|. \tag{3}$$

The residual sequence

$$\mathbf{r} = (r_1, \ldots, r_m) \tag{4}$$

serves as the anomaly signal throughout all stages of the pipeline. The three stages together extend the SigLLM Detector baseline along the dimensions of forecaster adaptation, residual scoring, and iterative refinement, as detailed below.

Stage 1 — LLM Forecaster. The first stage models nominal temporal behavior using an open-source decoder-only LLM. In contrast to prompting-based anomaly classification, the model is optimized only for forecasting, so that anomalies emerge indirectly through deviations between observed and predicted values. For a context window length w and a forecast horizon h, the forecaster f_θ generates

$$\hat{\mathbf{s}}_{i:i+h-1} = f_\theta(\mathbf{s}_{i-w:i-1}), \tag{5}$$

where θ denotes either the pretrained model or a variant that is adapted parameter-efficiently. We consider both zero-shot forecasting and parameter-efficient fine-tuning (PEFT) via LoRA and $(IA)^3$ [14, 24]. These methods adapt the forecaster to the target distribution while preserving the pretrained sequential prior of the underlying LLM. Their role in the present framework is not to transform the LLM into an anomaly detector, but to improve the quality of the normal-behavior forecast and, consequently, the quality of the residual signal available to the downstream detector. To improve numerical sensitivity during fine-tuning, we additionally evaluate Number Token Loss (NTL) as a forecasting-oriented objective in settings where regression-aware supervision is beneficial.

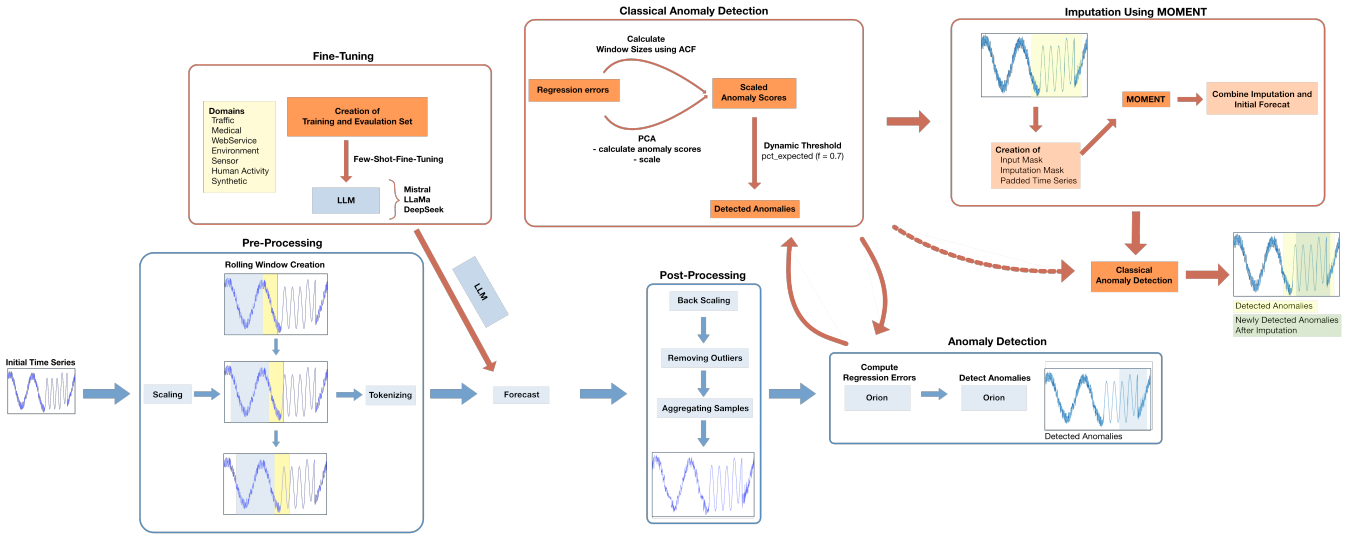


Figure 1: The Forecaster–Detector–Imputer pipeline. Blue components reproduce the SigLLM Detector baseline [1]; orange components are the contributions of this work. The LLM is fine-tuned on anomaly-free segments via LoRA or $(IA)^3$ and restricted to forecasting normal temporal behavior. Classical anomaly detectors operate on the resulting residuals with ACF-based adaptive windowing and contamination-aware thresholding. A subsequent Detect–Impute–Redetect loop uses MOMENT to reconstruct flagged regions, stabilize residuals, and reduce false positives before a final detection pass.

In this way, the token-level loss is augmented with a regression penalty on decoded numerical values, i.e.,

$$\mathcal{L}_{\text{NTL}} = \mathcal{L}_{\text{CE}} + \lambda \cdot \text{MSE}(\text{dec}(\hat{\mathbf{s}}_t),\ \text{dec}(\mathbf{s}_t))\,, \tag{6}$$

where $\mathcal{L}_{\text{CE}}$ denotes the standard cross-entropy loss over next-token predictions, $\text{dec}(\cdot)$ maps a generated token sequence to its scalar value and λ controls the contribution of the regression-aware term [35]. Note that NTL is not a universal component of the framework but rather a model-dependent adaptation mechanism that can sharpen the residual signal before downstream detection.

