Sri Balaji Societys The Balajian Journal of Management Research
3(2) 2026, 01-17
www.publishingindia.com/journals/bjmr/
ISSN: 3049-317X(Online)

A Systematic Comparative Review of Anomaly Detection Techniques in Cloud Computing Environments

Manisha Milind Patil*, Dhanraj P. Tambuskar**

Abstract

Cloud computing has emerged as the primary model for providing flexible computing resources, with applications ranging from common enterprise systems to those that are safety critical. However, the very aspects of cloud computing such as scalability and elasticity complicate the monitoring process, since jobs are transferred from one server to another, resource baselines change due to automatic resource scaling, and the amount of data to be analysed increases significantly. In this connection, anomaly detection technology, comprising techniques that help identify unusual activities in the system, has been developed and applied within the framework of cloud computing. This article conducts a systematic review of the existing literature, in line with the recommendations provided by Kitchenham. The review aims to answer four questions relating to the types of anomalies studied, the techniques chosen for analysis, the types of data used, and the factors that influence the choice of a particular methodology. Following a systematic search of five digital libraries (IEEE Xplore, Scopus, ACM Digital Library, ScienceDirect, and SpringerLink) from 2015 to 2026, duplicates were removed, and studies were subjected to two-stage screening and quality assessment to yield 78 primary studies for the subsequent qualitative synthesis. In this way, we establish a taxonomy of cloud anomalies in three dimensions, namely performance, security, and resources, and conduct a comparison of four major categories of anomaly detection techniques, i.e., statistical approaches, classical machine learning, unsupervised learning, and deep learning, as well as hybrids and ensembles, based on criteria that matter in a production cloud environment, i.e., detection accuracy, false-positive rate, computational cost, adaptability to concept drift, interpretability, and streaming capability. It is evident from our findings that there is no one-size-fits-all solution, since although deep learning algorithms like LSTM autoencoders provide high accuracy in detecting anomalies in highly complex temporal telemetry data, these solutions are highly costly and non-intuitive, whereas light-weight algorithms like Isolation Forest and Statistical Process Control manage to deliver satisfactory performance in real-time and constrained environments.

Keywords: Anomaly Detection, Cloud Computing, Systematic Literature Review, Machine Learning, Deep Learning, Intrusion Detection, Performance Monitoring, AIOps

Introduction

Cloud computing provides computing resources, storage, and network as a utility service available through Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). According to industry figures, most enterprise workloads run in either public or hybrid cloud environments, and hyperscale vendors have begun operating data centres with hundreds of thousands of servers whose combined telemetry - CPU usage, memory pressure, disk/network I/O, request latency, error rates, audit trails, and billing events - make up one of the largest continuously generated data flows in the world. In this stream, a minuscule number of samples represent those events that need to be known to the operators: a compromised VM running within a botnet, a memory leak gradually bringing down a customer-facing service, an incorrectly configured auto-scaler quietly ramping up costs, and a dying disk that is going to bring down a storage node.

* Research Scholar, Balaji Institute of Modern Management, Sri Balaji University, Pune, Maharashtra, India.
Email: manishamilindpatil@gmail.com (Corresponding author)
** Professor, Balaji Institute of Modern Management, Sri Balaji University, Pune, Maharashtra, India.
Email: dhanraj.tambuskar12@gmail.com

Anomaly detection is the discipline of identifying such observations automatically. According to Chandola et al. (2009), in their seminal survey, anomalies are "data patterns that do not conform to an expected pattern of behaviour." In cloud-based infrastructures, however, this characterisation is further complicated by three structural characteristics. The first is elasticity: normal behaviour is non-stationary, owing to legitimate changes to baseline resources caused by auto-scaling, live migration, and workload consolidation on an hourly basis. The second is multi-tenancy: co-resident tenants cause performance interference ("noisy neighbours") that mimics anomalous behaviour yet is external to the workload under consideration. The third is scale and velocity.

The fast-growing literature on anomaly detection for cloud computing has covered practically all major paradigms, from classical statistical process control and time-series forecasting, through supervised learning on labelled attack datasets, to unsupervised and semi-supervised learning based on the concept of normality, and even deep learning based on autoencoders, recurrent neural networks, graph neural networks, and transformers. The typical approach adopted in individual research papers consists of showing that an algorithm outperforms several baselines on just one or two datasets. What is really lacking is an aggregated comparative analysis that explains under what conditions the different approaches may be used and what their trade-offs are. In addition, it is important to know how the results obtained by these approaches should be interpreted, taking into account the peculiarities of widely used benchmark datasets. Most existing surveys are largely narrative and their selection of studies is not explicit (e.g., Chalapathy & Chawla, 2019; Fernando et al., 2021; Pang et al., 2021; Soldani & Brogi, 2022).

Most importantly, recent work identifies multiple recurring flaws in the existing literature that inspire the current review. While general deep learning surveys, such as Chalapathy and Chawla (2019) and Pang et al. (2021), provide a comprehensive overview of architectures, they ignore the constraints of actual deployment in cloud, namely concept drift, alert fatigue, and computational cost. While domain-specific surveys, such as performance anomaly detection and bottleneck identification by Ibidunmoye et al. (2015) and failure root-cause analysis in microservices by Soldani and Brogi (2022), provide more insight than general surveys do, they either overlook the transformer-based or graph-based techniques which make the state-of-the-art or only refer to outdated techniques. Finally, while some machine-learning-based anomaly detection surveys, such as Nassif et al. (2021), aggregate information from various application domains without addressing the cloud-specific challenges of technique selection, including elasticity, multi-tenancy, and telemetry heterogeneity. Importantly, no previous work offers an explicit, reproducible study selection protocol along with comparative analysis of selected techniques based on certain criteria for their deployment in production. This is the gap addressed in the present systematic review.

In this paper, we aim at addressing this gap through systematic literature review (SLR) performed in accordance with the guidelines of Kitchenham and Charters (2007) for evidence-based software engineering and reported in accordance with PRISMA statement (Page et al., 2021). The current review will be performed in accordance with the following four research questions:

- RQ1: What kinds of anomalies are studied in the domain of clouds, and in which layer(s) of the cloud stack are they detected?
- RQ2: What techniques of anomaly detection have been applied to clouds, and what are the assumptions, pros and cons of these techniques?
- RQ3: What datasets and evaluation metrics are used to evaluate these techniques, and how robust are the obtained comparisons?
- RQ4: What trade-offs exist between selecting an appropriate detection technique for a certain cloud setting, and what open problems exist?

The above questions give rise to the following four contributions of the paper:

- Anomalies found in cloud computing are classified according to the kind of anomaly (point anomaly, contextual anomaly, collective anomaly), the level of occurrence of the anomaly (infrastructure, platform, application) and the operational field (performance, security, resources/cost).
- Comparative synthesis of four major classes of methods (RQ2) statistical methods, traditional supervised machine learning methods, unsupervised learning methods, deep learning methods, and also

hybrid and ensemble methods according to 78 primary studies identified through explicit search and screening procedure.

