

Review

Not peer-reviewed version

Continual Anomaly Detection: A Comprehensive Survey and Research Roadmap

[Kamil Faber](#)*, Christopher Kanan, Vincenzo Lomonaco, [Roberto Corizzo](#)*

Posted Date: 26 January 2026

doi: 10.20944/preprints202601.1931.v1

Keywords: continual anomaly detection; continual learning; lifelong learning; anomaly detection; incremental learning



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Continual Anomaly Detection: A Comprehensive Survey and Research Roadmap

Kamil Faber^{1,*}, Christopher Kanan², Vincenzo Lomonaco³ and Roberto Corizzo^{4,*}

¹ AGH University of Krakow, Poland

² University of Rochester, Rochester, New York, USA

³ LUISS University, Rome, Italy

⁴ American University, Washington DC, USA

* Correspondence: kfaber@agh.edu.pl (K.F.); rcorizzo@american.edu (R.C.)

Abstract

Continual Anomaly Detection (CAD) addresses the challenge of identifying abnormal patterns in evolving data by integrating adaptation to new conditions with knowledge retention. Since real-world applications deal with dynamic and non-stationary environments, anomaly detection models must continuously evolve to remain effective. In this context, CAD, sometimes also referred to as lifelong anomaly detection, has recently emerged as a distinct research area at the intersection of Continual Learning (CL) and Anomaly Detection (AD). This survey discusses: i) CAD learning scenarios and the assumptions they impose, ii) methods for adaptation and retention, iii) application domains, and iv) evaluation metrics that capture both detection quality and continual dynamics. We identify key gaps in current research, including limitations in current approaches, scenarios, evaluation, and alignment with real-world conditions. Finally, we discuss open challenges and propose a roadmap to guide future research toward robust anomaly detection systems suitable for evolving real-world environments.

Keywords: continual anomaly detection; continual learning; lifelong learning; anomaly detection; incremental learning

1. Introduction

Anomaly Detection (AD) aims to identify deviations from normality in data, often operating in unsupervised or semi-supervised conditions, with very limited access to anomalous data [1]. In real-world applications, such as industrial monitoring, autonomous systems, and cybersecurity, the definition of *normality* is non-stationary and evolves due to factors such as new operating conditions, changes in users' behaviors, and seasonal effects.

Two commonly adopted solutions are either periodically retraining from scratch or naively fine-tuning the model. However, periodic retraining is expensive and may be infeasible under privacy or resources constraints, and naive online updates can overwrite useful knowledge inducing catastrophic forgetting. This tension motivates *Continual Anomaly Detection (CAD)* methods that adapt to evolving normality while retaining knowledge needed for long-horizon reliability [2].

CAD incorporates methods from general *Continual Learning (CL)*, which develops mechanisms for mitigating forgetting across sequences of tasks. However, while most CL work targets supervised classification with clean task structure [3], CAD differs from CL in several ways: 1) CAD is frequently unsupervised or semi-supervised, 2) CAD includes evolving definition of "normal" class while anomalies may change independently, and 3) Task boundaries or identities are often unavailable in CAD [4]. Consequently, CAD is not merely CL applied to AD. It is a distinct problem family with its own scenario assumptions, method-specific challenges, and operational constraints.

Despite the increasing interest on this subject, the research landscape is fragmented. This survey aims to make the field easier to navigate and to push it toward a more realistic and reproducible evaluation. We provide the first survey that gives a comprehensive and critical overview of CAD,

emphasizing that the “right” approach depends as much on scenario assumptions and operational constraints as on the algorithmic choices used to control forgetting.

Our survey is organized as follows:

1. We define the CAD problem and compare it to related settings (Section 2).
2. We discuss CAD learning scenarios and clarify the assumptions that most affect method choices (Section 3).
3. We survey CAD algorithms and methods, emphasizing design trade-offs and recurring patterns (Section 4).
4. We review CAD applications across data modalities and highlight domain-specific constraints (Section 5).
5. We analyze evaluation metrics that capture both detection quality and continual dynamics (Section 6).
6. We summarize open challenges and provide a roadmap toward robust and effective CAD systems (Section 7).

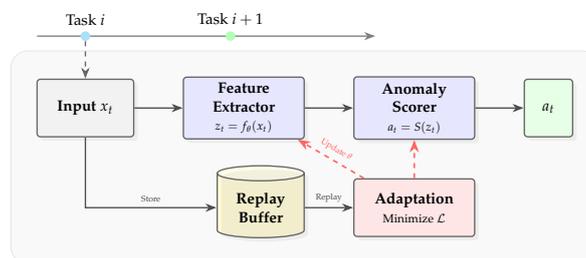


Figure 1. Example of a CAD workflow for replay-based strategy. The system processes a non-stationary sequence of tasks (top). When a new task is presented, an update is triggered and the *Adaptation* module retrieves past data from the *Replay Buffer* to update the model parameters, balancing knowledge retention with adaptation.

2. Background and Problem Definition

2.1. Related Settings and Boundaries

CAD methods often incorporate aspects from other areas concerned with learning from non-stationary data streams and with detecting anomalous inputs.

AD, OOD, and Open-Set Detection

Classical AD focuses on identifying deviations from an assumed notion of normality, while OOD detection and open-set recognition focus on rejecting samples from distributions or classes not seen during training [5]. CAD often uses similar scoring ideas, but adds a hard requirement that the detector itself must be updated as normality changes over time. In other words, the challenge is not only to flag unusual inputs, but to keep the notion of “usual” from going stale.

