Online Learning Supported by Foundation Models for Anomaly Detection in Industrial Settings

Aurora Esteban Toscano[1,*], Sepideh Pashami[1], Felix Nilsson[2], Luka Smeets[3] and Sławomir Nowaczyk[1]

[1]Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Halmstad, Sweden
[2]HMS Industrial Networks AB, Halmstad, Sweden
[3]HQ RAXTAR, Veldhoven, The Netherlands

Abstract

Modern industrial monitoring systems must detect anomalies in real time under evolving operating conditions and without reliance on labeled data. Traditional online anomaly detectors offer fast adaptation but struggle when normal behavior shifts or when rare anomalies are unintentionally learned as normal. On the other side, recently introduced foundation models for time series capture richer structure but are computationally expensive for continuous deployment.

We propose a dual-learner anomaly detection framework that bridges a fast online learner based on Half-Space Trees with a time-series foundation model (MOMENT) acting as a background learner. A confidence-based routing mechanism determines, for each incoming instance, whether to trust the online model, defer to the foundation model, or combine both through confidence-weighted ensembling. The confidence estimation method is fully unsupervised and robust to drift, requiring no labels or sliding windows.

We validate the approach on two real-world elevator (hoist) installations, demonstrating that the system operates efficiently in streaming conditions and matches or surpasses strong online baselines. Furthermore, we show that fine-tuning the foundation model on one installation provides measurable performance gains when transferred to a different installation, indicating that foundation-model adaptation can support cross-site knowledge transfer in industrial monitoring. The results highlight the promise of integrating online learning with foundation models to achieve both responsiveness and robustness in long-term industrial anomaly detection.

Keywords

Streaming Machine Learning, Time series analysis, Transfer learning, Anomaly Detection

1. Introduction

Modern industrial systems generate large volumes of sensor measurements at high frequency, often under continuously evolving operating conditions. In these environments, *online learning* is particularly attractive because models must process data as it arrives, without relying on long-term data storage or expensive offline retraining [1]. At the same time, real-world deployment constraints require minimal supervision, fast adaptation, and robustness to distributional shifts.

A central challenge in this context is anomaly detection [2]. Malfunctions are rare by definition, leading to an extreme imbalance where only normal data is abundant. Furthermore, labels for anomalous events are typically unavailable or arrive with significant delay, making fully supervised learning impractical. As a result, semi-supervised or unsupervised approaches—particularly those that learn normal behavior and detect deviations—are widely used. Common online anomaly detectors such as statistical models and incremental tree ensembles (e.g., Half-Space Trees) offer fast inference and incremental updates, but their performance degrades in complex industrial environments due to high dimensionality, varying anomaly signatures, and model contamination when anomalous instances are unintentionally incorporated into training [3].

1st Streaming Continual Learning Bridge at AAAI26, January 21, 2026, Singapore.

[*] Corresponding author.

aurora.esteban.toscano@hh.se (A. Esteban Toscano)
0000-0002-6290-5353 (A. Esteban Toscano); 0000-0003-3272-4145 (S. Pashami); 0000-0003-4524-4204 (F. Nilsson); 0000-0002-7796-5201 (S. Nowaczyk)

To address these limitations, we propose a dual-learner system that bridges fast online anomaly detection with the representational capacity of a time-series foundation model. Our design is inspired by the "fast and slow thinking" framework from cognitive psychology [4]: a lightweight, adaptive online model provides rapid decisions under normal conditions (*"think fast"*), while a more expressive but computationally heavier background model (*"think slow"*) is invoked when additional reasoning is needed. Specifically, we combine:

- **Half-Space Trees** [5] as an efficient online anomaly detector capable of processing high-speed industrial data streams.
- **MOMENT** [6], a large-scale foundation model for time series analysis, used as a background model to refine uncertain predictions and detect subtle temporal deviations.

A confidence-driven routing mechanism governs the interaction between the two learners. The online detector makes the initial prediction and assesses its reliability. If the confidence is low, the system queries the foundation model. If both models exhibit uncertainty, their anomaly scores are combined via confidence-weighted ensembling. This design enables real-time operation under limited compute while leveraging the generalization ability of a large pre-trained model when necessary.

A motivating application for this work is the monitoring of industrial elevator (hoist) systems used in high-rise construction projects. These systems operate under rapidly evolving mechanical and environmental conditions, leading to non-stationary sensor distributions and a mixture of subtle and abrupt anomalies. Ride durations, loading patterns, and vibration signatures vary throughout the construction process, making it difficult for purely online or purely offline approaches to remain reliable over long periods. This domain highlights the need for a method that combines (i) fast instance-by-instance adaptation to local changes and (ii) stable, domain-general representations that are robust to drift and noise. While our proposed architecture is general and applicable to any multivariate time-series stream, the industrial hoist scenario provides a concrete real-world environment in which these challenges—and the benefits of our dual-learner design—naturally arise. Thus, we evaluate our system on real industrial elevator monitoring data across two deployment projects. Additionally, we study the effect of fine-tuning the foundation model on data from the first project and transferring the adapted model to the second project. Our results demonstrate that foundation model adaptation improves performance and robustness of the online learner through indirect knowledge transfer, opening a new direction for integrating online learning and foundation models in industrial applications.