Stage 2 — Structured Residual Detection. The second stage replaces fixed scalar thresholding on residuals with classical TSAD algorithms that operate on local residual structure. This is the key methodological departure from the SigLLM baseline: anomalies are scored by dedicated detectors in the residual domain rather than by thresholding aggregated forecast errors alone. As our primary detector we employ Sub-PCA and apply it to the residual sequence $\mathbf{r}$. To capture structural anomalies extending across multiple time steps, we form overlapping residual windows

$$x_i = (r_i, r_{i+1}, \ldots, r_{i+w-1})^\top \in \mathbb{R}^w, \quad i = 1, \ldots, m - w + 1. \tag{7}$$

Let $P_k \in \mathbb{R}^{w \times k}$ denote the matrix whose columns are the top-k principal components learned from the collection of these residual windows. The anomaly score for window x_i is then defined as the low-rank reconstruction error

$$a_i = \left\| x_i - P_k P_k^\top x_i \right\|_2^2 . \tag{8}$$

Applied to LLM residuals, this projection suppresses high-frequency forecasting noise while preserving coherent deviation patterns characteristic of structured and range-wise anomalies. The window length w is selected adaptively rather than fixed globally. We use Autocorrelation Function (ACF)-based heuristics to estimate characteristic periodicity either from the original signal or from the residual sequence, depending on which yields the stronger downstream detector performance. In particular, we evaluate both `find_length_rank` and `acf_peak`, which differ in how they prioritize candidate lags and therefore capture different types of temporal structure [3, 25]: `acf_peak` selects the lag at the strongest autocorrelation peak, whereas `find_length_rank` ranks candidate periods by their prominence in the ACF and selects the most dominant one. This is important because the dominant structure of forecast errors do not need to match the dominant structure of the raw series. Continuous anomaly scores are converted to binary labels by contamination-aware thresholding.

Stage 3 — Detect–Impute–Redetect. Even after an initially successful detection pass, anomalous intervals can distort the residual distribution and trigger false positives in neighboring regions. This effect is particularly pronounced for structured and range-wise anomalies, where large residuals persist across multiple windows. To address this, we introduce a detect–impute–redetect loop that stabilizes the residual signal before final decision making. First, an initial detector, typically Sub-PCA with adaptive windowing, produces a candidate anomaly set $\hat{A}^{(1)}$. Second, the detected intervals are masked in the original series and reconstructed with the MOMENT foundation model, yielding a cleaned series

$$\tilde{s} = \text{Impute}(s, \hat{A}^{(1)}). \tag{9}$$

Then, we recompute the residual signal against the original forecaster output as

$$\tilde{r}_i = |\tilde{s}_i - \hat{s}_i|\,. \tag{10}$$

Third, a second detector is applied to the refined residual sequence,

$$\hat{A}^{(2)} = g(\tilde{r}), \tag{11}$$

where g denotes the second-pass anomaly detector (for example Isolation Forest) and $\hat{A}^{(2)}$ indicates the condensed candidate anomaly set. In this way, imputation is used to suppress residual contamination from already detected anomalous regions while preserving the forecasting-based interpretation of anomaly evidence. By replacing strongly corrupted regions with plausible normal-pattern estimates, the second detector operates on a cleaner signal in which true anomalies remain salient while forecast artifacts and spillover noise are reduced. In this sense, imputation is not merely a post-processing convenience, but a structural component of the pipeline that improves the separation between genuine anomaly patterns and residual noise [9].

The proposed three-stage architecture is shared across all evaluated backbone models, but its strongest instantiations are model-specific. Our experiments show that selected forecaster adaptations can be beneficial for individual models, whereas the dominant overall gain is driven by the combination of residual-based detection and imputation rather than by forecasting improvement alone. We therefore treat the Forecaster–Detector–Imputer framework as a common architectural principle with model-dependent forecaster and detector choices.

5 Experimental setup

This section summarizes the datasets, models, baseline configuration, fine-tuning setup, and evaluation metrics used in all experiments.

Datasets: We evaluate on TAMA [36], which provides point-wise labels for five structural anomaly types: global point, contextual point, shapelet, seasonal, and trend. Following the anomaly-centered protocol, long source sequences are segmented so that each evaluation sequence contains exactly one labeled anomaly event. We restrict evaluation to the real-world subsets only. After excluding the first 140 observations required by the forecasting lookback, the resulting benchmark comprises 13 real-world segments with a mean anomaly contamination rate of approximately 27.9%. For few-shot forecaster adaptation, anomaly-free training data is drawn from TSB-AD-U [25]. From each source series, we extract the longest continuous anomaly-free segment, ensuring that fine-tuning remains restricted to nominal behavior. As the evaluation set consists of 13 anomaly-centered real-world segments, we treat it as a controlled benchmark for comparing design choices rather than as evidence of universal superiority across TSAD settings.

Models and implementation: For the sake of reproducibility, we evaluate three open-source decoder-only LLMs in the 7–8B range: Mistral-7B [17], LLaMA-3.1-8B-Instruct [10], and DeepSeek-R1-Distill-Qwen-7B [5]. Mistral is included to maintain comparability with the open-source SigLLM setting, while LLaMA and DeepSeek extend the analysis to instruction-tuned and reasoning-distilled model variants. All models are loaded in 4-bit quantization using Unsloth [13]. Experiments are conducted in Google Colab on a single NVIDIA T4 (15 GB VRAM) or L4 (22.5 GB VRAM) GPU. During generation, outputs are constrained to valid digit tokens in order to reduce malformed numerical forecasts.