Concept Drift and Change-Point Detection

Concept drift and change-point detection aim to detect distributional shifts and can be used to trigger model updates. Triggers are useful in CAD, but they do not specify *how* to update safely under scarce supervision and possible contamination. As a result, CAD typically combines drift or change signals with retention mechanisms (e.g., replay or structured memory) to stabilize adaptation over long horizons [6].

Periodic Retraining

A straightforward strategy for updating an AD system is to periodically retrain it on an updated dataset that includes all accumulated data [4]. However, the model will degrade in performance between retraining phases, requiring frequent retraining, which can be expensive and may be infeasible under storage or privacy constraints. Even when retraining is possible, it can quietly inherit data

quality issues (e.g., contaminated “normal” logs) and provides no guarantees about retaining useful historical behavior without explicit retention mechanisms.

Online Learning

While online learning is present in anomaly detection [1], it is used only for adapting to incoming data without addressing catastrophic forgetting that can occur in neural networks. While these methods overcome some of the weaknesses of periodic retraining, they still require frequent and often expensive updates, while suffering from forgetting. CAD differs by treating retention and controlled adaptation as first-class design goals rather than side effects of continual updating.

2.2. Problem Definition

We model CAD as learning from a sequence of tasks $S = s_1, s_2, \dots, s_N$ coming from an evolving distribution \mathcal{D} . Each time a new task s_i is presented, the model should learn it without losing its knowledge of previously observed tasks. For inference, the system outputs an anomaly score $a_t = f_{\theta_t}(x_t) \in \mathbb{R}$ and optionally a binary decision $\hat{y}_t \in \{0, 1\}$ based on a threshold. Supervision may be absent, partial, or delayed: the true label y_t (normal vs anomaly, or anomaly type) may be unknown at decision time, may arrive later, or may only be available for a small subset of samples. The key difficulties in this setting may include:

- **Evolving normality:** the parameters of \mathcal{D} change over time requiring a model to adapt.
- **Evolving anomalies:** anomalies may appear, disappear, or change type independently from the evolving normality, their frequency is typically low and may be adversarial.
- **Lack of information on task changes:** Task boundaries and identities often unavailable during training or inference.

Informally, the goal of CAD is to maintain high detection quality over long horizons while adapting to distribution shift. This typically involves balancing multiple objectives: i) **detection performance** on current and past regimes (tasks), ii) **retention** of useful historical knowledge without overfitting to stale patterns, iii) **adaptation speed** to new regimes (including deciding when to update), and iv) **resource constraints** such as memory, compute, latency, and energy.

3. CAD Learning Scenarios

CAD results are often driven as much by the *scenario assumptions* as by the learning algorithm. In particular, current works vary in: i) what constitutes a “task” or *regime* of normality; and ii) what metadata the learner receives about regime changes.

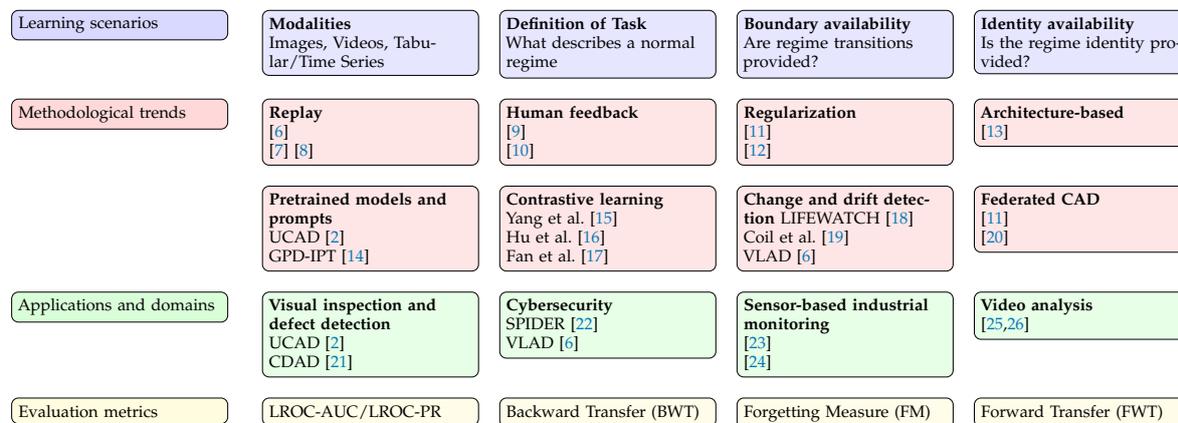


Figure 2. Overview of Continual Anomaly Detection (CAD): Learning scenarios, Methodological trends, Application and domains, and Evaluation metrics.

3.1. Modality-Driven Scenarios

Images

In the visual domain, CAD has been predominantly explored for detection of anomalies in images, often in industrial inspection settings, where new products or object types are introduced over time as new tasks. One of the early studies by Li et al. [27], frames CAD as a *domain-incremental* problem, arguing that all tasks involve only two classes (normal and anomaly) that evolve across all learning phases. They further introduce the notion of a category as a set of visually similar normal products of the same type. Many benchmarks instantiate regimes by grouping one or more categories per phase, then updating the detector sequentially as categories arrive. Zhang et al. [14] defines CAD as *one-for-more* scenario, in which a single model is incrementally updated to handle multiple products, contrasting it with two traditional approaches: *one-for-one* (one model per one product) and *one-for-all* (one model for all products with a single training).