- Critical review of the benchmarks and evaluation measures employed (RQ3), including the artifacts that can be found in the datasets used for benchmarking and the biases induced by point-based evaluation methodology.
- Comparison against a set of criteria that affect practical applicability (RQ4) accuracy, false positive rate, computational overhead, memory overhead, labelling needs, drift resistance, interpretability and streamability as well as current issues and future research directions.

The remainder of the paper is organised as follows: Section II provides background on cloud architectures and monitoring data sources. Section III describes the review methodology, including the search strategy, selection criteria, quality assessment, and the resulting body of primary studies. Section IV develops the anomaly taxonomy (RQ1). Section V surveys detection techniques by family (RQ2). Section VI reviews benchmark datasets and evaluation metrics (RQ3). Section VII presents the comparative analysis (RQ4). Section VIII synthesises and discusses the findings, including their implications and the limitations of the study. Section IX discusses challenges and open issues, Section X outlines future directions, and Section XI concludes.

Background

Cloud Service and Deployment Models

The NIST reference model characterises cloud computing by five essential properties on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service delivered through the IaaS, PaaS, and SaaS service models and deployed as public, private, hybrid, or community clouds (Mell & Grance, 2011). Each layer generates distinct telemetry and exhibits distinct anomaly surfaces (Ibidunmoye et al., 2015; Soldani & Brogi, 2022). At the IaaS layer, hypervisors and virtual machines expose hardware-proximate counters (CPU steal time, page faults, disk queue depth) and are exposed to attacks such as VM escape and cross-tenant side channels. The PaaS layer, often implemented with containers and orchestration systems such as Kubernetes, brings additional events in the form of schedule events, pod restarts, and service-mesh traffic. Application logs, user behaviour, and API audits belong to the SaaS layer. Function-as-a-Service serverless platforms offer another layer, where execution units are temporary, existing for only milliseconds at a time.

Monitoring Data Sources in the Cloud

Cloud observability signals used by the anomaly detection framework can be categorized into four classes (Ibidunmoye et al., 2015; Soldani & Brogi, 2022). (i) Metrics: numeric time-series samples like CPU utilization, memory utilization, network bandwidth, requests per second, latency percentiles, and error rate measurements, often gathered via agents (Prometheus exporters, CloudWatch agents, and the like), with sampling frequency varying between one second and one minute. (ii) Logs: text signals in structured or unstructured format that represent events generated by operating systems, middleware software, and applications; log-based anomaly detection involves the step of parsing where the input message is matched against an event template (He et al., 2017). (iii) Traces: distributed traces of a request traversing microservices and recording the causality trail of the request, allowing detection of latency anomalies localised to particular hops. (iv) Network and audit logs: packet-capture logs, flow records (NetFlow/IPFIX, VPC flow logs), and control-plane audit logs (e.g., AWS CloudTrail) underpinning security-related anomaly detection. Such diversity of data sources calls for multi-modal detection approaches and renders direct comparison difficult, since an approach applicable to metric time series may fail on tokenised sequences from logs.

The Anomaly Detection Problem

Formally, given a set of all possible observations X and a distribution P_normal over X that represents normal behaviour, an anomaly detector is defined by a function that maps observations to scores, which in turn are compared against a threshold to signal an anomaly (Chandola et al., 2009). Three learning regimes are distinguished by label availability (Chandola et al., 2009; Pang et al., 2021). Supervised detection assumes labelled examples

of both normal and anomalous behaviour and reduces to (typically highly imbalanced) classification. Semi-supervised detection assumes labels for normal data only and learns a model of normality, flagging poor fits at inference time. Unsupervised detection assumes no labels and relies on the working hypothesis that anomalies are rare and different, using density, distance, or isolation properties to rank observations. Cloud practice overwhelmingly favours the semi-supervised and unsupervised regimes, because labelled anomalies are scarce, expensive to produce, and quickly outdated as both workloads and attacks evolve (Chalapathy & Chawla, 2019; Pang et al., 2021).

Research Methodology

This review follows the systematic literature review process defined by Kitchenham and Charters (2007), comprising three phases planning (research questions, protocol definition), conducting (search, selection, quality assessment, data extraction), and reporting and is documented in accordance with the PRISMA statement (Page et al., 2021). The four research questions RQ1–RQ4 stated in Section I were fixed before the search was executed, together with the search strategy, selection criteria, and quality-assessment checklist described below, in order to minimise selection bias.

Search Strategy

Five databases were selected for this study: IEEE Xplore, Scopus, ACM Digital Library, ScienceDirect, and SpringerLink. The reasoning behind this choice is that these platforms include the most comprehensive databases of academic journals on cloud computing, systems, security, and machine learning. The search string entered in each platform (with slight modifications according to the requirements of specific databases) was the following:

("anomaly detection" OR "outlier detection" OR "intrusion detection") AND ("cloud computing" OR "cloud environment" OR "data center" OR "virtual machine" OR "container" OR "microservice")

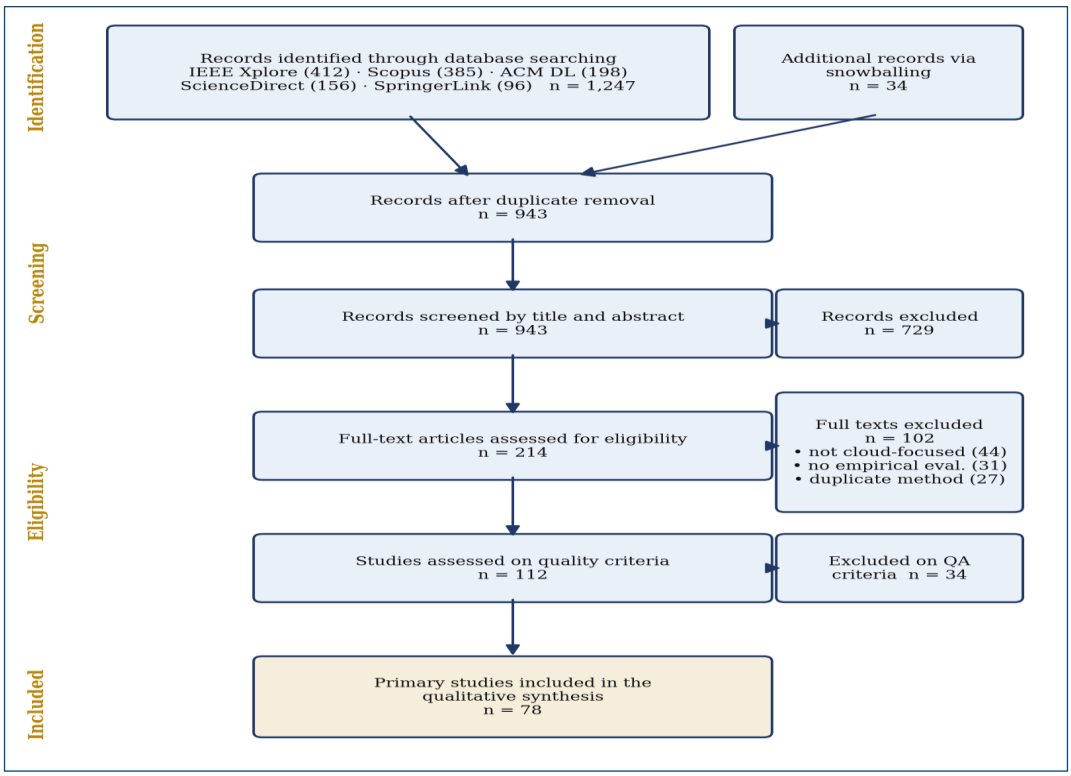
The search was limited to peer-reviewed journal articles and conference papers published between January 2015 and January 2026. The choice of 2015 as a starting point was motivated by the appearance of a significant dataset in network-security research, namely the modern intrusion benchmark UNSW-NB15, as well as a surge in the application of deep learning in operational anomaly detection. There is therefore a good basis for concluding that the review covers the whole recent history of the field while remaining up to date. Both backward and forward snowballing were applied to reference lists and citing articles of the included papers to find relevant studies missed by the initial search, which added another 34 papers to the pool of candidates.

