

Architecting Behavioural Anomaly Detection for Cloud-Scale Security Operations: An Interpretable Statistical Framework for SOC-Driven Cyber Threat Analytics

Justice Opara-Martins, PhD
EU-EC Digital4Business (D4B) Advanced Tech Consortium
Linköping University, SE-581 83 Linköping, Sweden

ABSTRACT

Cloud-native enterprises generate high-volume behavioural telemetry across identity, session, file-access, network, and privilege-management systems. Signature-based controls remain necessary but may miss compromised legitimate accounts and low-and-slow deviations that do not match known indicators. This paper develops an interpretable statistical framework for behavioural anomaly triage within Security Operations Centre (SOC) workflows. The framework comprises four stages: telemetry preparation, entity-specific baseline construction, standardised deviation scoring, and SOC correlation and response.

The method uses means, standard deviations, Z-scores, and explicitly defined thresholds to identify observations requiring investigation. A cold-start procedure moves new entities from documented heuristic thresholds to statistical baselines only after a prespecified minimum history is available. The framework is demonstrated using a 30-day synthetic enterprise dataset containing approximately 500 user entities and more than 100,000 activity events, with behavioural indicators including authentication, access latency, privilege escalation, and failed authentication activity. A worked example shows how an unusual login count can be transformed into an auditable anomaly score. The study does not claim that this demonstration establishes population-level detection accuracy or production-scale performance: the supplied evidence does not include independently labelled attack events, a confusion matrix, comparative baselines, or reproducible MTTD/MTTR calculations. Its contribution is therefore a transparent and extensible detection architecture that can be evaluated in a subsequent labelled, multi-source cloud experiment.

General Terms

Computer Science; Cloud Computing; Cybersecurity; Artificial Intelligence; Data Analytics; Security Operations Management; Regulatory Technology Governance.

Keywords

Anomaly detection; User and Entity Behaviour Analytics; Security Information and Event Management; statistical cyber analytics; Z-score; cloud security; SOC operations; digital resilience.

1. INTRODUCTION

1.1 Background of the study

Cloud migration has expanded the volume, heterogeneity, and velocity of security telemetry available to enterprise defenders. Identity providers, cloud audit trails, container orchestrators, API gateways, endpoint systems, and business applications generate behavioural traces that must be interpreted in context. The operational challenge is not merely collecting more logs; it

is distinguishing legitimate variation from activity that warrants investigation.

Signature-based detection is effective against known indicators, but legitimate credentials can be used in credential compromise, insider misuse, and lateral movement. Behavioural anomaly detection addresses a complementary question: does the current activity depart materially from the historical behaviour expected for this entity, role, resource, or time window? Survey research positions statistical, proximity-based, clustering-based, and machine-learning approaches as distinct families of anomaly detection [1,2,10]. Statistical methods remain operationally valuable where analysts require an explanation for each alert, particularly in regulated environments in which decisions must be auditable [3,16].

This paper develops a statistical baseline framework for SOC-driven anomaly triage. Its central proposition is deliberately bounded: transparent statistical scores can provide an auditable prioritisation signal when they are calculated against an appropriate baseline and then correlated with contextual security evidence. A statistical deviation is not, by itself, proof of malicious intent.

1.2 Research aim

The aim is to design and demonstrate a reproducible, interpretable statistical framework that transforms behavioural telemetry into SOC-investigable anomaly signals while making its assumptions, thresholds, and limitations explicit.

1.3 Objectives

- **O1:** Define a repeatable pipeline from telemetry preparation to anomaly scoring.
- **O2:** Specify baseline, variance, threshold, and cold-start procedures suitable for entity-level monitoring.
- **O3:** Demonstrate the calculations using the supplied synthetic laboratory dataset and a fully disclosed worked example.
- **O4:** Map anomaly scores to SIEM, UEBA, SOAR, and analyst workflows without treating an alert as a confirmed incident.
- **O5:** Establish the evaluation protocol required for a future labelled cloud-scale study.

1.4 Research questions

- **RQ1:** How can transparent statistical baselines identify behavioural deviations in heterogeneous enterprise telemetry?
- **RQ2:** How should Z-score thresholds, confidence

bounds, and cold-start rules be operationalised to support auditable SOC triage?

- **RQ3:** What evidence is required to evaluate detection quality and operational value beyond the identification of statistical outliers?