Videos

An alternative perspective on task definition emerges in the video continual anomaly detection, where the primary objective is to identify deviations from typical spatiotemporal patterns called *activities*, such as playing basketball or pedestrian moving in crowded scenes [26]. A task may correspond to a scene/camera, a set of activities, or a time segment in which dynamics are relatively stable. Unlike activity-recognition settings, the model is not required to explicitly label the underlying activity.

Tabular and Time Series

While tasks in image or video domains often align with natural categories, tabular data typically lacks this structure. Consequently, CAD research tends to emphasize the evolution of the underlying distribution of the normal class that corresponds to changes in the environment, for example, the addition of a new server in intrusion detection or starting a new activity in sensor-based human activity monitoring. Such self-consistent behaviors of the normal class are regarded as *concepts* [4], which recall the notion of *tasks* in conventional CL. For time-series data, it is possible to define regimes via time-based splits where, following the natural chronological order of the data, each task is defined as a contiguous time frame [8]. This approach assumes that changes in the underlying data distributions occur over time, making temporal segmentation a reasonable proxy for task boundaries.

3.2. Recommendations for Scenario Specification

Across modalities, CAD scenarios often share the same high-level pattern: the learner must model an evolving normal class while maintaining reliable detection under scarce or delayed supervision. However, labeling a setting as simply “domain-incremental” is insufficient. A practical issue is that terminology is not used consistently. For example, Zhang et al. [14] use *task-aware* for settings with boundaries known during training but not necessarily at inference, whereas Faber et al. [4] use *concept-aware* for settings where boundaries and identities are available. These mismatches complicate comparison and replication.

To reduce ambiguity and avoid overloading CL terminology, we recommend describing CAD learning scenarios along three orthogonal dimensions showcased in Table 1 (separately for training and inference when they differ). These dimensions are typically sufficient to reconstruct the intended experimental setup and to know whether comparisons are meaningful. This framing is consistent with widely used CL scenario taxonomies (task-/class-/domain-incremental) [28], but avoids forcing CAD problems into labels that can hide crucial details (e.g., whether boundaries or identities are available). Moreover, it keeps scenario descriptions compact and makes it clear which algorithms are compatible with the assumed setting.

4. CAD Algorithms & Methods

CAD methods are built around a few recurring moves: decide *when* to adapt, decide *what* to preserve, and keep updates from turning yesterday’s “normal” into tomorrow’s anomaly (Table 1). Below we group the literature into compact method families that show up repeatedly across modalities and scenarios.

Table 1. A minimal scenario specification for CAD experiments. We recommend reporting each dimension explicitly (and separately for training vs inference when applicable).

Dimension	What to report
Task (regime of normality)	What defines a regime of “normal”: e.g., product category (images), activity/scene (video), concept/operating mode (tabular), or time window (time series).
Boundaries availability	Are regime transitions given (train/inference), partially given (only train), or not given (the stream is task-free)?
Identities availability	If boundaries exist, is the regime identity provided (train/inference), or must it be inferred/ignored?

4.1. Replay

Replay is a strong baseline in which the learner maintains a buffer of past samples and combines them with incoming data to reduce forgetting [28]. In CAD, replay is also easy to reason about and debug, which is why many works either build on it or benchmark against it. As streams grow longer, the buffer itself needs structure to remain representative. Hierarchical memory is one way to do this: VLAD [6] and CPDGA [29] use change-detection signals to form hierarchical concept groupings that support stronger retention and inference under recurring regimes.

However, the practicality of the replay is governed by resource constraints. To this end, Pezze et al. [30] explore the idea of compressed-replay for vision-based CAD. A related idea from CL is *latent replay*, which stores intermediate embeddings that can often be aggressively quantized with minimal degradation [7], improving memory efficiency without redesigning the learner. Even when storage is not the bottleneck (as in many industrial settings), replay can still be compute-limited because it increases the amount of training required per update.

Another approach to reduce storage is *generative replay*, which replaces stored data with a generator trained to sample approximations of earlier regimes. In multivariate time series, GenDex uses a DC-VAE backbone to compress past patterns into a generator and regenerate prior task data [8]. Generative replay can help with privacy and memory, but it can also drift on its own: a generator that forgets, collapses, or biases its samples can quietly poison the learner [30]. As a result, despite its promise, generative replay remains relatively underexplored in CAD compared to conventional replay.

4.2. Regularization and Knowledge Distillation

Regularization aims to stabilize learning without relying on stored data. CAD evaluates classical regularizers, such as EWC and LwF, in discrete manufacturing and phishing detection [31,32]. Moreover, such methods often serve as baselines for more advanced methods [11]. Newer approaches often tailor the regularization to the specific data type or domain. For example, Tang et al. [33] propose compact feature regularization for images in small defect inspection. A different attempt at stabilization of weight updates without storing data is to introduce knowledge distillation. For example Guo et al. [12] combine clustering-based selection with multi-level distillation, while Yu and Wang [34] apply teacher-student distillation to retain prior knowledge while learning new categories.

4.3. Architectural Strategies

Architectural strategies reduce interference by changing *where* plasticity lives. A common approach is to allocate capacity across regimes, often via expert-style specialization that is conceptually similar to Mixture-of-Experts (MoE) [35]. In CAD this frequently takes the form of creating one expert per task with dynamic routing. Cai et al. [13] propose a dynamic expert router for unsupervised

CAD, and Dahmardeh and Setti [36] assign experts based on feature similarity with explicit memory management. Hybrid designs that combine architectural decoupling with prompt-based adaptation are also emerging [14]. These approaches can be effective, but they raise practical questions about how many experts we can create in a single scenario.