This work makes the following contributions:

- We introduce a dual-learner anomaly detection architecture that integrates a fast online model with a foundation time-series model via confidence-based routing.
- We propose a dual confidence estimation method that is fully unsupervised and allows reliable model-switching under scarce anomalies based on the consistency of the predictions produced by both models.
- We demonstrate that fine-tuning the foundation model in one industrial project improves online anomaly detection in a second project, revealing a pathway for domain-level transfer learning in streaming applications.
- We validate the proposed approach on real-world industrial systems with high-rate streaming sensor data.

The rest of the paper is organized as follows. Section 2 reviews the related work on foundation models for time series and previous approaches in the field of online anomaly detection. Section 3 presents the proposed dual-learner architecture. Section 4 presents the case study used for validate the proposal. Finally, Section 5 summarizes conclusions and future work.

2. Related Work

2.1. Online Anomaly Detection in Data Streams

Online anomaly detection in data streams focuses on learning models incrementally while adapting to evolving data distributions. Among the most influential approaches are ensemble-based tree models designed for non-stationary data. Half-Space Trees (HST) [5] introduced a streaming extension of isolation-based anomaly detection, where random projections define partitions that adapt to new data without explicit retraining. Similarly, Robust Random Cut Forest (RRCF) [7] uses randomized partitioning trees to detect points that cause large changes in the model structure, emphasizing robustness to high-dimensional and noisy data streams. StreamRHF [8] extends this line by incorporating adaptive memory management to maintain bounded model size. More recently, streaming variants of Isolation Forest have been proposed. Online Isolation Forest (OIF) [9] updates isolation trees incrementally using reservoir-based subsampling to preserve model diversity, while Streaming Isolation Forest (SIF) [10] focuses on maintaining statistical representativeness under drift by integrating age-based sample replacement. These models provide fast predictive performance, but their learning dynamics can inadvertently learn the abnormal behavior as the concept of “normal” shifts over time.

Hybrid systems combining streaming learners with offline or historical models have also been explored. For example, the work of [11] presents a hybrid system where an offline model retains general characteristics (a bias) and an online model continuously learns and adapts. Other approaches [12] explore unsupervised online learning methods, which are inherently reactive but typically still need a "warm-up" or initial training phase, or rely on a dynamically adapting threshold (like quantile-based filters). Particularly in industrial settings, several approaches include periodically retraining autoencoders or clustering models on buffered data [13].

Overall, these systems typically rely on scheduled batch retraining or assume stable long-term distributions, limiting their responsiveness in complex industrial processes. In contrast, our approach integrates an online detector with a foundation model in a fully instance-based, confidence-driven routing scheme that requires no batching, no labels, and no hold-out reference windows.

2.2. Foundation Models for Time Series

Foundation models for time series have recently emerged as a parallel to large pretrained models in vision and language. MOMENT [6] demonstrated that transformer-based architectures trained on large-scale multivariate time series collections can be reused for reconstruction and forecasting across domains, supporting zero-shot and fine-tuned anomaly detection. MOMENT introduced chunk-based temporal tokenization to enable efficient learning of long-range dependencies while maintaining moderate computation cost. Subsequent work has expanded foundation models to improve efficiency and transferability. Tiny Time Mixers (TTM) [14] proposed a lightweight architecture based on TSMixer with adaptive patching and multi-resolution sampling. TTM models achieve strong zero- and few-shot performance while remaining computationally efficient enough for CPU-only deployments, addressing one of the main deployment barriers of foundation models. Likewise, Mantis [15] focuses on time series classification, employing a ViT-based encoder trained with contrastive learning and equipped with channel-adaptive adapters to reduce fine-tuning cost and improve calibration, highlighting the growing interest in generalizable, reusable time-series backbones.

Unlike forecasting- or classification-oriented foundation models, our use case requires instance-level anomaly reconstruction under streaming conditions. The proposed dual-learner framework leverages the high-level temporal priors encoded in MOMENT while retaining the flexibility and adaptability of online learners. To our knowledge, this is the first work to integrate a time-series foundation model directly within a real-time, unsupervised anomaly detection and routing mechanism.