1.5 Hypotheses and analytical propositions

The study distinguishes testable propositions from the current demonstration:

- **H1:** Under a prespecified baseline, observations with larger absolute standardised deviations are more likely to be labelled anomalous by the detection rule.
- **H2:** Contextual correlation of statistical deviation with access latency, privilege change, or related security evidence improves alert prioritisation relative to a deviation score alone.
- **H3:** A dynamic, entity-aware baseline produces fewer false alerts than a pooled static baseline under changing workload conditions.

1.6 Structure of the paper

Section 2 reviews anomaly detection, UEBA, interpretability, and SOC integration. Section 3 describes the methodological philosophy and research design. Section 4 defines the statistical model, cold-start logic, and operational pipeline. Section 5 describes the synthetic dataset and experimental procedure. Section 6 reports the worked demonstration and states the limits of the available evidence. Section 7 addresses validity, reproducibility, and triangulation. Section 8 discusses SOC and regulatory implications. Section 9 presents recommendations and future evaluation requirements. The appendices provide the data dictionary, pseudocode, reporting template, and figure-production specifications.

2. LITERATURE REVIEW AND THEORETICAL BACKGROUND

2.1 Statistical anomaly detection

Anomaly detection is concerned with identifying observations that differ from a reference distribution or behavioural pattern. Chandola et al. classify approaches into statistical, proximity-based, clustering-based, and classification-oriented methods [1]. Patcha and Park describe anomaly detection as a complement to misuse and signature-based detection because it can surface previously unseen patterns [2]. Statistical models are particularly attractive as a first analytical layer because their assumptions and decision rules can be communicated to analysts.

The limitation is equally important: a deviation from a historical distribution may arise from a legitimate change in role, workload, business process, time zone, or deployment state. Statistical significance and security significance are therefore not equivalent. A production detector should treat a baseline deviation as a triage signal that is enriched by context, not as an automated declaration of compromise.

2.2 UEBA, SIEM, and SOC operations

UEBA systems construct behavioural profiles for users and other entities; SIEM systems aggregate and correlate telemetry; and SOC analysts investigate, contain, and document alerts. The framework proposed here is an analytical component within that architecture, not a replacement for SIEM, endpoint controls, threat intelligence, or analyst judgement. The operational value of a score lies in its traceability: the analyst

should be able to see the entity, feature, baseline window, reference population, deviation, threshold, and contextual evidence that produced the alert.

2.3 Machine learning and explainability

Machine-learning methods can model nonlinear, temporal, and relational behaviour, but operational deployment introduces challenges involving training data, labels, drift, computational cost, interpretability, and adversarial manipulation [3,10,12,13,16]. A transparent statistical layer can operate as a baseline detector, a monitoring control, or an interpretable comparator for more complex models. The claim is complementary rather than competitive: the statistical framework is a defensible first layer whose performance must be compared with appropriate baselines and machine-learning alternatives.

2.4 Regulatory and governance context

The regulatory relevance of behavioural monitoring is contextual. NIS2 and DORA increase the importance of continuous risk management, monitoring, incident handling, and operational resilience [6,7]. NIST and ISO/IEC 27001 provide governance and control reference points [8,9]. These instruments do not prescribe the specific Z-score method proposed here, nor do they establish that a statistical alert is a compliance finding. The defensible claim is that an auditable detection pipeline can support evidence production, monitoring, and response processes when embedded within an organisation's wider control framework.

2.5 Theoretical framing and Bounded Contribution

The framework is informed by a control-and-observability logic: a Security Operations Centre can act on a behavioural event only when the event is observable, the reference baseline is defensible, the deviation is measurable, and the resulting alert is connected to a governed response decision. The contribution is therefore not the invention of the Z-score, the confidence interval, UEBA, or SIEM integration. Those components are established in anomaly-detection and cyber-analytics research [1–3,10–16]. The contribution is the **explicit architectural integration** of four elements that are frequently discussed separately: entity-aware baseline construction, a governed cold-start and recalibration policy, contextual enrichment of statistical deviations, and a traceable hand-off from detection to SOC investigation.

This is a bounded architectural contribution rather than a claim of methodological priority or universal scalability. The framework is designed to be extensible to heterogeneous cloud telemetry, but the present evidence demonstrates only a synthetic laboratory implementation. Its novelty should therefore be evaluated against three criteria: (i) whether the baseline and threshold assumptions are sufficiently explicit to be reproduced; (ii) whether the statistical score is connected to operational context rather than treated as a security verdict; and (iii) whether the architecture specifies how adaptation, evidence preservation, and analyst oversight constrain automated response. A future comparative study should test the framework against pooled statistical baselines, peer-group baselines, dynamic baselines, and selected machine-learning comparators.