Architectures also help in the presence of conflicting objectives. Harun et al. [37] show a consistent trade-off, where stronger neural collapse in a layer tends to improve anomaly detection, while mitigating collapse improves forward transfer, suggesting that a single feature space is unlikely to optimize both goals. This observation motivates explicitly decoupled designs (e.g., separate heads or embedding spaces at different layers), aligning naturally with multi-branch, multi-head, and expert-based CAD systems [37].

4.4. Human-in-the-Loop

Real deployments rarely run without any supervision forever, and CAD increasingly acknowledges this fact. Human feedback can be treated as a scarce but valuable resource that can guide adaptation, mitigate error accumulation, and stabilize learning over long time horizons. A representative example is the incorporation of expert feedback to reduce contamination in replay buffers in settings, where anomalies in training can amplify performance degradation [9]. Closely related is the challenge of data collection and curation [38], as this problem could become the first-class design problem in real continual settings. Beyond algorithm design, it is important to account for human involvement when designing evaluation protocols [10]. Standard metrics and benchmarks often implicitly assume unrealistic levels of automation, overlooking the limited capacity of human collaborators.

4.5. Pretrained Models and Prompts

Pretrained and foundation models mitigate the burden of model updates. By freezing a strong backbone and adapting only small components, CAD can reduce both training time and forgetting pressure. At the same time, prompts and adapters provide lightweight, task-specific customization on top of frozen features. UCAD [2] and GPD-IPT [14] are prominent examples: both keep the backbone fixed while learning small trainable prompt vectors that specialize the model to the current regime.

A key advantage is efficiency, but an even bigger benefit is control: prompts localize plasticity, making it easier to adapt without destabilizing the representation. Extensions go beyond vision-only signals, for example, by using multimodal prompts that capture both visual and textual patterns to build richer descriptions of normality across tasks [39]. However, using pre-trained models requires careful study designs to prevent confounding results (see Section 7).

4.6. Contrastive Learning

Contrastive learning has recently emerged in CAD, as it offers more robust representation learning with minimal access to or a complete lack of labeled anomalies. Methods typically rely on self-supervised learning to extract and align representations of normal data across evolving tasks. For example, Yang et al. [15] leverage contrastive learning in combination with experience replay to identify the most salient features of normal data and constrain the updates in proximity to previous tasks. Similarly, Hu et al. [16] apply cross-condition feature contrast to enhance adaptability while mitigating forgetting, and Fan et al. [17] combine mean-shifted contrastive loss with teacher-student architecture to optimize the knowledge distillation process. Notably, this line of work is still heavily vision-centric, with a lack of methods designed for tabular and time series CAD.

4.7. Change and Drift Detection

As we pointed out in Section 3, in many real-world domains and applications there is no explicit knowledge about task boundaries or task identities. In such scenarios, CAD methods need a trigger for model adaptation and change point detection methods increasingly serve as such control signal. To address change point detection with task recurrence, Faber et al. [18] propose LIFEWATCH, a Wasserstein-based method that monitors distributional shifts in data streams and distinguishes be-

tween novel tasks and recurring ones. Coil et al. [19] provide a comprehensive analysis of the efficiency and detection performance of CAD with WATCH-based change point detection adopting different distance measures. Methods such as VLAD [6] and CPDGA [29] take advantage of statistical change point detection methods integrating them with model updates and knowledge retention strategies. Other approaches such as CITADEL [40] integrate drift detection through a KS-test between new data and buffer statistics to decide when to adapt.

4.8. Federated CAD

A noticeable trend in CAD is leveraging federated learning for scenarios involving decentralized data sources, such as the IoT network, edge devices, and distributed monitoring systems. One example is [11], where the authors utilize multiple standard federated learning approaches (e.g., FedAvg, FedProx) in combination with experience replay and regularization to identify DDoS attacks within distributed IoT systems. Similarly, [20] shows how federated learning can be included in continual unsupervised methods for network anomaly detection. As this trend is still emerging, there are still multiple challenges open for federated learning in CAD, including personalization under heterogeneous notions of normality and the absence of native CAD methods that jointly address continual adaptation, multi-source learning, and privacy.

5. Application and Domain Trends

Here, we discuss some of the most prevalent applications of CAD and recent trends.

Visual Inspection for Manufacturing

One of the most prominent and active application domains for continual anomaly detection is industrial inspection and manufacturing. In this context, visual inspection models must identify defects in product images, either at the image level (classifying entire samples as normal or anomalous) or at the pixel level (localizing defective regions) [41]. As new products are introduced over time, anomaly detection systems must adapt to novel inspection tasks while preserving detection capabilities for previously seen products. Continual anomaly detection setting in this context is also sometimes referred to as one-for-more anomaly detection or One-Model-N-Objects with incremental training. Notable methods include UCAD [2], CDAD [21], and IUF [33]. It should be noted that the predominant dataset in this domain is MVTec-AD, which has become the benchmark for the evaluation of industrial visual anomaly detection methods.

Cybersecurity Applications

Cybersecurity is a key application domain for continual anomaly detection (CAD) due to the highly dynamic and evolving nature of digital systems, in which traditional static anomaly detection methods are often insufficient. Consequently, CAD is essential for enabling adaptive detection without frequent retraining or manual intervention. The dominant application of CAD in cybersecurity is in intrusion detection systems (IDS), where legitimate network behavior can change due to multiple factors such as software updates, changes in user activity, or the adoption of new communication protocols. CAD works, such as SPIDER [22] and VLAD [6], incrementally learn from new traffic patterns while retaining knowledge of previously seen behaviors. CAD has also seen significant adoption in the context of security for the Internet of Things (IoT). Given the heterogeneity and volatility of IoT environments, where new devices, firmware updates, and communication protocols are frequently introduced, many CAD methods, such as CITADEL [40] and the work by Benameur et al. [11], have shown the benefits of continual learning. Beyond intrusion detection and IoT security, CAD has also been explored in malware detection [42] and financial fraud detection [10].