Inclusion and Exclusion Criteria

Studies were retained if they satisfied all inclusion criteria: (I1) the study proposes, applies, or empirically compares one or more anomaly detection techniques; (I2) the target environment is a cloud, data-centre, virtualised, containerised, or microservice system, or the evaluation uses cloud-representative data (cloud telemetry, cluster traces, cloud-generated intrusion datasets, system logs); (I3) the study reports an empirical evaluation with quantitative results; and (I4) the full text is accessible. Studies were excluded if: (E1) anomaly detection was peripheral rather than central to the contribution; (E2) the domain was unrelated to cloud or general system telemetry (e.g., medical imaging, video surveillance) with no transferable evaluation; (E3) the paper was a short abstract, poster, editorial, or non-peer-reviewed preprint without a subsequently published version; (E4) the paper duplicated an earlier study by the same authors without substantive extension (the most complete version was retained); or (E5) no empirical evaluation was reported.

Study Selection

Fig. 1 presents the selection process and provides a summary of the initial search and screening stage. From the initial database search, 1,247 records were identified, to which 34 were added by snowballing. After duplicates were removed, 943 records remained. These were screened against the selection criteria on titles and abstracts, which excluded 729 records. Therefore, 214 records were subjected to full-text review. From these, 102 articles were excluded (44 were not focused on cloud computing, 31 did not report an empirical study, and 27 were duplicates). Thus, 112 studies were included for the quality-assessment stage, out of which 78 were retained for the main synthesis, forming the basis of Sections IV–VI.



Source: Authors' own, following the PRISMA 2020 framework (Page et al., 2021).

Fig. 1: PRISMA Flow Diagram of the Study Identification and Selection Process

Quality Assessment

Each of the 112 studies included for full-text screening was assessed using a five-item checklist based on the criteria of Kitchenham and Charters (2007): (QA1) Are the research aims explicitly stated? (QA2) Is the detection method sufficiently detailed to be re-implemented? (QA3) Is the dataset described or publicly available? (QA4) Are evaluation metrics appropriate for imbalanced data (i.e., not just accuracy)? and (QA5) Are threats to validity or limitations adequately discussed? Each study was scored 1 (yes), 0.5 (partially), or 0 (no) on each item, with those scoring below 3.0 of 5 excluded on grounds of insufficient reporting. This cut-off excluded 34 studies, with QA3 (datasets) and QA4 (evaluation) being the most common reasons (see Fig. 2).

Data Extraction and Synthesis

The following information was extracted from each primary study: year and publication venue, type(s) of anomaly and cloud layer under consideration (RQ1); technique class and particular algorithms (RQ2); datasets, performance measures and reported results (RQ3); and reported cost, deployment and limitations (RQ4). Of the total 78 primary studies, 31 (40%) use deep learning techniques, 22 (28%) classical unsupervised learning techniques, 14 (18%) supervised machine learning techniques, and 11 (14%) statistical and hybrid methods. There has been a clear transition from using statistical or hybrid approaches to deep learning and hybrid approaches since 2019. Security anomalies have received the most attention in this literature (36 studies), while performance and availability/resource anomalies come in second and third place with 28 and 14 studies, respectively. Since there are great differences between primary studies in terms of datasets, methods and measures, quantitative meta-analysis (statistical pooling of results) has not been applied; thus, the synthesis in Sections IV-VII is qualitative and criteria-based.

Threats to Validity of the Review

Terminological variation (such as "novelty detection," "fault detection," "outage prediction") poses a threat to construct validity, and while snowballing helps, it does

not solve this problem. Internal validity is challenged by single-screener selection and subjectivity of quality assessment, and this is partially resolved by pre-registration and the scoring criteria. External validity is undermined by publication bias towards significant findings and by the fact that only a few benchmark datasets are used which have inherent artefacts (discussed in section 'Benchmark datasets and evaluation methodology), thus limiting the external validity of the included studies. This should be kept in mind when interpreting comparative findings (section "Comparative Analysis").

A Taxonomy of Anomalies in Cloud Environments

By Structural Type

Following the classical typology (Chandola et al., 2009), cloud anomalies may be point, contextual, or collective. A point anomaly is a single observation that is anomalous with respect to the entire dataset—for instance, a virtual machine reporting 100% CPU steal time. A contextual anomaly is anomalous only within a context, most commonly time: 70% CPU utilisation may be normal during business hours but anomalous at 3 a.m. for a batch-free workload. Cloud telemetry, dominated by seasonal daily and weekly patterns, makes contextual anomalies the most common and the most diagnostic category. A collective anomaly is a group of observations that is anomalous in aggregate although each member appears individually normal for example, a low-and-slow data exfiltration in which each network flow is unremarkable but the sustained pattern of outbound transfers is not, or a cascade of individually plausible microservice retries that collectively signal a failing dependency.

By Affected Layer

Anomalies manifest at every layer of the cloud stack. At the infrastructure layer they include hardware degradation (failing disks with rising I/O latency, correctable-memory-error bursts that precede DIMM failure), hypervisor faults, and network faults such as packet loss caused by oversubscribed top-of-rack switches. At the virtualisation layer they include noisy neighbours, VM migration hangs, container restart loops, and scheduler-related problems such as pod eviction due to memory pressure. Application-level problems include memory leaks, deadlock, latency increases caused by deployment, and sudden error-rate increases. Layer classification is important because the appropriate metrics, features, and mitigation techniques differ for each layer; latency-percentile monitoring at the application layer cannot, without proper infrastructure-layer data, distinguish code degradation from an infrastructure-layer problem such as a noisy neighbour.

By Operational Domain

The following are the principal domains from the standpoint of an operator. Performance anomalies lead to poor quality of service: latency, throughput, and resource-saturation issues, and slowly degrading issues such as leaks. Security anomalies signal malicious behaviour: distributed denial-of-service (DDoS) attacks, brute-force authentication attacks, lateral movement between compromised hosts, crypto-mining on hijacked capacity, container breakout, and cloud control-plane API anomalies such as IAM role assumption or bulk object download from storage buckets. Resource and cost anomalies relate to efficiency and governance: uncontrolled auto-scaling, orphaned resources, billing spikes, and quota exhaustion. While there is significant overlap in the underlying mathematics across all domains, the operational requirements differ significantly: security anomaly detection can tolerate some latency but must have very low false-negative rates for certain attack classes, whereas performance anomaly detection must be near real-time and focused on alert precision, since on-call fatigue from false-positive alerts is the main failure point of production monitoring systems.