Baseline: Our baseline follows the SigLLM Detector pipeline, with the three open-source backbones substituted for the original closed-source model. Time series values are scaled to integers and segmented into rolling windows with lookback size $w = 140$ and step size 1, then serialized as comma-separated digit strings. For each window, the model generates two forecast samples using temperature $T = 0.7$ and nucleus sampling with $p = 0.95$. Before aggregation, outliers in the overlapping forecasts are identified by calculating the interquartile ranges, and the remaining predictions are then aggregated point-wise using the median. Anomalies are then obtained by fixed thresholding on the aggregated residuals, following the SigLLM-style post-processing logic.

Fine-tuning configurations: Our configuration covers few-shot sample counts $k \in \{3, 10\}$, forecast horizons $h \in \{6, 48\}$, and context windows $w \in \{20, 140\}$. Optimization uses AdamW [26] with learning rate 2×10^{-4}, a linear schedule, and early stopping with patience 3. For LoRA, we use rank $r = 16$ and scaling factor $\alpha = 32$.

Evaluation metrics: We evaluate both forecasting quality and anomaly detection quality and measure the forecasting performance using Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Detection performance is measured using point-wise Precision, Recall, and F1 score. To account for temporal extent, we additionally report the range-aware metrics VUS-ROC and VUS-PR [28]. VUS-ROC is particularly informative in our setting because it reveals whether anomaly information is already present in the residuals even when fixed thresholding yields comparatively weak F1 scores.

6 Results

In this section, we report on the results in three stages that mirror the pipeline structure. First, we test whether PEFT or a regression-aware training objective improve anomaly detection (Sects. 6.1–6.2). Second, we show that the residual scoring step is the primary driver of detection quality (Sect. 6.3). Third, we examine per-type performance and validate the findings on a second benchmark (Sects. 6.4–6.5). As an overview, Tab. 2 shows how each pipeline stage contributes to the final F_1, with cumulative gains of up to +100.2% over the SigLLM baseline (Mistral) and +96.3% (LLaMA).

6.1 PEFT: Forecasting vs. Detection

To test whether fine-tuning turns an LLM into a better model of normal temporal behavior, we evaluate forecasting errors separately on normal and anomalous subsequences. The goal is to reduce prediction error on normal regions while preserving or enlarging residuals on anomalous ones, thereby widening the signal available to downstream detectors.

LoRA. LLaMA ($k = 3$, $w = 140$, $h = 48$) is the strongest forecaster on normal regions, reducing MSE by 9.76% relative to the zero-shot baseline while leaving anomalous-region errors nearly unchanged (+0.32% MSE). Across the full configuration grid, nearly all LLaMA variants produce the lowest normal-region prediction errors. Fewer training samples ($k = 3$) consistently outperform $k = 10$, suggesting that the larger shot count induces cross-domain overgeneralization and flattens residuals in both normal and anomalous regions.

Table 2: Anomaly detection performance (F_1, averaged over 13 TAMA segments) along the development path from the SigLLM zero-shot baseline to the strongest practical and idealized configurations for each LLM family. Relative gains are computed against the corresponding SigLLM baseline. RE denotes raw LLM regression errors; ImpRE denotes MOMENT-imputation-enhanced regression errors. The forecasting horizon is denoted by h, the number of few-shot training samples by k, and the ensemble size by K.

Model	Stage	Configuration	F1	Rel. Gain
Mistral	SigLLM Baseline	Zero-shot	0.392	—
	Hybrid (Zero-shot, RE)	Sub-PCA (`acfpeak`, pct70)	0.708	+80.7%
	Hybrid + PEFT (RE)	LoRA h=48, k=3; PCA (`acfpeak`, pct70)	0.726	+85.2%
	Imputation (ImpRE)	Sub-PCA (fixed w=20, pct70)	0.732	+86.8%
	NTL λ=0.8 (RE)	Sub-PCA (`acfpeak`, pct70)	0.760	+93.9%
	Best practical (ImpRE)	**NTL + IForest (`findlengthrank`, pct70)**	**0.785**	**+100.2%**
	Realistic ensemble (RE)	Top-k Per-Variant, K=3	0.723	+84.4%
	Idealized ensemble (RE)	Perfect Specialist, K=5	0.760	+93.9%
LLaMA	SigLLM Baseline	Zero-shot	0.402	—
	Hybrid (Zero-shot, RE)	Sub-PCA (`acfpeak`, pct70)	0.741	+84.2%
	Hybrid + PEFT (RE)	LoRA h=48, k=10; Sub-PCA (`acfpeak`, pct70)	0.741	+84.3%
	Best practical (ImpRE)	**IForest (`findlengthrank`, pct70)**	**0.789**	**+96.3%**
	Realistic ensemble (RE)	Top-k Per-Variant, K=2	0.760	+89.1%
	Idealized ensemble (RE)	Perfect Specialist, K=5	0.832	+106.9%
DeepSeek	SigLLM Baseline	Zero-shot	0.483	—
	Hybrid (Zero-shot, RE)	Sub-PCA (`acfpeak`, pct70)	0.715	+48.1%
	Best practical (RE)	**LoRA h=48, k=3; Sub-PCA (`findlengthrank`, pct70)**	**0.763**	**+58.1%**
	Imputation (ImpRE)	IForest (fixed w=20, pct70)	0.758	+56.9%
	Realistic ensemble (RE)	Top-k Per-Variant, K=3	0.758	+57.0%
	Idealized ensemble (RE)	Perfect Specialist, K=5	0.769	+59.2%

For Mistral, a shorter window size ($w = 20$) yields the strongest local forecasting gain, reducing normal-region MSE by 30% and anomalous-region MSE by 62.99%. This suggests that the standard $w = 140$ window averages out local structures important for short-range forecasting. However, these forecasting gains do not translate into reliable end-to-end anomaly detection improvements within the original SigLLM pipeline. Detection gains remain marginal for all models, typically within 0–3% in F1.