Sensor-Based Industrial Monitoring

Sensor-based industrial monitoring is a growing subdomain of CAD, in which time-series data from physical systems must be monitored in real time under evolving conditions. In contrast to

vision-based inspection, these applications focus on detecting deviations in sensor patterns that may indicate faults, inefficiencies, or irregular operational states. For example, [43] investigates anomaly detection in appliance-level power consumption, where device energy usage patterns evolve over time due to device degradation over time, changes in user behavior, or in the operational context. [23] focuses on high-speed rotating machinery, leveraging continual learning with streaming sensor data to address the non-stationarity of sensor readings and effectively detect early signs of degradation or malfunctions across varying operating conditions. A different but related use case is leveraging IoT-enabled continual anomaly detection in dairy farming to monitor the health and behavior of livestock [24].

Video Analysis

Continual anomaly detection has also gained attention in video analysis, particularly in applications where the environment and normal activity patterns evolve over time. The most prominent use case is video surveillance, where systems must monitor dynamic public spaces and detect abnormal behaviors or events in real time. CAD has been shown to improve anomaly detection in CCTV footage by enabling models to adapt to new scenes or behavior patterns without forgetting previous ones [25,26].

6. CAD Evaluation Metrics

The evaluation of CAD systems must capture both *detection quality* and *CL dynamics*. Standard AD metrics, including Area Under the Receiver Operating Characteristic Curve (AUROC/ROC-AUC) and Area Under the Precision–Recall Curve (AUPRC/PR-AUC), remain central for measuring task-level detection performance. However, CAD requires reporting these metrics *through time*, not just at the end of training, since the model is expected to both adapt to new regimes and retain prior knowledge.

Performance is typically evaluated on each task after each training stage. Let $R_{i,j}$ denote the detection performance (e.g., ROC-AUC) on task j after learning up to task i . N denote the number of tasks (or regimes) in the stream, $T \triangleq \frac{N(N+1)}{2}$ is the number of tasks on and below diagonal, and $T' \triangleq \frac{N(N-1)}{2}$ is the number of tasks in the lower or upper diagonal. **Lifelong ROC-AUC (LROC)** averages performance over all tasks encountered so far at every point in the scenario:

$$LROC = \frac{1}{T} \sum_{i \geq j}^N R_{i,j}. \quad (1)$$

Beyond detection performance, CAD studies increasingly report metrics quantifying retention and transfer. **Backward Transfer (BWT)** measures how learning new tasks affect performance on earlier tasks, capturing transfer/forgetting:

$$BWT = \frac{1}{T'} \sum_{i=2}^N \sum_{j=1}^{i-1} (R_{i,j} - R_{j,j}). \quad (2)$$

Closely related to BWT, **Forgetting Measure (FM)** isolates degradation by comparing current performance on a task to the best performance previously achieved on that same task. After learning N tasks, the average FM is defined as:

$$FM = \frac{1}{N} \sum_{j=1}^N \left(\max_{1 \leq i \leq N} R_{i,j} - R_{N,j} \right). \quad (3)$$

Unlike BWT, FM does not credit relative improvements on earlier tasks that may occur after learning new tasks. Instead, it focuses purely on comparison with a task's historical peak.

Forward Transfer (FWT) evaluates whether learning earlier tasks improves performance on future tasks:

$$FWT = \frac{1}{T'} \sum_{i < j}^N R_{i,j}. \quad (4)$$

In practice, FWT is under-adopted, partly because it is harder to interpret in CAD than in supervised classification, as future-task evaluation depends on how tasks are defined, how normality shifts, and

whether anomaly semantics remain stable. We argue that CAD evaluation should reflect deployment realities. Many works report only performance metrics, but practical systems are constrained by update cost, latency, and resource budgets. We recommend reporting efficiency measures (e.g., wall-clock update time, updates, FLOPS, peak memory footprint, and replay buffer size when applicable) alongside other metrics. Moreover, metrics alone are not sufficient, as CAD results can be highly sensitive to the adopted evaluation protocol, including the order of the tasks presented to the model.

7. Open Challenges and Roadmap

Although CAD has gained increasing research attention, significant gaps remain. Below, we outline key open challenges and a practical research roadmap.

CAD-Specific Datasets and Benchmark

As CAD remains an emerging field, there is an apparent lack of datasets specifically designed to support its unique requirements, such as sequential task presentation, recurring concepts, and significantly evolving definitions of normality. While in certain domains, such as industrial visual inspection, existing datasets like MVTec-AD can be naturally repurposed by organizing product categories into tasks, other modalities, especially non-vision, pose greater challenges. In these settings, the absence of inherent task structure often leads to arbitrary or ambiguous task definitions, unclear task similarity, and, at times, artificially low forgetting due to too small differences between tasks. Consequently, these benchmarks often underestimate the problem's difficulty and fail to reflect the complexities encountered in real-world deployment. In addition, claims about representation behavior and robustness to shift can be fragile to dataset diversity and scale, motivating benchmark suites that better match real-world variability [37].

Roadmap: As a result, there is a pressing need for CAD-specific datasets and benchmarks that reflect realistic and diverse scenarios, encompassing both gradual and abrupt distributional shifts, varying inter-task similarity, and well-defined evolution of normality. In parallel, there is also a need for a principled evaluation of the usefulness and difficulty of a given scenario, based on aspects such as expected forgetting and task dissimilarity, to guide meaningful benchmarking and comparison.