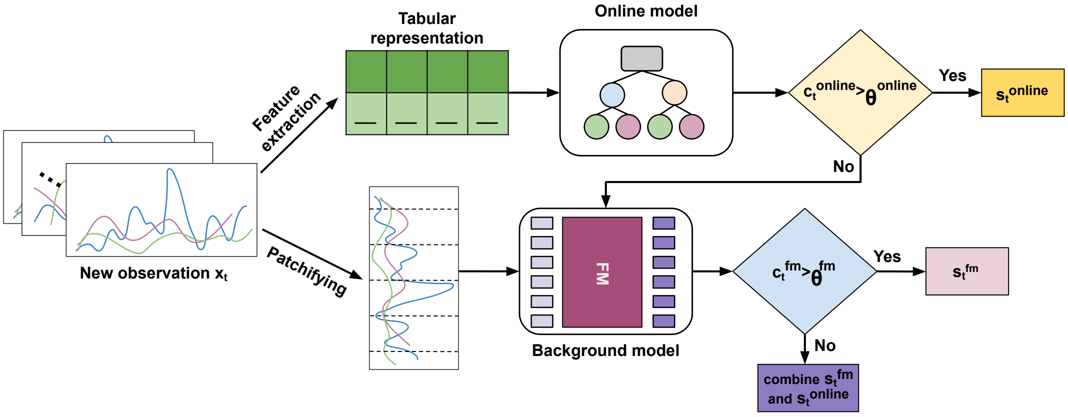

Figure 1: Overview of the Dual-Learner Architecture. At each time step t, a new observation x_t arrives. (1) The online learner produces an anomaly score s_t^{online} and confidence c_t^{online}. (2) If c_t^{online} exceeds its dynamic threshold θ^{online}, the system outputs s_t^{online} directly. (3) Otherwise, the instance is evaluated by the foundation model, which returns s_t^{fm} and c_t^{fm}. (4) If $c_t^{\text{fm}} > \theta^{\text{fm}}$, the system outputs s_t^{fm}; otherwise, the final score is obtained via confidence-weighted ensembling of s_t^{online} and s_t^{fm}.

3. Online Learning Supported by a Foundation Model for Anomaly Detection

We propose a dual-learner anomaly detection framework that integrates a fast online detector with a foundation model for time series as shown in Fig. 1. The system operates continuously on streaming data and adapts its prediction strategy based on model confidence. The architecture consists of three main components: (1) an online learner based on Half-Space Trees for real-time detection, (2) a background learner based on the MOMENT foundation model for deep representation-driven anomaly scoring, and (3) a confidence-based routing mechanism that dynamically selects or combines predictions from both learners.

We assume a streaming setting in which each incoming instance x_t is a multivariate time series of D dimensions and variable length L. To support both fast online processing and deep temporal modeling, the architecture uses two complementary representations: (i) a compact tabular descriptor derived from statistical and spectral features, and (ii) the raw or lightly preprocessed time-series input consumed directly by the foundation model. The processing pipeline for each new observation $x_t \in \mathbb{R}^{D\times L}$ is therefore:

$$\begin{aligned} x_t^{\text{tab}} &\leftarrow \text{statistical descriptors of } x_t \\ x_t^{\text{seq}} &\leftarrow x_t \text{padded to nearest 8 to match MOMENT's patch embedding} \end{aligned}$$

The proposed architecture is intentionally modular: each component addresses a specific limitation of either pure online methods or pure foundation-model approaches. The online learner provides fast streaming predictions; the foundation model contributes rich temporal priors; and the confidence router manages when to query which model under a fixed computational budget. This modularity does introduce several design choices (e.g., the online and background methods, the form of the confidence metric, or routing thresholds), which we motivate in the following subsections.

3.1. Online Learner: Half-Space Trees with Ensemble-Based Confidence

The online learner is implemented using HST [5], an incremental anomaly detection model designed for streaming environments. HST recursively partition the feature space into nested half-spaces, maintaining compact statistics that approximate data density in each region. Given an observation x_t^{tab}, each tree in the model outputs a score $s_t^{(i)}$, and the overall anomaly score is the average.

In this setting, we introduce a confidence metric that minimizes the computational load and maintains the original approach of unsupervised learning since does not require labels. Our confidence metric combines two complementary notions of agreement within the tree ensemble: (i) the variance of the raw anomaly scores, and (ii) the entropy of the induced binary anomaly votes. We use the score variance to capture *continuous* agreement: if all trees assign similar scores, we treat the prediction as reliable, regardless of whether the absolute score is high or low. However, relying on variance alone can be brittle when a few trees produce outlier scores due to local partitioning artifacts. To mitigate this, we also consider the entropy of binary votes obtained by thresholding each tree's score, since entropy is insensitive to the scale of the scores and instead measures the *robustness* of the decision: if most trees agree on either "normal" or "anomaly", the entropy is low and confidence is high. This term stabilizes the confidence estimate in regions where the variance may be inflated by a small subset of disagreeing trees. However, it is highly sensitive to the choice of the anomaly threshold, that here we set in 0.5.

We compute:

$$\text{Consist}_{\text{var}} = 1 - \frac{\sigma(s_t^{(i)})}{0.5} \tag{1}$$

to measures continuous agreement among trees. We also compute binary voting entropy by converting each score to a binary anomaly vote:

$$v_t^{(i)} = 1[s_t^{(i)} > \tau], p = \frac{1}{T}\sum_{i=1}^{T} v_t^{(i)}, \tag{2}$$

$$\text{Consist}_{\text{entr}} = 1 + (p \log_2 p + (1-p)\log_2(1-p)) \tag{3}$$

The final per-instance confidence $c_t^{\text{online}} \in [0,1]$ of the online model is obtained by averaging the two previous consistency metrics to preserve each one's benefits. Using only one of the terms would make the confidence overly sensitive either to outlier scores (variance only) or to the particular choice of the voting threshold (entropy only). Using the minimum of the two would be overly conservative in practice, leading the router to treat many instances as low-confidence and unnecessarily query the foundation model, increasing computational load without clear gains in detection quality.