Table 1: Operational Taxonomy of Cloud Anomalies with Representative Telemetry Cues

Domain	*Representative Anomalies*	*Typical Telemetry and Detection Cues*
Performance	Latency spikes; memory leaks; CPU/disk saturation; throughput collapse; degradation after deployment.	Latency percentiles, resource metrics, traces; contextual deviation from seasonal baseline; slow trend drift.

Domain	*Representative Anomalies*	*Typical Telemetry and Detection Cues*
Security	DDoS; brute force; lateral movement; cryptomining; data exfiltration; control-plane abuse.	Flow records, packet features, audit logs, login events; volumetric spikes; rare sequences; collective patterns.
Resource / cost	Runaway auto-scaling; orphaned resources; billing spikes; quota exhaustion.	Billing records, provisioning events, utilisation-to-allocation ratios; step changes; sustained low utilisation
Availability	Node failures; crash loops; cascading microservice failures; zone impairment.	Health checks, restart counts, error budgets; collective correlated failures across dependency graph.

Source: Compiled by the authors.

Anomaly Detection Techniques: A Comparative Survey

This section surveys the four principal families of techniques applied to cloud anomaly detection, followed by hybrid and ensemble designs. For each family we describe representative algorithms, underlying assumptions, and strengths and weaknesses as documented on cloud workloads.

Statistical and Time-Series Methods

Statistical techniques model probable behaviour over a time series and flag low-probability observations. The most basic techniques are control-chart methods, examples of which include the three-sigma rule and moving-average charts. CUSUM charts are used to analyse small deviations and detect, for example, memory leaks. For seasonal telemetry, decomposition and outlier-detection approaches employ two classical techniques Holt-Winters exponential smoothing and STL (Seasonal-Trend decomposition using Loess); a related approach is the Seasonal Hybrid ESD method, which combines STL with the generalised extreme studentised deviate test. Autoregressive models (ARIMA/SARIMA) predict the next observation and score the prediction error. Extreme value theory (EVT), as used in the SPOT/DSPOT algorithms, models the tails of the score distribution to obtain thresholds without assuming a specific distribution type (Siffer et al., 2017).

Strengths of statistical approaches include computational efficiency, clarity, and ease of use: an EWMA detector needs constant memory per time series, operates in microseconds, and produces warnings that are easily understood from the displayed deviation. These advantages justify their presence in commercial cloud monitoring systems. Their weaknesses are equally clear: they are typically univariate, so they fail to detect anomalies that are only visible in the relationships among several metrics; they embed strong distributional or stationarity assumptions that elastic cloud workloads cannot meet; and manual tuning of such detectors for each series is simply impossible when there are millions of time series. As shown in comparative evaluations for cloud and web-operations time series, well-tuned statistical approaches can sometimes compete with much more complex deep-learning models (Braei & Wagner, 2020).

Classical Supervised Machine Learning

Where labelled data exist chiefly in security, thanks to public intrusion datasets—supervised classifiers have been applied extensively. Support Vector Machines (SVMs) construct maximum-margin decision boundaries, handle high-dimensional feature spaces well through kernels, and were the workhorse of intrusion-detection research on the KDD Cup 99 and NSL-KDD datasets, typically achieving accuracies above 95% on those benchmarks; their drawbacks are quadratic-to-cubic training complexity, sensitivity to feature scaling, and poor calibration under the severe class imbalance typical of anomalies. The most successful classical methods on tabular network-security data today are decision trees and, especially, tree ensembles Random Forests and gradient-boosted machines such as XGBoost and LightGBM; on CICIDS2017 and UNSW-NB15, gradient-boosted trees yield F1 scores in the 0.97–0.99 range for well-represented attack classes, are fast to train, tolerant of mixed feature types without scaling, and offer feature-importance metrics to assist analyst triage (Sharafaldin et al., 2018; Moustafa & Slay, 2015). While k-Nearest Neighbours is a simple non-parametric method, its inference cost of

O(n) per query makes it impractical in streaming cloud telemetry without approximate indexing. Naïve Bayes remains a quick, lightweight model constrained by its independence assumption on cloud features.

Two systemic caveats attach to the entire supervised family in the cloud context. First, the closed-world assumption: a classifier trained on known attack classes can, by construction, only recognise those classes; novel ("zero-day") anomalies fall outside its label space, and reported per-class accuracies say nothing about them. Second, benchmark leakage and unrealistic class balance: many published results rest on datasets whose train/test splits share flows from the same sessions, or whose attack proportions (up to 20–50% anomalous) bear no relation to production base rates of well under 0.1%, at which even a 99% precision figure translates into overwhelming absolute false-alarm volumes. These caveats motivate the semi-supervised and unsupervised families that dominate operational practice.

Unsupervised and Semi-Supervised Learning

Unsupervised techniques assume anomalies are rare and structurally different, and score observations by density, distance, isolation, or reconstruction criteria without labels. Clustering-based detection (k-means, DBSCAN) treats points far from cluster centroids, or points left unclustered by density-based algorithms, as anomalous; DBSCAN's native noise concept fits anomaly semantics naturally, but its distance parameter is difficult to set in high dimensions. Local Outlier Factor (LOF) compares each point's local density with that of its neighbours, detecting anomalies in data with varying density regimes useful when a cloud fleet mixes heterogeneous instance types—but at $O(n^2)$ worst-case cost that limits it to modest sample sizes or index-accelerated implementations. One-Class SVM learns a boundary enclosing normal data in feature space and is a classical semi-supervised choice, though it is sensitive to hyperparameters and scales poorly beyond tens of thousands of points.

Unlike the classical detection paradigm, Isolation Forest has emerged as the default unsupervised technique for cloud monitoring (Liu et al., 2008). Isolation Forest isolates observations using randomised trees; because anomalies are by definition rare and different, they are expected to have shorter paths to termination in such trees, thereby producing anomaly scores. The algorithm's appeal for cloud applications lies in its linear training time, logarithmic inference time, low memory footprint, ability to handle multivariate data, insensitivity to irrelevant features, and lack of tuning parameters. The method has proven to be one of the most effective, alongside several other outlier-detection approaches benchmarked on standard datasets and real-world cloud metrics. Variants of Isolation Forest with sliding windows over time can handle drifting distributions, although only moderate drift. The main limitation of Isolation Forest, and of many unsupervised methods, is limited performance on certain classes of local anomalies, particularly those that do not follow axis-aligned distributions; these shortcomings are alleviated in later extensions. Finally, dimensionality-reduction techniques such as PCA and its robust equivalents can also serve as unsupervised anomaly detectors: they model the distribution of normal points as a low-dimensional subspace and use either reconstruction error or outlier scores as detection statistics. Such methods are particularly useful when the data contain correlated fleet-wide metrics but have limited ability to model more complex structures, which deep autoencoders (Section V-D) alleviate. Streaming versions such as Half-Space Trees and Robust Random Cut Forest (used in Amazon Kinesis anomaly scoring) maintain sketches of the data distribution updated at every observation, operating in a streaming fashion at the expense of some accuracy (Guha et al., 2016).