Accordingly, the paper makes no “first,” “unique,” or “irrefutable” claim. It contributes a **reproducible, interpretable, and operationally situated design pattern** for anomaly triage that is ready for evaluation on independently labelled multi-source cloud telemetry

3. METHODOLOGICAL PHILOSOPHY AND RESEARCH DESIGN

3.1 Methodological philosophy

The study adopts a post-positivist quantitative orientation: behavioural activity is represented through measurable telemetry variables, statistical deviation is defined through explicit calculations, and claims are constrained by the quality of the data and reference labels. The philosophy has a practical corollary: an alerting rule is scientifically useful only when another analyst can reproduce its input, baseline, calculation, threshold, and classification.

3.2 Research design

The current study is a quantitative computational demonstration conducted on a simulated enterprise environment. It has four stages:

1. **Data acquisition and preparation:** parse, normalise, and quality-check telemetry.
2. **Baseline construction:** estimate entity- and feature-specific location and dispersion parameters over a defined reference window.
3. **Anomaly scoring:** calculate standardised deviations and apply prespecified thresholds.
4. **Operational analysis:** enrich candidate anomalies with contextual signals and map them to SIEM/UEBA/SOAR workflows.

This design demonstrates computational feasibility and interpretability. It does not, in its present form, establish real-world detection accuracy because the manuscript does not provide independent ground-truth labels or a complete performance evaluation.

3.3 Unit of Analysis, Reference Construction and Data Leakage Control

The primary unit of analysis is an **entity–feature–time observation**. An entity may be a user, service account, workload identity, device, or other monitored principal; a feature may be login frequency, access latency, session duration, file-access count, failed-authentication count, or privilege-change activity; and the time component is the timestamp or aggregation interval associated with that observation. The supplied laboratory dataset reports authentication timestamps, session and access behaviour, file-access activity, network-access events, privilege escalation, and failed authentication attempts across a 30-day simulated enterprise environment containing approximately 500 user entities and more than 100,000 activity events.

The baseline must be constructed from a **reference window that precedes the observation being scored**. The scored observation is excluded from its own baseline. Where an independent attack or incident label is available, known attack periods are excluded from the reference window or are handled through a documented contamination-sensitivity analysis. This prevents two forms of leakage: self-inclusion, which can dilute the deviation, and attack contamination, which can cause malicious behaviour to be learned as normal.

For count variables, the implementation should aggregate observations at the time resolution defined by the telemetry schema and retain the original event timestamp for investigation. For continuous variables such as access latency and session duration, the implementation should preserve the measurement unit and calculate the baseline on the same unit

used for scoring. The final camera-ready version must state the exact aggregation interval, entity identifier, reference-window length, missing-value rule, duplicate-event rule, and any exclusion applied during preprocessing. Those values are not disclosed in the supplied manuscript and must be inserted from the laboratory analysis record rather than inferred retrospectively.

The analysis should maintain three logically separate partitions where the data permit: (i) a reference partition used to estimate the baseline; (ii) a scoring partition used to generate alerts; and (iii) an evaluation partition containing independent labels or adjudicated analyst outcomes. If the present synthetic exercise does not contain independent labels or separate partitions, it should be described as a computational demonstration and not as a leakage-controlled accuracy benchmark.

3.4 Reproducibility and computational artefacts

A reproducible release should contain: an anonymised or synthetic dataset; a data dictionary; preprocessing code; baseline and scoring code; threshold configuration; environment and dependency information; random seeds where applicable; and the exact outputs used to produce every table and figure. If the underlying laboratory dataset cannot be released, the author should release a statistically equivalent synthetic dataset and a machine-readable analytical notebook that reproduces the worked example and all reported aggregates.

4. STATISTICAL DETECTION FRAMEWORK

4.1 Baseline Construction

For feature values $(x_1, \dots, x_n)$ in a defined reference window, x the baseline mean is:

$$\left[\mu = \frac{1}{n} \sum_{i=1}^n x_i \right]$$

For a population

– style descriptive baseline, the standard deviation is:

$$\left[\sigma_{pop} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \right]$$

For a sample estimate intended to generalise from the reference window, the denominator is $(n - 1)$:

$$\left[s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2} \right]$$

The manuscript must use one convention consistently. The denominator is not a cosmetic detail because it changes the resulting Z-score and alert classification.