The strongest detection improvement in this setting is observed for Mistral ($k = 3$, $w = 140$, $h = 6$), which reaches +11.63% AUC-ROC and +6.77% VUS-ROC. Notably, this configuration increases normal-region MSE by 5.0% and reduces anomalous-region MSE by 2.27%, directly contradicting the hypothesis that greater normal–anomalous forecasting separation alone predicts better detection.

$(IA)^3$. The fine-tuning with $(IA)^3$ ($k = 3$, $w = 140$, $h = 6$, Mistral) yields only marginal forecasting improvements in both normal and anomalous regions (0–2.5%) while achieving the lowest normal-region errors among all tested PEFT configurations for Mistral (MAE= 0.093, MSE= 0.039, RMSE= 0.197). Yet downstream detection remains slightly below the corresponding baseline, with AUC-ROC= 0.508 and VUS-ROC= 0.650, compared with 0.516 and 0.656 in the same experimental context. Since $(IA)^3$ offered no detection advantage over either the baseline or LoRA, it was not applied to the remaining model families.

ACF-Based Windowing. Using ACF-based window estimation as a forecasting context in the SigLLM pipeline — evaluated on the zero-shot Mistral baseline — reduces forecasting error on anomalous regions but slightly worsens detection. AUC-ROC drops from 0.491 to 0.473, AUC-PR from 0.338 to 0.292, and VUS-ROC from 0.648 to 0.647, while VUS-PR changes only marginally from 0.466 to 0.470. The resulting normal–anomalous separation remains close to chance. ACF windowing is therefore not retained for the SigLLM detection step itself, but is instead repurposed in Sect. 4 as a window estimator for the downstream classical detector, where it proves beneficial.

Symbolic Input Formats. We evaluate a symbolic enrichment variant in which each serialized time-series value is prefixed with a direction indicator (↑, ↓, →) reflecting whether the scaled value increased, decreased, or remained constant relative to the previous step [8]. We find that symbolic enrichment yields inconsistent and model-specific effects, without any robust TSAD benefit across architectures. For Mistral, symbolic inputs increase normal-region MSE by 170% while increasing anomalous-region MAE by only 10.34%, producing only a slight detection gain (AUC-ROC 0.521, VUS-ROC 0.670 vs. baseline 0.516 / 0.656). For DeepSeek, symbolic inputs reduce anomalous-region errors by roughly 13% on average, yet all four threshold-free detection metrics decrease, indicating worse class separation. No symbolic input format generalizes consistently across the three LLM families.

Table 3: Top-5 anomaly detection configurations on raw time series vs. LLM forecast residuals (zero-shot hybrid, `acf_peak_pct_expected_70`). Bold entries appear in multiple top-5 lists. Residuals from LLaMA achieve $F_1 = 0.741$, a 37% improvement over the best raw-series method (POLY, $F_1 = 0.540$). Sub-PCA ranks first across all three LLM backbones.

Original Time Series			Mistral (Regression Errors)		
Rank	**Model (Variant)**	**F1**	**Rank**	**Model (Variant)**	**F1**
1	**POLY** (`fixed20`, pct70)	0.540	1	**Sub-PCA** (`acfpeak`, pct70)	0.708
2	Sub-IForest (`acfpeak`, pct70)	0.524	2	**Sub-PCA** (`flen`, pct70)	0.702
3	Series2Graph (`fixed20`, pct70)	0.520	3	**Sub-PCA** (`fixed20`, pct70)	0.663
4	Sub-IForest (`acfpeak`, pot70)	0.513	4	**PCA** (`acfpeak`, pct70)	0.663
5	Sub-IForest (`flen`, pot70)	0.512	5	PCA (`fixed20`, pct70)	0.647
LLaMA (Regression Errors)			**DeepSeek (Regression Errors)**		
Rank	**Model (Variant)**	**F1**	**Rank**	**Model (Variant)**	**F1**
1	**Sub-PCA** (`acfpeak`, pct70)	**0.741**	1	**Sub-PCA** (`acfpeak`, pct70)	0.715
2	**Sub-PCA** (`flen`, pct70)	0.712	2	**PCA** (`acfpeak`, pct70)	0.706
3	**PCA** (`acfpeak`, pct70)	0.695	3	**Sub-PCA** (`flen`, pct70)	0.686
4	**Sub-PCA** (`acfpeak`, pot70)	0.682	4	PCA (`flen`, pct70)	0.684
5	**Sub-PCA** (`fixed20`, pct70)	0.667	5	**Sub-PCA** (`acfpeak`, pot70)	0.684