Deep Learning Methods

Deep learning entered cloud anomaly detection along two routes: representation learning for high-dimensional, multi-modal telemetry, and sequence modelling for temporal and log data. Autoencoders (AEs) are the foundational architecture: trained to reconstruct normal data through a low-dimensional bottleneck, they score anomalies by reconstruction error, generalising PCA to non-linear manifolds. Variational autoencoders (VAEs) add a probabilistic latent space and score by reconstruction probability, improving calibration; Xu et al.'s (2018) Donut demonstrated VAE-based detection on seasonal web/cloud KPIs with F-scores of 0.75–0.90, where statistical baselines scored substantially lower. For temporal telemetry, LSTM (long short-term memory) networks model normal sequence dynamics either as forecasters (score = prediction error) or, combined with autoencoding, as LSTM-AE/encoder-decoder models

that reconstruct windows of multivariate metrics; these dominate published comparisons on server and cloud time-series benchmarks, with typical F1 improvements of 5–15 points over Isolation Forest on contextual and collective anomalies (Braei & Wagner, 2020; Malhotra et al., 2016). OmniAnomaly extended this line with stochastic recurrent modelling of multivariate server telemetry, reporting F1 around 0.86–0.89 across the SMD server-machine dataset (Su et al., 2019). Convolutional networks (including temporal convolutional networks) exploit local patterns in metric windows at lower training cost than recurrence, and graph neural networks (e.g., GDN) explicitly model inter-metric or inter-service dependency structure, improving detection and localisation of collective anomalies in microservice topologies.

Generative adversarial networks (GANs) train a generator of normal data against a discriminator and score anomalies by a combination of discrimination and reconstruction; despite promising results (e.g., MAD-GAN on multivariate sensor and infrastructure data), training instability has limited their operational adoption. Transformer architectures are the current research frontier: attention-based models such as Anomaly Transformer and TranAD report state-of-the-art F1 on public multivariate benchmarks (frequently 0.88–0.94 on SMD, SMAP, and MSL) (Tuli et al., 2022), and log-sequence models from DeepLog's LSTM over parsed log templates (Du et al., 2017) to BERT-style pretrained models such as LogBERT (Guo et al., 2021) achieve high precision and recall (commonly above 0.95 on HDFS and BGL log benchmarks) in detecting anomalous execution paths in system logs. Two important caveats temper these numbers. First, several influential analyses have shown that widely used point-adjustment evaluation protocols dramatically inflate reported F1 for deep time-series detectors, and that under strict protocols the advantage over simple baselines narrows considerably (Kim et al., 2022). Second, deep models impose costs that matter in production: GPU-hours of training per model, per-workload retraining under drift, hyperparameter sensitivity, and near-opaque decision logic that complicates on-call triage and post-incident review.

Hybrid and Ensemble Approaches

Because families fail in different ways, production systems increasingly combine them. Common hybrid patterns include: (i) cascade designs, in which a cheap statistical or Isolation Forest first stage screens the full telemetry stream and a heavier deep-model rescoring stage examines only the small flagged fraction, cutting compute by orders of magnitude at negligible recall cost; (ii) feature-learning hybrids, in which an autoencoder or embedding model learns representations that a classical detector (e.g., one-class SVM or k-NN on the latent space, as in Deep SVDD-style designs) then scores; (iii) score ensembles, which aggregate normalised scores from diverse detectors by averaging, maximisation, or learned weighting, exploiting the low error correlation between, say, density-based and reconstruction-based detectors; and (iv) statistical-neural compositions such as decomposing seasonality with STL and applying an LSTM only to the residuals. Empirically, heterogeneous ensembles deliver the most consistent precision–recall trade-offs across mixed anomaly types, at the price of additional operational complexity and the need for score calibration. Cloud providers' managed offerings (e.g., AWS DevOps Guru/Lookout for Metrics, Azure Anomaly Detector, Google Cloud operations suite) are, from public documentation or inference, ensembles of exactly this character.

Benchmark Datasets and Evaluation Methodology

Datasets

Comparability across studies depends on shared datasets, of which several families dominate. In network security, KDD Cup 99 and its cleaned successor NSL-KDD remain ubiquitous despite their age (derived from 1998 traffic) and well-documented artefacts; UNSW-NB15 (2015) and CICIDS2017/CSE-CIC-IDS2018 provide modern attack mixes DDoS, brute force, botnet, infiltration, web attacks captured with realistic background traffic, and the CIC datasets in particular have become the default for cloud-oriented intrusion detection because CSE-CIC-IDS2018 was generated on AWS infrastructure. In performance and telemetry, the Yahoo S5 Webscope benchmark supplies labelled real and synthetic KPI series; the Numenta Anomaly Benchmark (NAB) provides streaming series with a time-aware scoring protocol; SMD (Server Machine Dataset), SMAP, and MSL supply labelled multivariate machine telemetry widely used for deep detectors; and the Google and Alibaba cluster

traces provide production-scale scheduler and utilisation data unlabelled, and therefore used mainly for realistic workload modelling and injected-anomaly evaluation (Reiss et al., 2012). Log-based detection standardly uses the HDFS and BGL corpora from the LogHub collection.

Table 2: Benchmark Datasets Commonly Used for Cloud Anomaly Detection Research

Dataset	*Year*	*Type/Scale*	*Notes and Known Limitations*
NSL-KDD	2009	Network flows, ~150k records, 4 attack families	Cleaned KDD99; outdated traffic; still saturated (many methods >99% accuracy)
UNSW-NB15	2015	~2.5M flows, 9 attack families, 49 features	Modern attacks; moderate class overlap makes it harder than NSL-KDD
CICIDS2017 / IDS2018	2017–18	Flow features; IDS2018 generated on AWS	Realistic background traffic; label noise and duplicated flows documented
Yahoo S5	2015	367 labelled KPI time series (real + synthetic)	Univariate; short series; synthetic portions easy
NAB	2015	58 streaming series, time-aware scoring	Rewards early detection; small size limits deep-model training
SMD / SMAP / MSL	2018–19	Multivariate server & spacecraft telemetry	Standard for deep detectors; point-adjust evaluation inflates F1
Google / Alibaba traces	2011–19	Production cluster traces, thousands of machines	Unlabelled; used with injected anomalies or as workload models
HDFS / BGL (LogHub)	2009–	Labelled system-log corpora	Standard for log anomaly detection; template-parsing sensitivity

Source: Compiled by the authors.