4.2 Z-score Normalisation

For an observation (x) , the standardised deviation is:

$$[z = \frac{x - \mu}{\sigma}]$$

where (σ) is replaced by (s) when the sample standard deviation is used. 2.5) identifies a candidate anomaly under the stated model.

It does not establish maliciousness, and it should not be called a universal statistical significance threshold without specifying the distributional and multiple-testing assumptions.

4.3 Confidence Bounds, Prediction Bounds and Operational Bounds

The framework distinguishes three concepts that serve different purposes. A confidence interval quantifies uncertainty around the estimated mean. A prediction interval describes the expected range of a future individual observation. An operational threshold is the decision rule used to route an event for investigation. These should not be presented as interchangeable.

For a reference sample of size (n), with mean (μ) and sample standard deviation (s), a two-sided confidence interval for the mean at confidence level ($1-\alpha$) is:

$$CI_{\{mean\}} = \mu \pm t_{\{1-\frac{\alpha}{2}, n-1\}} \frac{s}{\sqrt{n}}.$$

For an approximately independent future individual observation, the corresponding two-sided prediction interval is:

$$PI_{\{individual\}} = \mu \pm t_{\{1-\frac{\alpha}{2}, n-1\}} s \sqrt{1 + \frac{1}{n}}.$$

The confidence interval is narrower because it describes uncertainty about the mean; it is not, by itself, an appropriate boundary for classifying every future event. The prediction interval incorporates both uncertainty in the estimated mean and the expected dispersion of an individual observation. In high-volume monitoring, a prediction interval may still be insufficient because repeated testing, non-normal distributions, seasonality, peer-group differences, and concept drift can alter the alert rate.

The present framework therefore uses the standardised-deviation rule ($|z| > 2.5$) as a **candidate-alert threshold**, not as proof of statistical or security significance in every telemetry context. The threshold should be selected and sensitivity-tested against a validation set or analyst-adjudicated outcomes. Where labels are unavailable, the manuscript should report alert volume, alert rate, threshold sensitivity, and qualitative analyst review rather than claim a measured false-positive or false-negative rate.

A candidate anomaly is consequently defined as:

$$[Candidate(x) = \{cases\} 1, \& |z(x)| > \tau, 0, \& |z(x)| \leq \tau,$$

Symbol/term	Definition	Specification
(x)	Scored activity observation	A single entity–feature–time value
(n)	Number of reference observations	Must be stated for each baseline
(μ)	Reference mean	$(\mu = \frac{1}{n} \sum x_i)$
(s)	Sample standard deviation	$(s = \sqrt{\frac{1}{n-1} \sum (x_i - \mu)^2})$
($\sigma_{\{pop\}}$)	Population-style descriptive standard deviation	$(\sqrt{\frac{1}{n} \sum (x_i - \mu)^2})$
(z)	Standardised deviation	$(z = \frac{x-\mu}{s})$ when sample convention is used
(τ)	Candidate-alert threshold	($\tau = 2.5$) in the supplied demonstration

$$\} \{cases\}$$

$$Candidate(x) = \{cases\} 1, \& |z(x)| > \tau, 0, \& |z(x)| \leq \tau \{cases\}$$

where (τ) is a prespecified operational threshold and not a universal constant. In the present demonstration, ($\tau=2.5$) is retained as the stated rule. Its operational adequacy remains to be evaluated using independent labels and threshold-sensitivity analysis.

4.4 Cold-Start and Dynamic Recalibration

A new entity may not have sufficient history for a stable individual baseline. The framework therefore uses a staged policy: (i) apply a documented heuristic or peer-group threshold during cold start; (ii) accumulate observations under monitoring; (iii) transition to an entity-specific statistical baseline once a prespecified minimum sample and data-quality condition are met; and (iv) recalibrate under a documented drift policy. The manuscript’s current rule of ($n>30$) is retained only as a proposed operating criterion, not as a universally sufficient sample-size theorem.

Dynamic recalibration must be constrained by change detection, attack-period exclusion, and analyst or governance review. Uncontrolled updating creates a “boiling frog” failure mode in which persistent malicious activity shifts the baseline toward the attack pattern.

4.5 Contextual Risk Scoring

A candidate anomaly should be enriched with signals such as access latency, privilege escalation, failed authentication, resource sensitivity, geographic or temporal inconsistency, and related threat intelligence. The contextual layer does not erase the statistical score; it changes the operational priority. A high-privilege event with ($|z|=2.82$) and a concurrent role change should receive different analyst attention from a similarly unusual event generated by an approved maintenance process.