Finally, since the online score is computed for all the instances in the data stream, we track reliability over time applying an Exponential Moving Average (EMA) approach:

$$c_t^{\text{online}} = \lambda \cdot c_{t-1}^{\text{online}} + (1-\lambda)\cdot \text{Consistency}_t \tag{4}$$

3.2. Background Learner: MOMENT Foundation Model

The background learner is MOMENT [6], a time-series foundation model pre-trained on large-scale multivariate sensor datasets. MOMENT consumes the raw time series x_t^{seq} (padded to length divisible by 8), producing a reconstruction $\hat{x}_t^{\text{seq}}$.

To estimate the confidence of the model, we apply perturbation-based analysis by injecting small Gaussian noise into $T-1$ perturbation copies of x_t^{seq}, producing T reconstructions in one parallel forward pass, which does not increment the temporal complexity of the model with respect to produce the reconstruction of the original time series. The confidence is obtained from the standard deviation across the reconstruction errors, with low reconstruction variability meaning high confidence:

$$c_t^{\text{fm}} = \exp(-\sigma(s_{t,1:T}^{\text{fm}})) \tag{5}$$

Optionally, MOMENT may be fine-tuned on unlabeled data from similar domains, improving domain alignment and enabling knowledge transfer to subsequent deployments.

3.3. Confidence Router with Online Beta Distribution Fitting

The router determines whether to trust the online learner (*fast path*), the foundation model (*slow path*), or a weighted ensemble. For each model, we maintain a running estimate of its confidence distribution using the method of moments to fit a Beta(α, β) distribution:

$$\mu = \mathbb{E}[c],\ \sigma^2 = \text{Var}(c),\ \alpha = \mu\left(\frac{\mu(1-\mu)}{\sigma^2} - 1\right), \tag{6}$$

$$\beta = (1-\mu)\left(\frac{\mu(1-\mu)}{\sigma^2} - 1\right) \tag{7}$$

From this Beta models, we compute a dynamic trust threshold for each model, θ^{online} and θ^{fm} respectively, as the p-th percentile:

$$\theta = \text{Beta}^{-1}(p; \alpha, \beta) \tag{8}$$

and selects predictions falling in the top-p percentile region of each model's own historical confidence range. This happens because the routing rule is to always first obtain the output from the online model (s_t^{online}, c_t^{online}), which has as little complexity as is $O(T \log d)$ for T trees. This confidence is evaluated against its percentile, $c_t^{\text{online}} > \theta^{\text{online}}$, meaning that it is in the top X% relative to that model's typical confidence. If this condition is true, the model's output is the online anomaly score s_t^{online}. Only when this condition is not met, the router queries the background model, and its confidence is evaluated against its percentile following the same principle: $c_t^{\text{fm}} > \theta^{\text{fm}}$. If this condition is true, the model's output is the background anomaly score s^{fm}. When none of the models achieve the desired percentile, the final score is obtained through confidence-weighted ensembling:

$$s_t = \alpha_t s_t^{\text{online}} + (1 - \alpha_t) s_t^{\text{fm}} \tag{9}$$

where α_t is adapted to the relative confidence of the two learners:

$$\alpha_t = \frac{c_t^{\text{online}}}{c_t^{\text{online}} + c_t^{\text{fm}}} \tag{10}$$

This makes the routing procedure scale-invariant across models and robust to drift; unsupervised since it does not depend on labels; and efficient as does not store historical windows and has constant memory and computation.

However, the efficiency of the model depends heavily on the configuration of these thresholds. In particular, the one associated to the online model, θ^{online}, that essentially controls how likely it is to consult the background model with each new observation. Therefore, this parameter should be set in relation to the speed of arrival of new instances, in a range that allows the model to minimize the calls to the background model and avoid bottlenecks. For example, setting $\theta^{\text{online}} = 0.3$ allows the model to trust the top 70% of the observed confidences.

4. Case Study: Industrial Hoist Installations

4.1. Problem Description

We evaluate the system on two distinct industrial elevator (hoist) installations in high-rise construction environments. Each instance corresponds to a full hoist ride, represented both as a multivariate time series and a derived feature vector for the online model. The details of each installation is presented in Table 1.

Across both installations, the average ride duration is approximately 25 seconds, but varies substantially with construction progress, as the hoist load, travel height, and acceleration phases change over time. Each ride provides two complementary data views, both of multivariate time series nature:

- Mechanical sensors related to vibration intensity and impulsiveness (v-RMS, a-Peak, a-RMS), crest factor (shock-dominated behavior) and operating temperature.
- Programmable Logic Controller (PLC) sensors that include motor electrical variables (current, torque, frequency, voltage), mechanical variables (position, speed, brake state), and drive and motor temperature estimates.