Evaluation Metrics

Because anomalies are rare, raw accuracy is uninformative: a detector that never alerts achieves 99.9% accuracy at a 0.1% base rate. Useful metrics include precision (the proportion of detections that are indeed anomalies), recall or detection rate (the proportion of true anomalies detected), their harmonic mean (F1), and the false-positive rate. The threshold-free approach uses the ROC curve and area under the ROC (AUC-ROC). Yet, under the strong class imbalance, precision–recall curve and AUC-PR are much more discriminating and better suited to the task. In addition, time series introduce some subtleties: range-based precision/recall evaluates the partial overlap of the predicted and actual anomalous ranges; NAB scoring system emphasizes early detection in the interval; and the point adjustment, that assumes an entire anomalous interval as detected if at least one point within the interval was recognized, introduces such a bias in the F1-score calculation, that the random scoring becomes almost as good as that, prompting recent research to suggest stricter or affiliation-based procedures (Kim et al., 2022). The operational approach also incorporates the detection latency (time between the occurrence of the anomaly and its detection), alerts per operator day, cost of training/inference, and memory utilization – factors not taken into consideration in academic research, but important in practical applications.

Comparative Analysis

Comparison Framework

In this part, we will answer RQ4 using a comparative evaluation of technique categories based on eight characteristics that have been chosen based on their importance in deploying cloud systems, using the 78 studies identified in Section III: (C1) anomaly detection accuracy on point, contextual, and collective anomalies; (C2) false positives at production base rates; (C3) labelling requirements; (C4) computation cost of training and inference; (C5) scalability to fleet-wide, high dimensional telemetry data; (C6) ability to cope with concept drift and elastic baselines; (C7) interpretability of alerts; and (C8) real-time applicability. Table 3 presents a summary of our qualitative analysis.

Table 3: Qualitative Comparison of Technique Families Across Deployment-Relevant Criteria

Criterion	*Statistical*	*Supervised ML*	*Unsupervised ML*	*Deep Learning*
Point anomalies	Good	Very good (known classes)	Very good	Very good
Contextual anomalies	Moderate (seasonal models)	Moderate	Moderate	Very good (LSTM/ Transformer)
Collective anomalies	Weak	Moderate	Moderate	Very good (sequence/graph models)
False-positive burden	High unless tuned per series	Low on known classes; blind to novel	Moderate	Low–moderate; threshold sensitive
Label requirement	None	Full labels	None	None–normal-only (AE) / labels (supervised DL)
Train / inference cost	Negligible / µs	Moderate / fast	Low–moderate / fast	High (GPU) / moderate
Scalability (fleet-wide)	Excellent per-series; weak multivariate	Good	Good (iForest, RRCF)	Good with serving infra; costly to retrain
Drift adaptability	Good (EWMA, DSPOT)	Poor without retraining	Moderate (windowed retrain)	Poor–moderate; retraining expensive
Interpretability	High	Moderate–high (trees)	Moderate	Low (post-hoc XAI needed)
Streaming suitability	Excellent	Good (inference)	Good (streaming variants)	Moderate

Source: Compiled by the authors.

Detection Effectiveness

With regard to effectiveness, there is definitely an order when taking into account the existing studies' findings. With respect to simple point anomalies within one-dimensional time series (spikes and level shift), all the detection models work equally well, with little improvement from deep learning the F1 score is no better than 5% that of statistical baselines and Isolation Forest. For contextual anomalies in seasonal/autocorrelated time series, sequential deep-learning detectors (LSTM autoencoder, VAE) and the classical Holt-Winters exponential smoothing show clear superiority over point estimators: on operational KPI datasets of this type, LSTM-based detectors report F1 scores of 75–90%, compared with 55–70% for Holt-Winters-class methods (Xu et al., 2018; Malhotra et al., 2016). For collective anomalies anomalous patterns in multivariate time series or in logs with dependent events the gap is even larger: methods that explicitly model temporal or other dependencies (LSTM-AE, OmniAnomaly, graph-based detectors, LogBERT-class) report F1 scores of 85–95% on SMD/SMAP/MSL and above 95% on HDFS/BGL logs, while standard point detectors are ineffective (Du et al., 2017; Guo et al., 2021). In network-traffic security, gradient-boosted decision trees still dominate for flow-level features (F1 of 97% or better on CICIDS2017 for most classes), while deep-learning detectors are at best comparable on tabular features (Sharafaldin et al., 2018; Moustafa & Slay, 2015). Overall, all results should be read with the caveat of Section VI-B that scores are inflated by point-wise adjustment; more conservative, standard evaluations decrease F1 by tens of points, with some methods performing nearly on par with random guessing, though the relative effectiveness ranking is mostly preserved for the task categories described above (Kim et al., 2022).

Operational Cost, Drift, and Interpretability

Effectiveness rankings reverse almost exactly when cost, flexibility, and interpretability are the criteria. The statistical approach has near-zero overhead per time series and adapts automatically on the fly (EWMA/DSPOT). Its output is self-explanatory; even scaled naively to a million time series, it remains virtually cost-free. Isolation Forests and their streaming cousins strike a good balance: CPU-minutes of training on millions of examples, milliseconds

per inference, and re-training under drift at fixed intervals accounting for their prominence in cloud-managed services. At the other end of the spectrum lie deep neural detectors, which take per-workload GPU-hours to train, are highly sensitive to tuning, and must retrain through their own monitoring pipeline; in elastic workloads with constantly shifting baselines, an un-retrained LSTM detector would degrade within a few days in false-positive rate. Interpretability lies along the same axis. Tree ensembles provide feature importances; Isolation Forest scores can be decomposed into feature splits; and deep models must be interpreted post hoc (SHAP, attention inspection, per-dimension reconstruction-error decomposition). In on-call practice, an alert an engineer cannot interpret in seconds is frequently an alert ignored, which is why several industrial reports describe deliberately replacing more accurate deep detectors with simpler ones after measuring end-to-end incident-response outcomes rather than offline F1.

Synthesis: Matching Techniques to Cloud Scenarios

The evidence is more comparative than prescriptive, but it still favours certain options depending on context. For metric screening at very large scale with low latency budgets, statistical detectors with automatically chosen thresholds (SH-ESD, DSPOT) or streaming isolation ensembles (RRCF, Half-Space Trees) are the most rational first step, since they handle heavy data streams in real-time. For multivariate host or service telemetry, where collective and contextual anomalies are of interest and the compute budget is moderate, Isolation Forest ensembles and LSTM/VAE autoencoders are the best options, the latter preferable when there is enough data to benefit from training deep models. For security analytics on flow features with labelled data, gradient-boosted trees offer the best performance-to-effort ratio and can be complemented by an unsupervised channel to detect zero-day attacks. For log and trace analysis, template-based sequence models (DeepLog-class LSTM or LogBERT-class transformers) are without serious classical competition. And for organisations able to operate them, cascaded hybrids cheap screening followed by deep rescoring, with heterogeneous score ensembling consistently deliver the best precision at fixed recall, aligning with the architecture of commercial cloud anomaly services. Table 3's pattern can be stated compactly: deep learning buys detection power for contextual and collective anomalies at the price of cost, drift-fragility, and opacity; classical and statistical methods buy operability at the price of expressiveness; and hybrids trade engineering complexity for the best of both.