CAD for Streaming Data

In many real-world applications, anomaly detection systems operate in streaming environments, where data arrives sequentially and predictions must be delivered in real time, without retraining phases. Although most CL research, including CAD, remains centered on *offline CL*, with access to clear task structure and (almost) unlimited computational resources, Ghunaim et al. [44] discusses more constrained settings. *Online CL* requires the model must learn using just a small batch of data at a time, while *real-time online CL* further imposes limits based on computational latency and data throughput. Such streaming conditions provide challenges such as detection/adaptation delay, i.e., the period during which the model continues to rely on outdated knowledge as it has not yet adapted to the new distribution.

Roadmap: Despite its practical relevance, the streaming perspective remains unexplored in CAD. To close this gap, there is a the need for streaming-oriented evaluation protocols, realistic benchmarks, and methods explicitly designed for real-time operation.

Compute, Memory, and Update Budgets

Storage limitations in continual learning research are frequently justified by concerns about memory costs [45], but in practice storage is often significantly cheaper than computational power. Moreover, in real-world scenarios, especially in edge computing and time-sensitive environments, reducing computational complexity is often more crucial than minimizing memory usage. Despite that, most CAD methods neglect the computational cost of continual updates, and compute-related metrics are rarely reported, which limits suitability of CAD methods for constrained real-time deployment.

In addition, recent CL methods have shown that evaluating at more frequent intervals can reveal opportunities for improving computational efficiency and address performance gaps [46].

Roadmap: Future work should treat compute as a budgeted resource during design and benchmarking, and explicitly measure the trade-off between detection gains and added gradient steps, update frequency, and wall-clock time [45]. Evaluate at more frequent intervals than only at task gaps to gain insights into how the algorithm would behave throughout deployment.

Experimental Design, Reproducibility, and Fair Comparison

Robust and efficient progress in CAD depends on consistent experimental design and strong reproducibility standards. Robust and efficient progress in continual anomaly detection (CAD) depends on consistent experimental design and strong reproducibility standards. The current challenges include under-specification of learning scenarios (see Section 3), limited set of metrics (see Section 6), lack of consistent use of baselines, and limited public availability of tools.

Roadmap: 1) It is essential to clearly define task structure, specifying what constitutes a unit of normality, and to state whether task boundaries and identities are available during training and inference (see Table 1). 2) Since aggregated metrics like average continual ROC-AUC or PR-AUC provide only a limited view of performance, they should be complemented with metrics such as BWT, FM, and FWT. 3) Adopting granular reporting, such as per-task or per-timestep performance, can reveal trends, instabilities, and model characteristics that are hidden in summary statistics. Visualization tools, such as heatmaps and time-aligned performance plots, can be particularly useful in this regard. 4) We advocate the inclusion of strong baselines as essential reference points: i) Naive incremental learning, which performs sequential updates without forgetting mitigation; ii) Task-specific experts, with a separate model per task; iii) Cumulative learning, where models are trained on all data seen so far; and iv) Simple replay strategies, using constrained memory buffers. 5) Finally, we strongly advocate for the availability of public code, datasets, and experimental pipelines. Despite some progress in the creation of publicly available libraries, such as pyCLAD¹, further development of open-source tools is essential to foster transparency.

Robustness under Feedback and Contamination

The reality of deployment for CAD systems includes many challenges that are easy to miss in offline experimental setup. The aspects such as error accumulation from self-training, lack of task boundaries, and threshold drift can silently degrade performance long before headline metrics react. In addition, supervision levels and data cleanliness are often overestimated. Although some domains may offer labeled, clean data, most CAD applications must operate in semi-supervised or unsupervised conditions, often with contaminated training sets.

Roadmap: There is a need for more substantial alignment with real-world challenges by, for example, adding stress tests that inject contamination. Moreover, we advocate considering less restrictive scenarios without assumptions about task boundaries or identities, or with limited supervision.

Representations and Objective Conflict

Pretrained backbones and lightweight adaptation are an increasing trend (see Section 4), but CAD needs a clearer guidance on what representation properties matter. Evidence suggests that objectives for anomaly detection and objectives for knowledge transfer can pull embeddings in different geometric directions, making a single best feature space unlikely [37].

Roadmap: It is important to distinguish improvements resulting from: i) feature representations; ii) scoring and calibration mechanisms; iii) adaptation algorithms. Moreover, the evaluation should provide a separate analysis of these spaces.

¹ <https://github.com/lifelonglab/pyCLAD>

Humans and Operations

Real deployments rarely run without any supervision forever, and, even though human feedback is often scarce, delayed, and expensive, it is the only reliable way to resolve ambiguity about borderline cases and contamination. The open question is not whether humans help, but how to use them efficiently.

Roadmap: There is a need for devising human-AI suitable for specific domains. We should move beyond “query uncertain points” as the default interaction and study query types that target coverage gaps (e.g., long-tail regimes), measuring end-to-end gains under realistic label budgets and delays.