Table 1
Datasets information for the two case studies

	Installation A	Installation B
Time span	February 2023 to July 2023	December 2023 to May 2025
# rides	23,771	60,690
Ride duration	~ 23.8 seconds	~ 30.9 seconds
Sampling frequency	$20Hz \rightarrow\sim 471$ time-steps/ride	$20Hz \rightarrow\sim 617$ time-steps/ride
Data nature	Vibration measures (5 variables). PLC measures related to motor, position, speed, temperature... (12 variables).	
Anomalies ratio	15%	36%

This input is processed in two different ways by our Dual-Learner Architecture:

- Tabular descriptor for the online learner: The multivariate time series is summarized in statistical features (mean, std, min, max, iqr), time-domain features (rms, skewness, kurtosis), frequency-domain features (dominant, weighted average...), and trend features (trend slope and intercept). In total, the tabular representation of the series has 302 features.
- Raw multivariate time series for the background learner: The original sequence is injected for reconstruction in the foundation model almost as it is, preserving all temporal dynamics. However, because elevator rides are multivariate and nonstationary at startup and shutdown, we compute anomaly score using the central 80% of the ride:

$$s_t^{\text{fm}} = \frac{1}{0.8L} \sum_{k=\lfloor 0.1L \rfloor}^{\lfloor 0.9L \rfloor} \|x_{t,k} - \hat{x}_{t,k}\|^2 \tag{11}$$

where L is the ride length. This choice does not remove anomalous behavior, but instead addresses two practical issues: (i) MOMENT reconstructions exhibit systematic edge artifacts due to chunk-based patching and padding requirements, which inflate the reconstruction error at the beginning and end of each sequence independently of the true system state. (2) The physical hoist presents transient behavior during brake release, initial acceleration, and final deceleration. These short intervals show large but normal fluctuations in vibration and torque, which would dominate the reconstruction loss and lead to false positives without improving anomaly separability. Importantly, only the foundation-model score applies this trimming; the online learner continues to use a full representation of the ride. Thus, trimming reduces model-induced noise without discarding diagnostically relevant information.
After this, following MOMENT's chunk-based transformer architecture, the sequence is zero-padded to the nearest multiple of 8, and this extra portion is masked so it does not alter error-reconstruction metric. We configure this metric as the Mean Squared Error (MSE).

4.2. Experimental Settings

The proposed dual-learner system is implemented in the CapyMOA streaming analytics framework [16], and integrates the MOMENT-base implementation provided by the authors [6]. The experiments are evaluated in a fully unsupervised, prequential (test-then-train) setting. Each ride is evaluated when it arrives, producing an anomaly score before any model update. The true label, when available for offline inspection, is not used for training or thresholding. This evaluation protocol matches real deployment conditions, where anomalies are rare, labels are delayed, and models must adapt online.

We compare our method against widely used streaming anomaly detectors available in CapyMOA:

- Half-Space Trees (HST) [5]
- Robust Random Cut Forest (RRCF) [7]

Table 2
Accumulated results in the prequential evaluation

Model	Installation A				Installation B			
	AUC-PR	AUC-ROC	F1	Time	AUC-PR	AUC-ROC	F1	Time
HST	0.1914	0.5987	0.2567	**1.67**	0.3501	0.4698	0.5338	**3.65**
RRFC	0.1606	0.5016	0.2567	1007.99	**0.3655**	0.4953	0.5338	2470.41
StreamRHF	0.1408	0.4659	0.1690	61226.84	0.3610	0.4916	0.2824	162661.08
OIF	0.1165	0.3998	0.0028	494.68	0.2790	0.3516	0.0000	1252.37
SIF	0.1350	0.3966	0.0657	14.14	0.3561	0.4821	0.1354	30.07
DAD$_{\text{zs}}$ (ours)	**0.1933**	**0.6148**	**0.2591**	180.88	0.3454	0.4670	**0.5345**	702.77
DAD$_{\text{ft}}$ (ours)	-	-	-	-	0.3591	**0.5018**	0.5343	1294.28

Best results are in bold.

- StreamRHF [8]
- Online Isolation Forest (OIF) [9]
- Streaming Isolation Forest (SIF) [10]

No per-dataset hyperparameter optimization is performed. Instead, we use default configurations and unify parameters where applicable (e.g., number of trees=25 in all cases) to ensure a fair and deployment-realistic comparison. For our approach, the parameters related to HST and to MOMENT stay the same as the ones proposed by the authors. Apart from these, the only relevant parameters are the mentioned percentile-based confidence thresholds, that are set to $\theta^{\text{online}} = 0.1$ and $\theta^{\text{fm}} = 0.3$, and a warm start parameter that controls that for the first $n = 100$ observations, the foundation model is used until the online learner has built a stable model.

4.3. Evaluating Scenarios

Performance is assessed using offline ground-truth inspection logs and event reports from maintenance records, but these are never used during learning. Metrics reported include both anomaly ranking quality (AUC-PR, AUC-ROC) and alarm efficiency (F1-score under fixed anomaly threshold at 0.5)

We evaluate our approach Dual Anomaly Detection (DAD) on two modes:

1. **Zero-shot:** The foundation model is used without adaptation. We refer to this as DAD$_{\text{zs}}$
2. **Cross-Domain Fine-Tuning:** The foundation model is fine-tuned on Installation A and tested to Installation B. We refer to this as DAD$_{\text{ft}}$

This setting tests whether knowledge gained from one hoist installation can improve anomaly detection in another—an important capability for scalable industrial monitoring systems.