6.2 Number Token Loss

Standard cross-entropy assigns the same penalty to predicting token "5" instead of "4" and predicting "9" instead of "4", even though the two numerical errors are qualitatively different. Following Zausinger et al. [35], we therefore augment the training objective with the NTL from Eq. 6. The NTL can be integrated into the standard Unsloth fine-tuning workflow without additional heads or architectural modifications; however, its applicability proved model-dependent. DeepSeek would have required substantial architectural modifications that were outside the scope of this work, and LLaMA produced numerical hallucinations across all tested $\lambda \in \{0.3, 0.5, 0.8\}$, reverting to repetitive non-digit token sequences and yielding unusable forecasts. Therefore, NTL we applied to Mistral only. For the latter, an NTL weighting with $\lambda = 0.8$ provides a reasonable trade-off between residual separation and numerical stability. Fairly lower weights, like $\lambda = 0.3$, leave detection essentially unchanged relative to the baseline (F1= 0.427), while larger weights, like $\lambda = 1.0$, cause numerical instability and hallucinated outputs, and were therefore excluded.

Adding NTL fine-tuning to the hybrid pipeline improves Mistral's F1 score from 0.708 to 0.760 on raw residuals. After MOMENT imputation, F1 reaches 0.785, i.e.,roughly doubling the SigLLM zero-shot baseline of 0.392.

6.3 Hybrid Pipeline: Decoupling Forecasting from Detection

Although F1 scores under fixed-threshold SigLLM scoring remain low (0.392–0.483), VUS-ROC stays above 0.65 for all three model families. This gap shows that the residuals already contain a strong anomaly signal, but that static point-wise thresholding fails to extract it effectively. Replacing fixed thresholding with classical detectors operating on the residual sequence recovers this signal.

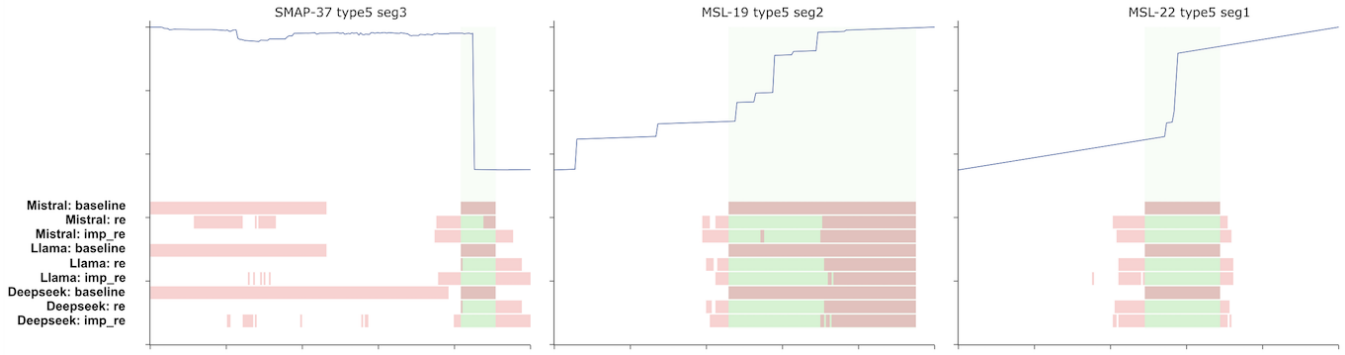
Stage 1 — LLM Forecaster. Sub-PCA with `acf_peak` windowing and a `pct_expected_70` contamination-aware threshold yields F_1 gains above 80% for Mistral and LLaMA and 48% for DeepSeek relative to their respective SigLLM baselines. The smaller relative gain for DeepSeek reflects its stronger baseline ($F_1 = 0.483$), not a weaker hybrid effect. As shown in Tab. 3, applying Sub-PCA to LLM forecast residuals instead of the raw series raises the F_1 score by 37%, i.e., from 0.54 to 0.74. This confirms that the LLM forecasting stage makes anomalies substantially easier to detect.

Stage 2 — Structured Residual Detection. As shown in Tab. 3, applying Sub-PCA to LLM forecast residuals instead of the raw series raises F_1 from 0.54 to 0.74 (approx. 37%), confirming that residual-based detection substantially outperforms detection on the raw series. Within the hybrid framework, PEFT yields additional but model-dependent gains. For LLaMA ($h = 48$, $k = 10$, Sub-PCA with `acf_peak`), fine-tuning produces essentially no improvement over the zero-shot hybrid ($F_1 = 0.741$ vs. 0.741), suggesting that LLaMA already provides near-optimal residuals for downstream detection. For DeepSeek ($h = 48$, $k = 3$, Sub-PCA with `find_length_rank`), PEFT yields the largest gain among the three model families in the hybrid context, reaching $F_1 = 0.763$ (+58.1% vs. baseline). For Mistral ($h = 48$, $k = 3$, PCA with `acf_peak`), PEFT reaches $F_1 = 0.726$ (+85.2%). A notable reversal emerges relative to the original SigLLM pipeline. There, short horizons ($h = 6$) were more favorable for detection. In the hybrid setup, longer horizons ($h = 48$) consistently work better because they stabilize rolling-window forecasts and provide Sub-PCA with a cleaner estimate of normal behavior.

Stage 3 — Detect–Impute–Redetect. The dominant failure mode of the zero-shot hybrid is false positives: residual contamination in already-flagged regions distorts later detection passes. The detect–impute–redetect loop targets this directly. Sub-PCA first identifies candidate anomaly regions, MOMENT reconstructs these regions with plausible normal-behavior estimates, and a second pass of



(a) Detection results for global point anomalies across three MSL segments.