8. Conclusions

Continual anomaly detection is emerging as a practical response to high-stakes applications characterized by non-stationary data, limited supervision, and evolving definitions of normality. In this survey, we discuss the current literature across key dimensions to provide a structured and comprehensive view of the field. We can observe a consistent picture across different domains and methods: progress is increasingly driven not only by how to update models to prevent forgetting, but also by when to update and what to preserve under tight operational constraints. These challenges are central to building CAD systems that are both adaptive and reliable over a long-time horizon. We also provide a research roadmap aimed at supporting the next stages of the field development. In the future, it is essential to develop more realistic and standardized scenarios, domain-specific benchmarks, as well as robust and transparent evaluation protocols. Moreover, future methods must address challenges such as the absence of task boundaries, streaming conditions, and incorporating scarce yet important human feedback into the learning loop. By analyzing current state-of-the-art and identifying open problems, we aim to stimulate foundational research that will advance reliable and robust continual anomaly detection.

Acknowledgments: KF was supported by funding from the Polish Ministry of Science and Higher Education under the program “Co-funded International Projects,” and Digital Europe Programme (Grant Agreement No. 101128073); CK was partly supported by NSF award #2326491; RC was partly supported by American University through a CAS Faculty Grant for Scholarship & Creative Inquiry; KF & RC were partly supported by “Excellence initiative—research university” for the AGH University in Krakow”. The views and conclusions contained herein are those of the authors and should not be interpreted as representing any sponsor’s official policies or endorsements.

References

1. Lu, T.; Wang, L.; Zhao, X. Review of anomaly detection algorithms for data streams. *Applied Sciences* **2023**, *13*, 6353.
2. Liu, J.; Wu, K.; Nie, Q.; Chen, Y.; et al. Unsupervised continual anomaly detection with contrastively-learned prompt. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2024, Vol. 38, pp. 3639–3647.
3. Wang, L.; Zhang, X.; Su, H.; Zhu, J. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence* **2024**, *46*, 5362–5383.
4. Faber, K.; Corizzo, R.; Sniezynski, B.; Japkowicz, N. Lifelong continual learning for anomaly detection: New challenges, perspectives, and insights. *IEEE Access* **2024**, *12*, 41364–41380.
5. Roady, R.; Hayes, T.L.; Kemker, R.; Gonzales, A.; Kanan, C. Are open set classification methods effective on large-scale datasets? *PLOS ONE* **2020**, *15*, e0238302.
6. Faber, K.; Corizzo, R.; Sniezynski, B.; Japkowicz, N. VLAD: Task-agnostic VAE-based lifelong anomaly detection. *Neural Networks* **2023**, *165*, 248–273.
7. Harun, M.Y.; Gallardo, J.; Hayes, T.L.; Kemker, R.; Kanan, C. SIESTA: Efficient Online Continual Learning with Sleep. *Transactions on Machine Learning Research* **2023**.
8. González, G.G.; Casas, P.; Fernández, A. Fake it till you detect it: Continual anomaly detection in multivariate time-series using generative AI. In Proceedings of the 2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). IEEE, 2023, pp. 558–566.

9. Faber, K.; Corizzo, R.; Sniezynski, B.; Japkowicz, N. Active lifelong anomaly detection with experience replay. In Proceedings of the 2022 IEEE 9th international conference on data science and advanced analytics (DSAA). IEEE, 2022, pp. 1–10.
10. Lebichot, B.; Sibli, W.; Paldino, G.M.; Le Borgne, Y.A.; Oblé, F.; Bontempi, G. Assessment of catastrophic forgetting in continual credit card fraud detection. *Expert Systems with Applications* **2024**, *249*, 123445.
11. Benameur, R.; Dahane, A.; Souihi, S.; Mellouk, A. A Robust and Scalable Federated Continual Learning Framework for Adaptive DDoS Detection in Heterogeneous IoT Environments. In Proceedings of the ICC 2025-IEEE International Conference on Communications. IEEE, 2025, pp. 3063–3068.
12. Guo, C.; Li, X.; Cheng, J.; Yang, S.; Gong, H. Continual Learning for Intrusion Detection Under Evolving Network Threats. *Future Internet* **2025**, *17*, 456.
13. Cai, Y.; He, X.; Tong, A.; Bai, X. Dynamic Expert Routing for Unsupervised Continual Anomaly Detection. *IEEE Transactions on Industrial Informatics* **2025**.
14. Zhang, Z.; Zou, G.; Chen, C.; Qi, Z.; et al. A Task-Aware Parameter Decoupling Framework for Continual Anomaly Detection. *IEEE Transactions on Industrial Informatics* **2025**.
15. Yang, J.; Shen, Y.; Deng, L. Continual Contrastive Anomaly Detection under Natural Data Distribution Shifts. In Proceedings of the 2023 8th International Conference on Automation, Control and Robotics Engineering (CACRE). IEEE, 2023, pp. 144–149.
16. Hu, C.; Xu, H.; Li, Y.; Sun, C.; Chen, X.; Yan, R. Continual anomaly detection with Cross-Condition feature contrast for nonstationary industrial data stream. *Mechanical Systems and Signal Processing* **2025**, *238*, 113254.
17. Fan, W.; Shangguan, W.; Bouguila, N. Continuous image anomaly detection based on contrastive lifelong learning. *Applied Intelligence* **2023**, *53*, 17693–17707.
18. Faber, K.; Corizzo, R.; Sniezynski, B.; Baron, M.; Japkowicz, N. Lifewatch: Lifelong wasserstein change point detection. In Proceedings of the 2022 International joint conference on neural networks (IJCNN). IEEE, 2022, pp. 1–8.
19. Coil, C.; Faber, K.; Śnieżyński, B.; Corizzo, R. Distance-based change point detection for novelty detection in concept-agnostic continual anomaly detection. *Journal of Intelligent Information Systems* **2025**. <https://doi.org/10.1007/s10844-025-00949-1>.
20. Tomas, P.R.; Felix, P.; Rosa, L.; Gomes, A.S.; Cordeiro, L. A novel approach for continual and federated network anomaly detection. In Proceedings of the Proceedings of the Future Technologies Conference. Springer, 2024, pp. 212–225.
21. Li, X.; Tan, X.; Chen, Z.; Zhang, Z.; et al. One-for-more: Continual diffusion model for anomaly detection. In Proceedings of the Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 4766–4775.
22. Amalapuram, S.K.; Tamma, B.R.; Channappayya, S.S. Spider: A semi-supervised continual learning-based network intrusion detection system. In Proceedings of the IEEE INFOCOM 2024-IEEE Conference on Computer Communications. IEEE, 2024, pp. 571–580.
23. Gori, V.; Veneri, G.; Ballarini, V. Continual Learning for anomaly detection on turbomachinery prototypes-A real application. In Proceedings of the 2022 IEEE congress on evolutionary computation (CEC). IEEE, 2022, pp. 1–7.
24. Kumar, G.A.; Lavanya, B.M.; Mohiddin, M.K.; Mitra, S. IoT-enabled recurrent spatio-temporal adaptive attention of temporal convolutional transformer with continual learning for dairy farming. *Expert Systems with Applications* **2025**, p. 127712.
25. Doshi, K.; Yilmaz, Y. Continual learning for anomaly detection in surveillance videos. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 254–255.
26. Doshi, K.; Yilmaz, Y. Rethinking video anomaly detection-a continual learning approach. In Proceedings of the Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2022, pp. 3961–3970.
27. Li, W.; Zhan, J.; Wang, J.; Xia, B.; et al. Towards continual adaptation in industrial anomaly detection. In Proceedings of the Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 2871–2880.
28. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual lifelong learning with neural networks: A review. *Neural networks* **2019**, *113*, 54–71.
29. Corizzo, R.; Baron, M.; Japkowicz, N. Cpdga: Change point driven growing auto-encoder for lifelong anomaly detection. *Knowledge-Based Systems* **2022**, *247*, 108756.