4.4. Experimental Results

Table 2 summarizes the performance of all methods across both hoist installations. Several trends highlight the strengths of the proposed dual-learner system and the role of cross-installation adaptation. First, the zero-shot configuration DADzs achieves competitive performance on both installations, matching or surpassing the best-performing baselines. On Installation A, it provides the highest AUC-PR (0.1933) and AUC-ROC (0.6148), with a slight improvement in F1-score over the nearest alternative (HST). On Installation B, DADzs achieves performance comparable to the best baseline configurations. This suggests that the representation and reconstruction mechanisms of MOMENT in combination with the online learner generalize to new data distributions without explicit model tuning.

However, the benefits of cross-domain fine-tuning are especially visible in the second installation. DAD$_{\text{ft}}$, which incorporates adaptation from Installation A to Installation B, improves both AUC-PR (from

0.3454 to 0.3591) and AUC-ROC (from 0.4670 to 0.5018). This improvement is especially relevant because Installation B has a higher anomaly ratio (36% vs. 15%): The presence of more operational irregularities increases the variance of the reconstruction error distributions, making threshold selection more challenging for simpler detectors, but fine-tuning on normal data from previous projects recalibrates the reconstruction baseline toward the characteristics of the new system, improving separability of anomalous behavior. This supports the hypothesis that foundation models for industrial time-series benefit from lightweight adaptation rather than from being applied in a completely zero-shot manner.

In terms of execution time, methods such as HST and SIF remain the fastest, making them suitable for lower-resource deployments. However, their performance is consistently lower, particularly in Installation A, where they present limited ability to capture subtle deviations. In contrast, classical random-cut forest methods (RRCF, StreamRHF) show significantly higher computational cost (up to 160k seconds in the largest setting), making them impractical for real-time monitoring. DAD_{zs} and DAD_{ft} fall between these two extremes. While their execution times (55–180 seconds) are higher than the simplest models, they remain easily compatible with real-time operation, since processing a ride occurs after acquisition and well within cycle times of industrial maintenance systems. Crucially, the added computational cost translates into improved modeling of temporal dynamics and more accurate anomaly scoring.

The comparison between installations underlines the differences in data complexity. Installation B has nearly three times more rides and more anomalies, both due to more prolonged operation and evolving mechanical wear during the later construction phase. Models that rely solely on online learning can struggle to maintain precision as the normal behavior drifts. The dual-learner architecture mitigates this through continual online adaptation while still preserving a stable baseline representation via the foundation model.

Finally, it is important to note that the ground-truth labels used for evaluation in this real-world case study derive from maintenance reports and operator annotations, which do not always correspond to precise or isolated temporal events. In many cases, an anomaly label may describe a period of degraded performance spanning multiple rides rather than a sharply defined instance. This misalignment between real anomaly manifestation and discrete ride-level labels introduces uncertainty into the calculated metrics. Improving label granularity remains an open challenge in industrial predictive maintenance and is a direction for future work.

4.5. Model Behavior across the Data Stream

Figure 2 illustrates the dynamics of the dual-learner system during the full prequential evaluation of Installation A. The top panel shows the ground-truth anomalous periods, which appear in concentrated bursts corresponding to well-documented mechanical degradation phases and maintenance interventions, together with the anomaly score produced by DAD, which closely follows changes in system behavior and increases during annotated anomalous intervals, indicating effective sensitivity to deteriorating ride dynamics. Equivalently, Figure 3 shows the dynamics of the model for Installation B.

The most informative aspect is the second panel, which shows the model selection mechanism: whether the final anomaly score arises from the background model (0), the online model (2), or their ensemble combination (1). In particular, for Installation A there are 22,877 instances scored by the online model, 690 scored by the background model and 204 scored by the ensemble, In Installation B we have 51,119, 137 and 9,434, respectively. In both installations, at the beginning of the stream, the system relies almost exclusively on the background model, since the online one may be affected by the cool start problem. As the hoist behavior gradually evolves due to increased load, wear and environmental conditions, the online model becomes more influential, reflecting the increasing need for local adaptation. The switching pattern is neither abrupt nor purely periodic, but instead responds to changes in the statistical consistency of incoming data. Meanwhile, the online model confidence (third panel) starts relatively high but slowly decreases as operational variability grows, before stabilizing in a regime where both learners contribute. This coordinated interplay demonstrates the intended function of the dual-learner architecture: the background model provides robust performance to prevent overreacting