(b) Detection results for trend anomalies across SMAP and MSL segments.

Figure 2: Detection results for global point anomalies (top) and trend anomalies (bottom) across Mistral, LLaMA, and DeepSeek. Bars show true positives (green), false positives (light red), and false negatives (dark red) for three configurations: SigLLM baseline, hybrid residual detection (RE), and hybrid with imputation (Imp_RE). For point anomalies, Imp_RE yields the tightest boundaries with fewest false positives. For trend anomalies, the baseline largely fails; RE and Imp_RE successfully capture the trend shift.

Isolation Forest operates on the cleaned residuals. As exemplarily depicted in Fig. 2, this imputation step consistently reduces false positives across all three LLM families while preserving or improving recall, adding roughly 2–5% F1 on top of the zero-shot hybrid. The mechanism is especially clear in individual segments. For Mistral on `MSL-19` (type 5, trend), false positives drop from 101 in the SigLLM baseline to 52 after Sub-PCA on raw residuals and to 25 after MOMENT imputation, while recall reaches 1.0 and F1 improves from 0.000 to 0.615. For LLaMA on `NormA-4type3`, false positives decrease by more than 65% (from 64 to 16), precision rises to 0.901, and F1 improves from 0.606 to 0.806. For DeepSeek, by contrast, the effect of imputation is less consistent: gains on shapelet segments are offset by additional false positives on some trend segments, indicating that DeepSeek benefits more from PEFT alignment than from residual reconstruction.

6.4 Per-Type Analysis and Ensemble Ceiling

No single model–detector pair dominates all five anomaly categories (Tab. 4). Type 4 (seasonal) is solved almost perfectly by Mistral with Isolation Forest ($F_1 = 1.00$), consistent with the idea that periodic residual structure is especially easy for a tree-based density estimator to separate. Types 1 and 2 (point anomalies) are handled best by LLaMA with MERLIN [27] and Sub-PCA, respectively, while types 3 (shapelet) and 5 (trend) benefit most from dimensionality reduction through Sub-PCA or PCA. LLaMA dominates four of the five specialist categories, suggesting that it produces the cleanest residuals in terms of signal-to-noise ratio. Still, the variation across anomaly types shows that there is no single universally optimal detector in this setting. Selecting the best-performing preprocessing configuration and TSAD algorithms for each anomaly type based on ground-truth type labels yields $F_1 = 0.832$ for LLaMA and $F_1 = 0.769$ for DeepSeek on the 13-segment TAMA evaluation

Table 4: Optimized model–detector configuration per anomaly type on the 13-segment TAMA evaluation set, selected by highest F1. Each row shows the LLM backbone, classical detector, and ACF-based window estimator (`acf_peak` or `find_length_rank`) that jointly achieve the highest F1 for that anomaly category. All configurations use a contamination-aware threshold at $\rho = 0.70$.

Type	Model	Detector	Windowing	F1
1 (Global Point)	LLaMA	MERLIN	`acf`	0.79
2 (Contextual Point)	LLaMA	Sub-PCA	`acf`, pct70	0.78
3 (Shapelet)	LLaMA	Sub-PCA	`flen`, pct70	0.75
4 (Seasonal)	Mistral	IForest	`flen`, pct70	1.00
5 (Trend)	LLaMA	PCA	`acf`, pct70	0.73

Table 5: Window-based evaluation on 19 NAB files across three subsets. For each subset, true positives (TP), false positives (FP), false negatives (FN), and window-based F_1 are reported. Our FDI pipeline (LLaMA forecaster, Sub-PCA residual detector, MOMENT imputation) is compared against the SigLLM-Mistral baseline. Bold F_1 values indicate the better-performing system per row.

	FDI (LLaMA)				SigLLM-Mistral			
Subset	TP	FP	FN	F_1	TP	FP	FN	F_1
`artificialWithAnomaly`	7	6	2	**0.636**	2	2	4	0.400
`realAdExchange`	11	16	3	0.537	8	3	3	**0.727**
`realTraffic`	27	6	2	**0.871**	6	5	8	0.480
Grand Total	45	28	7	**0.720**	16	10	15	0.561

set, and thus constitutes a theoretical upper bound for type-aware detector selection.[1] A realistic Top-k per-variant ensemble that requires no anomaly-type knowledge at inference time reaches $F_1 = 0.760$ for LLaMA, $F_1 = 0.758$ for DeepSeek, and $F_1 = 0.723$ for Mistral. The remaining gap between the realistic ensemble and this upper bound is therefore primarily a type-identification problem at inference time.

6.5 NAB Evaluation on a Subset of the Benchmark

To assess whether the pipeline's detection ability extends beyond the TAMA benchmark, we additionally evaluate the LLaMA-based FDI pipeline on a subset of the Numenta Anomaly Benchmark (NAB) [22]. Unlike TAMA, NAB uses a streaming-aware scoring rule that credits only the first correct detection per anomaly window, applies time-decaying rewards, and penalizes false alarms, normalizing results between 0 and 100. This makes NAB complementary to point-wise F_1: it rewards early and precise detections rather than aggregate overlap.