Discussion

Interpretation of Findings in Relation to the Research Questions

In general, all four research questions show a field in which detection accuracy has developed more than operational complexity. Regarding RQ1, as presented in Section IV, the taxonomy suggests that cloud anomalies are more diverse than a singular occurrence; there can be different classifications of cloud anomalies by the structure type, layer, and domain; contextual and collective anomalies, unlike simple point anomalies, are the most common and impactful in the elastic multi-tenant environment. Thus, the answer to RQ2 is the following: the comparative study presented in Section V shows that technique efficiency is unique for anomalies they target since the techniques based on sequence- and graph-aware deep models outperform other algorithms in terms of detecting contextual and collective anomalies, whereas lightweight statistical and isolation techniques keep being efficient for detecting point anomalies at much lower cost. This study extends prior research on this issue in the way that Chalapathy and Chawla (2019), Pang et al. (2021), and Soldani and Brogi (2022) provide taxonomies on the basis of model architecture and failure analysis workflow, respectively.

The answer to RQ3 is somewhat more negative. Section VI demonstrates that the benchmarking conventions used to make claims of accuracy saturated legacy datasets, unrealistically optimistic anomaly base rates, and point-adjusted F1 consistently favor the same deep neural network architectures that rule over published leaderboards, reinforcing the issues noted by Kim et al. (2022). This changes the response to RQ4: when one takes into account the costs and constraints of detection performance (Table 3) beyond simple accuracy, no one family reigns supreme. Rather, there is a case for a contingent approach to anomaly detection, where the "best" algorithm is determined solely relative to a set anomaly class, telemetry modality, label scheme, and budget, and where mixed/hybrid architectures perform

best at balancing the competing aims. Thus, viewed as a whole, the present literature review makes the connection between the descriptive problem of what sorts of anomalies exist, and the prescriptive problem of detecting them cost-effectively one which narrative reviews cannot demonstrate through implicit selection bias.

Implications for Research, Practice, and Policy

Scientifically, the results suggest that additional gains in accuracy achieved on saturated datasets have limited scientific utility, and that meaningful contributions to the field at this time are primarily made through evaluation realism, drift-resistant learning, and explainable modeling (Section X). By providing a criteria-based process of article selection, this review thus additionally provides a reusable framework for the evaluation of future primary research, one that allows future studies to identify how their contributions fit into the growing body of literature as opposed to merely adding their results to yet another leaderboard.

Practically, the review has significant implications for economics and operations. Given that the correlation between effectiveness and cost ranking is almost perfectly inverse (Section VII-C), practitioners have a clear basis upon which to select a particular approach: statistical and isolation-based detection for the purposes of large scale low-latency filtering; deep sequence and graph learning models saved exclusively for the high value cases of multivariate and log data where accuracy benefits make up for their training, drift, and explanation overheads; and cascaded approaches wherever sufficient engineering resources are available. As false positive alert fatigue not offline F1 is repeatedly noted as the most common cause of failure of industrial monitoring, the review's focus on these aspects clearly translates into cost reduction and improved incident handling results.

When it comes to education and policy making, the classification system and the comparative approach offer a vocabulary-based structure that can be useful for curricula on cloud operation and AIOps, while the discussion on the problem of adversarial robustness and the concentration of the capability to manage detection in a few hyperscale operators (Sections V-E and IX-E) are important in the context of ongoing debates on critical-infrastructure resilience, security assurance, and data-sovereignty regulation in cloud monitoring. This paper tries to fill the gap between anomaly detection theories and practice.

Limitations of the Study

Apart from the issues related to threats to validity arising out of the review process highlighted in Section III-F, some aspects of the limitations associated with the study need to be recognized in order to interpret its findings appropriately. First of all, the synthesis carried out within this project is qualitative and criteria-based, not quantitative meta-analysis, because the significant differences in datasets, methods and metrics used in the primary studies made the statistical combination of effect sizes impossible, and thus the comparison of ranking provided in Table 3 can be regarded as a synthesized interpretation of the information obtained. The second limitation is associated with the fact that the values of accuracy quoted from the primary sources are prone to the benchmarking biases and point adjustment effects outlined in Section VI. Finally, the review was carried out among the peer-reviewed journal and conference publications between 2015 and 2026 across five databases and the grey literature, including industrial white papers and vendor engineering reports – where a lot of the practical experiences have been accumulated – was not included in the formal scope of the search.

Challenges and Open Issues

Label Scarcity and Ground-Truth Quality

Labels for production cloud anomalies have never been created; when labels are obtained in postmortems for the incidents, they correspond to anomalies identified by humans, hence leading to biased supervised learning algorithms for already detected anomaly classes. Labeling errors in publicly available benchmarks exist; for example, CICIDS2017 is known to have labeling noise issues, while Yahoo S5 and SMD have disputed parts. It is usually seen that a minor difference in F1 scores between algorithms published in papers stems from labeling errors.

Concept Drift and Non-Stationarity

Elasticity entails non-stationarity: auto-scaling, deployments, and traffic and seasonality shifts mean that

"normal" keeps changing all the time. Anomaly detection needs to be able to tell a genuine anomaly from a shift in the baseline – an issue of semantics, rather than mere statistics – and may profit from side information like deployment and scaling events. The available methods vary from forgetting schemes (sliding windows, exponential decay) to explicit drift detection (ADWIN, DDM) that triggers retraining to continual learning for neural networks; none of them provides a satisfactory solution to the problem of abrupt and deliberate regime shifts typical of clouds, and even retraining opens a window of vulnerability.

Scalability, Cost, and Energy

A small cloud estate produces millions of active time series, while hyperscalers produce billions. Deep-learning models per individual series become too costly to run, leading to trade-offs between global models shared across series (with associated generalisation problems), hierarchically composed screening pipelines, and aggressive telemetry down-sampling that destroys anomalies. Monitoring infrastructure must avoid becoming a denial-of-service attack on the target infrastructure: compute for detection, storage of windows and models, and alert pipelines must all be designed under tight constraints.

Interpretability and Actionability

An anomaly score is only the starting point of the workflow; engineers need to know which metrics, services, or log lines generated the score, whether the pattern corresponds to a known failure mode, and what the next step should be. Post-hoc explanations borrowed from supervised XAI (SHAP values for anomaly scores, reconstruction-error attribution, attention visualisation) offer partial solutions, yet the faithfulness of such approaches is disputed for one-class and reconstruction models, and there is no standard for explaining anomaly-detection results.

Adversarial Robustness and Evasion

Threat-sensitive detectors face adaptive opponents that can probe thresholds, emulate typical traffic distributions, poison training windows (tampering with the normal pattern prior to an attack), and craft perturbations that reduce reconstruction error. The literature on adversarial machine learning for anomaly detection is less mature than its supervised counterpart, and cloud characteristics such as multi-tenancy and public knowledge of managed detectors' behaviour increase the attack surface.

Evaluation Realism

Finally, the field's measurement instruments distort its conclusions. Saturated legacy datasets, unrealistic anomaly base rates, point-adjusted scoring, and the absence of cost, latency, and alert-volume reporting jointly favour method families whose advantages are least relevant to production. Community moves toward strict evaluation protocols, streaming benchmarks with injected but realistic faults, and operational metrics alongside F1 are necessary conditions for the literature's comparisons to predict deployment outcomes (Kim et al., 2022).