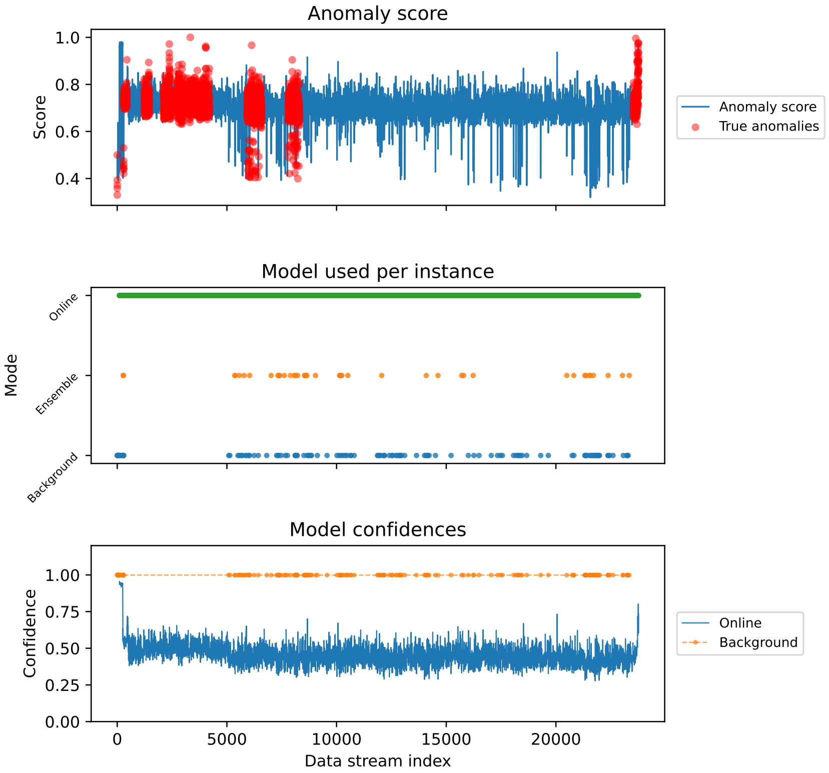

Figure 2: DAD_{zs} behavior in Installation A: real anomalies, final anomaly score, from which component this score has being obtained, and the confidence of the online and the background model, respectively

to early noise, while the online model specializes to installation-specific dynamics and provides fast predictions. When anomalous periods occur, the online learner's confidence drops, emphasizing that these rides deviate from learned local behavior and possible also from the broader expected operational patterns encoded by the foundation model, as we can see in Installation B, where the model recurs to the ensemble a considerable number of times. In this case, looking at the high confidences that the background model exhibits most of the times, this can be due to the finetuning on the previous installation, that is overfitting the model and making it less effective on new anomalies from the second installation.

5. Conclusions

This work introduced a dual-learner architecture for unsupervised anomaly detection in industrial streaming environments. The system integrates a fast online model with a time-series foundation model through a confidence-based routing mechanism that adapts dynamically to changes in data distribution. In this way, the online learner and the foundation model carry complementary strengths: the former adapts quickly to local conditions, while the latter provides a stable representation of normal operation learned from large-scale data. The confidence estimation and routing strategy allows the system to balance these strengths without requiring labels or maintain a historical buffer. Experiments on two industrial elevator installations showed that the proposed approach performs competitively in a fully unsupervised and prequential evaluation. The zero-shot configuration demonstrates strong generalization to new deployments, while fine-tuning the foundation model on one installation and transferring it to another yields further improvements. These findings confirm that foundation models can act as reusable priors for industrial streaming tasks, reducing the need for manual configuration and prior training data.

While still a preliminary work, this study suggests that integrating online learners with foundation models is a viable path toward scalable and reliable industrial data stream systems. However, there is

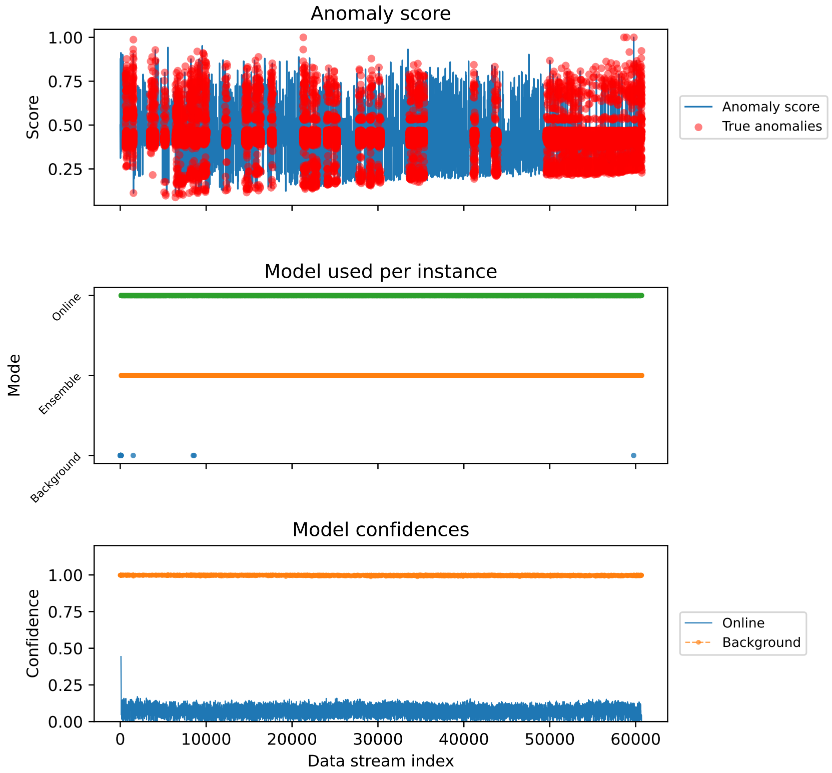

Figure 3: DAD$_{\text{ft}}$ behavior in Installation B: real anomalies, final anomaly score, from which component this score has being obtained, and the confidence of the online and the background model, respectively

still room for improvement in terms of validation assessment across wider domains. Future work will explore integration with other promising architectures for time series analysis, like diffusion models, active learning strategies for selective labeling under minimal supervision, and broader cross-domain transfer across different domains and types of machinery.