A production CRS should specify its features, weights, calibration data, and decision thresholds. The present manuscript describes the logic but does not report a fitted CRS or labelled comparison; it therefore presents CRS as an architectural extension rather than a validated result. In **Table 1**, the notation separates statistical unusualness from security significance: the detector produces a candidate anomaly, whereas maliciousness requires independent contextual or incident evidence.

Table 1: Statistical Notation and Decision Rules

Symbol/term	Definition	Specification
Candidate anomaly	Rule-based statistical flag	(
Confidence interval	Uncertainty around the estimated mean	Not an individual-event boundary
Prediction interval	Expected range for an individual future event	Required if individual-event bounds are claimed
Malicious event	Independently adjudicated security outcome	Not equivalent to candidate anomaly
False positive/negative	Label-dependent evaluation outcomes	N/A without independent labels

5. EXPERIMENTAL DATASET AND PROCEDURES

5.1 Dataset Provenance and Schema

The experimental dataset was generated from a simulated enterprise environment rather than collected from a named production organisation. It covers a 30-day period and is described as containing approximately 500 user entities and more than 100,000 activity events. The reported behavioural indicators are authentication timestamps, session duration, file-access frequency, network-access events, access latency, privilege escalation, and failed-authentication attempts.

The dataset is therefore suitable for demonstrating the mechanics of baseline estimation and anomaly scoring, but its provenance places a clear boundary on inference. It does not by itself establish how the detector behaves under production identity churn, multi-cloud clock differences, provider-specific audit schemas, adversarial baseline poisoning, or real incident-response conditions. Those claims require a separately documented evaluation dataset.

Table 2: Dataset Provenance and Scope

Data element	Required operational definition in the final analysis record
Entity identifier	Stable or pseudonymised user, service-account, workload, or device key
Event timestamp	Original event time, time zone, and normalisation rule
Login frequency	Count of authentication events per entity and stated aggregation interval
Access latency	Response or access delay, with unit and censoring rule
Session duration	Elapsed session time, with treatment of open or interrupted sessions
File-access frequency	Count of file-access events and resource-sensitivity field, if available
Network-access event	Event type, source/destination context, and aggregation rule

Data element	Required operational definition in the final analysis record
Privilege escalation	Role or access-level change, with event semantics
Failed authentication	Count, rate, and treatment of repeated failures
Label field	Attack, benign, uncertain, or unavailable; source and adjudication rule

Table 1 establishes the evidentiary boundary of the experiment: it supports a reproducible demonstration of the scoring pipeline, but the supplied record does not support population-level detection-performance or production-scale operational claims.

The dataset is synthetic or simulated and has limited temporal scope. It may support a reproducible demonstration of the calculations, but it cannot by itself establish generalisability to production cloud environments.

5.2 Preprocessing and Quality Controls

Preprocessing comprises timestamp normalisation, schema validation, entity resolution, duplicate-event handling, missing-value treatment, unit harmonisation, and feature aggregation. Raw observations that may represent anomalies are retained unless they are demonstrably malformed; an anomaly detector should not remove the very events it is designed to examine. Every exclusion is recorded with a reason code.

The processing pipeline produces an audit record containing: input event count; events rejected for malformed structure; duplicate count; missing-value count by field; resolved entity count; feature count; reference-window boundaries; scoring-window boundaries; and evaluation-label availability. When labels are absent, the output must state “unlabelled demonstration” and not substitute the anomaly flag for a ground-truth attack label.

The baseline update policy is conservative. A candidate alert is not automatically incorporated into the normal-behaviour reference set until it has passed the applicable review or quarantine rule. This limits baseline contamination and prevents persistent malicious activity from becoming normal through uncontrolled recalibration.

5.3 Experimental procedure

The procedure comprises four sequential stages:

1. Parse and quality-check the telemetry.

2. Construct reference baselines without leakage from the scored observation.
3. Calculate standardised deviations and apply the prespecified rule.
4. Correlate candidate anomalies with contextual indicators and route them to a conceptual SIEM/UEBA/SOAR workflow.

The present experiment should be described as a **demonstration of the detection pipeline**, not as a complete supervised detection benchmark, unless the author supplies labels and outcome metrics.