We evaluate the three subsets `realTraffic`, `realAdExchange`, and `artificialWithAnomaly`, i.e., 19 files in total.[2] The pipeline uses LLaMA for forecasting, Sub-PCA with adaptive ACF-based windowing for initial residual-based detection, and MOMENT-based imputation for residual refinement, consistent with the configuration evaluated on TAMA.

Our results are summarized in Tab. 5. The pipeline reaches a Grand Total F_1 of 0.720 against 0.561 for the SigLLM-Mistral baseline, with `realTraffic` yielding the strongest result (F_1 = 0.871 vs. 0.480). The Standard NAB score across the three subsets reaches 60.66, with `realTraffic` scoring 81.14; six of the seven `realTraffic` files reach scores above 87 and five exhibit zero false positives at the optimal threshold. The coexistence of a moderate point-wise F_1 of 0.213 and a subset-level Standard NAB score of 60.66 on `realTraffic` (F_1: 0.409, NAB Standard: 81.14) illustrates a systematic underestimation of LLM-based pipeline quality by point-wise metrics: detections lie near anomaly windows without matching exact labels, which point-wise F_1 penalizes but NAB's sigmoid reward tolerates. The `artificialWithAnomaly` subset scores 53.50; two files yield negative scores of −11.00 each, reflecting missed windows combined with residual false positives.

The NAB results confirm that the imputation step reduces false positives and improves detection quality on a second independent benchmark.

7 Conclusion

The central finding of this work is that the anomaly signal in LLM forecast residuals is considerably stronger than zero-shot F_1 scores suggest. When LLaMA, Mistral, and DeepSeek are evaluated without any fine-tuning, their residuals already rank anomalous regions reliably above normal ones, as reflected by VUS-ROC scores exceeding 0.65 across all three architectures. The low corresponding F_1 scores, all below 0.49, are therefore attributable to the inadequacy of static point-wise thresholding rather than to the quality of the forecasts themselves. Replacing this thresholding step with a classical anomaly detector operating on the residuals, stabilising those residuals via imputation of previously detected anomalous regions, and aligning the fine-tuning objective more closely with numerical prediction together raise the peak F_1 to 0.789 without recourse to additional labeled data. These findings reframe LLM-based TSAD as a residual exploitation problem, suggesting that future progress lies in more elaborate post-hoc analysis rather than forecasting optimization alone.

Acknowledgments

The authors gratefully acknowledge the support of Jiaxin Zhuang, Leon Yan, Zhenwei Zhang, Ruiqi Wang, Jiawei Zhang, Yuantao Gu, who provided access to the TAMA dataset. Their assistance in sharing the annotated data enabled us to conduct the experiments described in this work. Parts of the project have been developed in the project PLan_CV (reference number 03FHP109), which is funded by the German Federal Ministry of Research, Technology, and Space (BMFTR) and Joint Science Conference (GWK). The contribution of HA was carried out under the project Advanced Technologies for High Energetic Atmospheric Flight of Launcher Stages (ATHEAt) by the German Aerospace Center (DLR).

[1] Fairly spoken, this setting is not feasible at inference time. Automating this selection is left for future work.

[2] The remaining NAB subsets (`realAWSCloudwatch`, `realKnownCause`, `realTweets`, and the `artificialNoAnomaly` stream set) are outside the scope of this study. The aggregate figures reported below are therefore conditioned on this subset and must not be read as a full NAB benchmark result.

GenAI Usage Disclosure

In this manuscript, generative AI (Claude) was used exclusively for refinement of author-written text, i.e., grammatical and typographical corrections, rephrasing for clarity, and minor stylistic polishing.

References

[1] Sarah Alnegheimish, Linh Nguyen, Laure Berti-Equille, and Kalyan Veeramachaneni. 2024. Large language models can be zero-shot anomaly detectors for time series? arXiv:2405.14755 [cs.LG] https://arxiv.org/abs/2405.14755

[2] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv:1803.01271 [cs.LG]

[3] Kristiyan Blagov, Carlos Enrique Muñiz-Cuza, and Matthias Boehm. 2025. Fast, Parameter-Free Time Series Anomaly Detection. In *Proceedings of the 20th International Conference on Database Systems for Business, Technology and Web (BTW 2025)*. Gesellschaft für Informatik, 453–474. https://dl.gi.de/items/10727095-61f5-4c39-8766-f60e518268e5

[4] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. ACM, 93–104. doi:10.1145/342009.335388

[5] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] https://arxiv.org/abs/2501.12948

[6] Manqing Dong, Hao Huang, and Longbing Cao. 2024. Can LLMs Serve As Time Series Anomaly Detectors? arXiv:2408.03475 [cs.LG] https://arxiv.org/abs/2408.03475

[7] Arik Ermshaus, Patrick Schäfer, and Ulf Leser. 2022. Window Size Selection in Unsupervised Time Series Analytics: A Review and Benchmark. In *Advanced Analytics and Learning on Temporal Data: 7th ECML PKDD Workshop, AALTD 2022, Grenoble, France, September 19–23, 2022, Revised Selected Papers* (Grenoble, France). Springer-Verlag, Berlin, Heidelberg, 83–101. doi:10.1007/978-3-031-24378-3_6

[8] Elizabeth Fons, Rachneet Kaur, Soham Palande, Zhen Zeng, Tucker Balch, Manuela Veloso, and Svitlana Vyetrenko. 2024. Evaluating Large Language Models on Time Series Feature Understanding: A Comprehensive Taxonomy and Benchmark. arXiv:2404.16563 [cs.CL] https://arxiv.org/abs/2404.16563

[9] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. MOMENT: A Family of Open Time-series Foundation Models. arXiv:2402.03885 [cs.LG] https://arxiv.org/abs/2402.03885

[10] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[11] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2024. Large Language Models Are Zero-Shot Time Series Forecasters. arXiv:2310.07820 [cs.LG] https://arxiv.org/abs/2310.07820

[12] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. 2016. Robust Random Cut Forest Based Anomaly Detection on Streams. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. 2712–2721.