Future Research Directions

- *Foundation and Transformer Models for Telemetry*: Pretraining large sequence models across heterogeneous metrics, logs, and traces, then adapting them cheaply per workload (few-shot or zero-shot anomaly detection), analogous to the trajectory of language models; early log-pretraining results (LogBERT-class) suggest substantial headroom.
- *Federated and Privacy-Preserving Detection*: Training shared detectors across tenants, regions, or organisations without centralising raw telemetry, using federated averaging with differential privacy particularly relevant for cross-cloud threat intelligence.
- *Multi-Modal and Cross-Layer Fusion:* Joint models over metrics, logs, traces, and topology that detect anomalies invisible in any single modality and localise them across layers, likely via graph neural architectures over service-dependency graphs.
- *Drift-Native Learning:* Detectors designed around change continual learning without catastrophic forgetting, change-point-aware baselining, and integration of deployment/scaling event streams as exogenous context rather than periodic wholesale retraining.

- *Explainable and Causal Anomaly Analysis:* Moving from post-hoc attribution to models with built-in explanatory structure, and from correlation to causal root-cause inference over dependency graphs.
- *Serverless and Edge-Cloud Detection:* Methods for ephemeral execution units and edge nodes with millisecond lifetimes and kilobyte memory budgets, where per-instance history does not exist.
- *AIOps Integration and Closed-Loop Remediation*: Coupling detection with automated diagnosis and safe remediation (rollback, scaling, isolation), with detection quality measured by end-to-end incident outcomes rather than offline scores.
- *Standardised, Realistic Benchmarking:* Community benchmarks with production-scale base rates, labelled drift, cost/latency reporting requirements, and strict scoring protocols.

Conclusion

This paper has presented a systematic literature review and comparative analysis of anomaly detection in cloud computing environments. Following Kitchenham's guidelines and PRISMA reporting, a structured search of five digital libraries covering 2015–2026 yielded 78 primary studies, from which we derived a taxonomy of cloud anomalies (RQ1), a comparative synthesis of statistical, supervised, unsupervised, deep, and hybrid techniques (RQ2), a critical review of benchmark datasets and evaluation pitfalls (RQ3), and a criteria-based comparison oriented to production deployment (RQ4). The key point is that asking "which is better?" is meaningless unless the anomaly class, telemetry type, label availability, and budget are fixed. Deep-learning models dominate for contextual and collective anomalies in high-quality temporal and log data, but their training cost, drift sensitivity, and lack of interpretability are not side effects they are inherent production limitations. Classical unsupervised techniques, particularly Isolation Forest and its streaming versions, clearly lead in accuracy-to-cost ratio on general multivariate telemetry, whereas highly automated statistical detectors retain an unchallengeable position for massive-scale primary filtering. Supervised classifiers, with gradient-boosted trees at the forefront, perform best on labelled data but are intrinsically incapable of detecting novelties. Most production-ready architectures should therefore be hybrid, using cheap initial filtering, selective re-scoring, and heterogeneous ensembles. Moving forward, we are less likely to see minor improvements over saturated benchmarks and more likely to see breakthroughs in evaluation, drift-resistant and interpretable modelling, and incident-driven detection systems.

Acknowledgement

The authors gratefully acknowledge the Balaji Institute of Modern Management, Sri Balaji University, Pune, for the institutional support and academic environment that made this research possible. The authors received no specific grant for this work from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

AI Disclosure

During the preparation of this manuscript, the authors used generative artificial intelligence tools solely to assist with language editing, readability, and formatting. All research design, literature selection, analysis, interpretation, and conclusions are the authors' own. After using these tools, the authors reviewed and edited the content as required and take full responsibility for the content of the publication.

References

Braei, M., & Wagner, S. (2020). Anomaly detection in univariate time-series: A survey on the state-of-the-art. *arXiv preprint arXiv:2004.00433*. https://doi.org/10.48550/arXiv.2004.00433

Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*. https://doi.org/10.48550/arXiv.1901.03407

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys, 41*(3), 1–58.

Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)* (pp. 1285–1298). ACM. https://doi.org/10.1145/3133956.3134015

Fernando, T., Gammulle, H., Denman, S., Sridharan, S., & Fookes, C. (2021). Deep learning for medical anomaly detection – A survey. *ACM Computing Surveys, 54*(7). https://doi.org/10.1145/3464423

Guha, S., Mishra, N., Roy, G., & Schrijvers, O. (2016). Robust random cut forest based anomaly detection on streams. In *Proceedings of the 33rd International Conference on Machine Learning* (pp. 2712–2721). PMLR.

Guo, H., Yuan, S., & Wu, X. (2021). LogBERT: Log anomaly detection via BERT. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). IEEE.

He, P., Zhu, J., Zheng, Z., & Lyu, M. R. (2017). Drain: An online log parsing approach with fixed depth tree. In *Proceedings of the IEEE International Conference on Web Services (ICWS)* (pp. 33–40). IEEE.

Ibidunmoye, O., Hernández-Rodriguez, F., & Elmroth, E. (2015). Performance anomaly detection and bottleneck identification. *ACM Computing Surveys, 48*(1), 1–35.

Kim, S., Choi, K., Choi, H.-S., Lee, B., & Yoon, S. (2022). Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 7194–7201). AAAI Press.

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (EBSE Technical Report EBSE-2007-01). Keele University and Durham University.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *Proceedings of the Eighth IEEE International Conference on Data Mining (ICDM)* (pp. 413–422). IEEE.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). LSTM-based encoder-decoder for multi-sensor anomaly detection. In *Proceedings of the ICML Anomaly Detection Workshop.*

Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing* (NIST Special Publication 800-145). National Institute of Standards and Technology.

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. In *2015 Military Communications and Information Systems Conference (MilCIS)* (pp. 1–6).

Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine learning for anomaly detection: A systematic review. *IEEE Access, 9*, 78658–78700.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., . . . Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, *372*, Article n71.

Pang, G., Shen, C., Cao, L., & van den Hengel, A. (2021). Deep learning for anomaly detection: A review. *ACM Computing Surveys, 54*(2), 1–38.

Reiss, C., Tumanov, A., Ganger, G. R., Katz, R. H., & Kozuch, M. A. (2012). Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In *Proceedings of the ACM Symposium on Cloud Computing (SoCC)*. ACM.

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP)* (pp. 108–116).

Siffer, A., Fouque, P.-A., Termier, A., & Largouët, C. (2017). Anomaly detection in streams with extreme value theory. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 1067–1075). ACM.

Soldani, J., & Brogi, A. (2022). Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey. *ACM Computing Surveys, 55*(3), Article 62.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., & Pei, D. (2019). Robust anomaly detection for multivariate time series through stochastic recurrent neural network (OmniAnomaly). In *Proceedings of*

the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) (pp. 2828–2837). ACM.

Tuli, S., Casale, G., & Jennings, N. R. (2022). TranAD: Deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment, 15*(6), 1201–1214.

Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., Chen, J., Wang, Z., & Qiao, H. (2018). Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In *Proceedings of the 2018 World Wide Web Conference* (pp. 187–196). Association for Computing Machinery.