30. Pezze, D.D.; Anello, E.; Masiero, C.; Susto, G.A. Continual learning approaches for anomaly detection. *Evolving Systems* **2025**, *16*, 111.
31. Maschler, B.; Pham, T.T.H.; Weyrich, M. Regularization-based continual learning for anomaly detection in discrete manufacturing. *Procedia CIRP* **2021**, *104*, 452–457.
32. Ejaz, A.; Mian, A.N.; Manzoor, S. Life-long phishing attack detection using continual learning. *Scientific reports* **2023**, *13*, 11488.
33. Tang, J.; Lu, H.; Xu, X.; Wu, R.; et al. An incremental unified framework for small defect inspection. In Proceedings of the European conference on computer vision. Springer, 2024, pp. 307–324.
34. Yu, M.; Wang, F. Ada-CAD: Adaptive Distillation and Dynamic Neighbor Masked Attention for Continual Anomaly Detection. In Proceedings of the International Conference on Intelligent Computing. Springer, 2025, pp. 182–194.
35. Zhou, Y.; Lei, T.; Liu, H.; Du, N.; Huang, Y.; et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems* **2022**, *35*, 7103–7114.
36. Dahmardeh, M.; Setti, F. MECAD: A multi-expert architecture for continual anomaly detection. *arXiv preprint arXiv:2512.15323* **2025**.
37. Harun, M.Y.; Gallardo, J.; Kanan, C. Controlling Neural Collapse Enhances Out-of-Distribution Detection and Transfer Learning. In Proceedings of the International Conference on Machine Learning, 2025.
38. Chavan, V.; Koch, P.; Schlüter, M.; Briese, C.; Krüger, J. Active Data Collection and Management for Real-World Continual Learning via Pretrained Oracle. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2024, pp. 4085–4096.
39. Zhou, M.; Liu, J.; Wan, J.; Li, G.; Li, M. Exploring Multimodal Prompts For Unsupervised Continuous Anomaly Detection. In Proceedings of the Proceedings of the 33rd ACM International Conference on Multimedia, 2025, pp. 1510–1519.
40. Li, E.; Gungor, O.; Shang, Z.; Rosing, T. CITADEL: Continual Anomaly Detection for Enhanced Learning in IoT Intrusion Detection. *arXiv preprint arXiv:2508.19450* **2025**.
41. Bugarin, N.; Bugaric, J.; Barusco, M.; Pezze, D.D.; Susto, G.A. Unveiling the anomalies in an ever-changing world: A benchmark for pixel-level anomaly detection in continual learning. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 4065–4074.
42. Chin, M.; Corizzo, R. Continual semi-supervised malware detection. *Machine Learning and Knowledge Extraction* **2024**, *6*, 2829–2854.
43. Sayed, A.N.; Bensaali, F.; Nhlabatsi, A. Adaptive Appliance-Level Anomaly Detection Using Regularization and Replay Continual Learning Methods. In Proceedings of the Proceedings of the Sixth International Conference on Digital Age & Technological Advances for Sustainable Development, 2025, pp. 74–80.
44. Ghunaim, Y.; Bibi, A.; Alhamoud, K.; Alfarra, M.; et al. Real-time evaluation in online continual learning: A new hope. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 11888–11897.
45. Verwimp, E.; Aljundi, R.; Ben-David, S.; Bethge, M.; Cossu, A.; Gepperth, A.; et al. Continual learning: Applications and the road forward. *Transactions on Machine Learning Research* **2024**.
46. Harun, M.Y.; Kanan, C. Overcoming the stability gap in continual learning. *TMLR* **2024**.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.