[13] Daniel Han, Michael Han, and Unsloth team. 2023. *Unsloth*. http://github.com/unslothai/unsloth

[14] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

[15] Kyle Hundman, Valentinos 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*. ACM, 387–395. doi:10.1145/3219819.3219845

[16] Michel Jacobsen and Marina Tropmann-Frick. 2024. Imputation Strategies in Time Series Based on Language Models. *Datenbank-Spektrum* 24, 3 (2024), 197–207. doi:10.1007/s13222-024-00488-z

[17] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[18] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. arXiv:2310.01728 [cs.LG] https://arxiv.org/abs/2310.01728

[19] Jongseon Kim, Hyungjoon Kim, HyunGi Kim, Dongjun Lee, and Sungroh Yoon. 2025. A Comprehensive Survey of Deep Learning for Time Series Forecasting: Architectural Diversity and Open Challenges. arXiv:2411.05793 [cs.LG] https://arxiv.org/abs/2411.05793

[20] Wadim Koslow, Kathrin Rack, Tobias D. Grabosch, Alexander Rüttgers, Luca Dell Amore, and Paola Rizzoli. 2026. Patch-based anomaly detection on SAR images to localize hotspots on the North and Baltic Sea coasts. *Remote Sensing Applications: Society and Environment* 41 (2026), 101958. doi:10.1016/j.rsase.2026.101958

[21] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. 2021. Revisiting Time Series Outlier Detection: Definitions and Benchmarks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*. https://openreview.net/forum?id=r8IvOsnHchr

[22] 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)*. IEEE, 38–44. doi:10.1109/icmla.2015.141

[23] Kunqi Li, Thomas Pitts, Bin Liang, Chenya Huang, and Yuxi Lu. 2026. A Survey of Large Language Models for Time Series Anomaly Detection: Methods, Challenges, and Future Directions.

[24] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tsendsuren Huang, Mohit Bansal, and Colin Raffel. 2022. Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 35. 1950–1965.

[25] Qinghua Liu and John Paparrizos. 2024. The Elephant in the Room: Towards A Reliable Time-Series Anomaly Detection Benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. https://openreview.net/forum?id=R6kJtWsTGy

[26] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=Bkg6RiCqY7

[27] Takaaki Nakamura, Makoto Imamura, Ryan Mercer, and Eamonn Keogh. 2020. MERLIN: Parameter-Free Discovery of Arbitrary Length Anomalies in Massive Time Series Archives. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1190–1195.

[28] John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S. Tsay, Aaron Elmore, and Michael J. Franklin. 2022. Volume Under the Surface: A New Accuracy Evaluation Measure for Time-Series Anomaly Detection. *Proceedings of the VLDB Endowment* 15, 11 (2022), 2774–2787. doi:10.14778/3551793.3551830

[29] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly detection in time series: a comprehensive evaluation. *Proc. VLDB Endow.* 15, 9 (May 2022), 1779–1797. doi:10.14778/3538598.3538602

[30] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. 2001. Estimating the Support of a High-Dimensional Distribution. *Neural Computation* 13, 7 (2001), 1443–1471. doi:10.1162/089976601750264965

[31] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. arXiv:2210.02186 [cs.LG] https://arxiv.org/abs/2210.02186

[32] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2021. Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. *CoRR* abs/2110.02642 (2021). arXiv:2110.02642 https://arxiv.org/abs/2110.02642

[33] Alan Yang, Yulin Chen, Sean Lee, and Venus Montes. 2025. Refining Time Series Anomaly Detectors using Large Language Models. arXiv:2503.21833 [cs.CL] https://arxiv.org/abs/2503.21833

[34] Zahra Zamanzadeh Darban, Geoffrey I. Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. 2024. Deep Learning for Time Series Anomaly Detection: A Survey. *Comput. Surveys* 57, 1 (Oct. 2024), 1–42. doi:10.1145/3691338

[35] Jonas Zausinger, Lars Pennig, Anamarija Kozina, Sean Sdahl, Julian Sikora, Adrian Dendorfer, Timofey Kuznetsov, Mohamad Hagog, Nina Wiedemann, Kacper Chlodny, Vincent Limbach, Anna Ketteler, Thorben Prein, Vishwa Mohan Singh, Michael Morris Danziger, and Jannis Born. 2025. Regress, Don't Guess – A Regression-like Loss on Number Tokens for Language Models. arXiv:2411.02083 [cs.CL] https://arxiv.org/abs/2411.02083

[36] Jiaxin Zhuang, Leon Yan, Zhenwei Zhang, Ruiqi Wang, Jiawei Zhang, and Yuantao Gu. 2024. See it, Think it, Sorted: Large Multimodal Models are Few-shot Time Series Anomaly Analyzers. arXiv:2411.02465 [cs.LG] https://arxiv.org/abs/2411.02465