Acknowledgments

This project is funded by the Swedish Knowledge Foundation (KKs).

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and DeepL in order to: Grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] V. Rodriguez-Fernandez, D. Camacho, Recent trends and advances in machine learning challenges and applications for industry 4.0, Expert Systems 41 (2024) e13506.

[2] L. Correia, J.-C. Goos, P. Klein, T. Bäck, A. V. Kononova, Online model-based anomaly detection in multivariate time series: Taxonomy, survey, research challenges and future directions, Engineering Applications of Artificial Intelligence 138 (2024) 109323. doi:10.1016/j.engappai.2024.109323.

[3] W. Hao, T. Yang, Q. Yang, Hybrid statistical-machine learning for real-time anomaly detection in

industrial cyber–physical systems, IEEE Transactions on Automation Science and Engineering 20 (2021) 32–46.

[4] D. Kahneman, Thinking, fast and slow, Macmillan, 2011.

[5] S. C. Tan, K. M. Ting, T. F. Liu, Fast anomaly detection for streaming data, in: 22nd International Joint Conference on Artificial Intelligence, 2011, pp. 1511–1516. doi:10.5591/978-1-57735-516-8/IJCAI11-254.

[6] M. Goswami, K. Szafer, A. Choudhry, Y. Cai, S. Li, A. Dubrawski, Moment: A family of open time-series foundation models, in: 41st International Conference on Machine Learning (ICML 2024), 2024, pp. 1–16.

[7] S. Guha, N. Mishra, G. Roy, O. Schrijvers, Robust random cut forest based anomaly detection on streams, in: 33rd International Conference on Machine Learning (ICML 2016), 2016, pp. 3987–3999.

[8] S. Nesic, A. Putina, M. Bahri, A. Huet, J. M. Navarro, D. Rossi, M. Sozio, Streamrhf: Tree-based unsupervised anomaly detection for data streams, in: IEEE/ACS International Conference on Computer Systems and Applications, AICCSA, volume 2022-December, 2022, pp. 1–8. doi:10.1109/AICCSA56895.2022.10017876.

[9] F. Leveni, G. W. Cassales, B. Pfahringer, A. Bifet, G. Boracchi, Online isolation forest, in: 41st International Conference on Machine Learning (ICML 2024), 2024, pp. 27288–27298.

[10] J. J. Liu, G. W. Cassales, F. T. Liu, B. Pfahringer, A. Bifet, Streaming isolation forest, in: Advances in Knowledge Discovery and Data Mining, 2025, pp. 95–107. doi:10.1007/978-981-96-8170-9_8,.

[11] M. Odiathevar, W. K. G. Seah, M. Frean, A hybrid online offline system for network anomaly detection, in: 2019 28th International Conference on Computer Communication and Networks (ICCCN), 2019, pp. 1–9. doi:10.1109/ICCCN.2019.8847011.

[12] L. Cerdà-Alabern, G. Iuhasz, G. Gemmi, Anomaly detection for fault detection in wireless community networks using machine learning, Computer Communications 202 (2023) 191–203. doi:10.1016/j.comcom.2023.02.019.

[13] M. Nsor, Predictive maintenance using machine learning for engineering systems through real-time sensor data and anomaly detection models, International Journal of Research Publication and Reviews 6 (2025) 5167–5183. doi:10.55248/gengpi.6.0725.2541.

[14] V. Ekambaram, A. Jati, P. Dayama, S. Mukherjee, N. H. Nguyen, W. M. Gifford, C. Reddy, J. Kalagnanam, Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series, in: 38th Conference on Neural Information Processing Systems (NeurIPS 2024), 2024.

[15] V. Feofanov, S. Wen, M. Alonso, R. Ilbert, H. Guo, M. Tiomoko, L. Pan, J. Zhang, I. Redko, Mantis: Lightweight calibrated foundation model for user-friendly time series classification, in: 1st ICML Workshop on Foundation Models for Structured Data FMSD @ ICML 2025, 2025, pp. 1–21.

[16] H. M. Gomes, A. Lee, N. Gunasekara, Y. Sun, G. W. Cassales, J. J. Liu, M. Heyden, V. Cerqueira, M. Bahri, Y. S. Koh, B. Pfahringer, A. Bifet, CapyMOA: Efficient machine learning for data streams in python, 2025. URL: https://arxiv.org/abs/2502.07432. arXiv:2502.07432.