6. RESULTS AND ANALYTICAL INTERPRETATION

6.1 Worked Calculation

The manuscript provides the illustrative login sequence:

[4,5,6,5,4,5,6,5,4,21.]

The mean is:

[$\mu = 6.5$.]

The sum of squared deviations is 238.5. Two valid conventions are possible:

- using the population standard deviation, $\left(\sigma_{pop} = \sqrt{\frac{238.5}{10}} \approx 4.885 \right)$, which gives $(z_{21} \approx 2.969)$;
- using the sample standard deviation, $\left(s = \sqrt{\frac{238.5}{9}} \approx 5.147 \right)$, which gives $(z_{21} \approx 2.817)$.

The source manuscript reports a Z-score of approximately 2.82, which corresponds to the **sample standard deviation**, while its earlier displayed formula uses denominator (n). The revised manuscript resolves the inconsistency by stating that the

worked example uses (s) with denominator (n-1). The chosen convention must be applied consistently throughout the dataset analysis.

Under a rule of $(|z| > 2.5)$, the observation 21 is a candidate anomaly under either calculation. That conclusion is a classification under a stated rule—not evidence that the event is malicious. Contextual corroboration and analyst review remain necessary.

The worked login sequence is:

[4,5,6,5,4,5,6,5,4,21.]

Its mean is $(\mu=6.5)$, and the sum of squared deviations is 238.5. The source manuscript reports a Z-score of approximately 2.82 for the final observation. That value is obtained when the **sample** standard deviation is used:

$$[s = \sqrt{\frac{238.5}{10 - 1}} \approx 5.147, \\ \text{\\quadquad} \\ z_{21} = \frac{21 - 6.5}{5.147} \approx 2.82.]$$

The revised manuscript therefore uses the sample standard deviation for this worked example and removes the earlier ambiguity created by a population-style denominator. If a population descriptive standard deviation is preferred for a particular production baseline, the implementation must state that convention and recalculate all resulting scores consistently. Under either convention, the observation 21 exceeds the stated candidate-alert rule $(|z| > 2.5)$. It is consequently a **statistical candidate anomaly**, not a confirmed malicious event. The analytical meaning of the example is bounded and operational: the event is sufficiently unusual relative to the selected reference distribution to warrant additional context. The SOC should then inspect privilege state, access latency, resource sensitivity, geographic or temporal consistency, recent maintenance activity, and related threat-intelligence or incident evidence. If corroborating context is absent, the event remains an unusual observation; if corroborating context is present, the alert priority may increase. The statistical score does not determine intent.

Table 3: Worked Login-Deviation Calculation

Observation	Value (x _i)	Deviation (x _i - μ)	Squared deviation
1	4	-2.5	6.25
2	5	-1.5	2.25
3	6	-0.5	0.25
4	5	-1.5	2.25
5	4	-2.5	6.25
6	5	-1.5	2.25
7	6	-0.5	0.25
8	5	-1.5	2.25
9	4	-2.5	6.25
10	21	14.5	210.25

Observation	Value (x_i)	Deviation ($x_i - \mu$)	Squared deviation
Total	65	0	238.50
Derived quantity	Calculation		Value
Number of observations	(n)		10
Mean	(65/10)		6.5
Sample variance	(238.5/(10-1))		26.50
Sample standard deviation	($\sqrt{26.50}$)		5.148 approximately
Final-event deviation	(21-6.5)		14.5
Final-event Z-score	(14.5/5.148)		2.82 approximately
Candidate threshold	(		z
Security interpretation	Independent label required	Candidate anomaly only	

The exact sample standard deviation is approximately 5.1478; rounding to three decimals gives 5.148. The value ($z \approx 2.817$) rounds to 2.82. The final observation satisfies the statistical candidate-alert rule, but the table does not establish maliciousness, a true positive, or a false-positive rate.

6.2 Reported Behavioural Patterns

The supplied manuscript reports three qualitative patterns: authentication events outside expected activity windows, unusually high file-access frequencies, and extended session durations. These patterns are plausible candidates for

investigation, but the revised article should attach each claim to a table containing the number of scored observations, baseline definition, threshold, number flagged, and—if labels exist—true positives, false positives, false negatives, and true negatives.

6.3 Required experimental evaluation table

The current manuscript does not provide the values needed to complete this table. It should therefore be populated from the analysis artefacts rather than estimated rhetorically.

Table 4: Experimental Evaluation

Evaluation quantity	Required definition	Value
Number of entities	Unique monitored users/entities	Approximately 500
Number of events	Scored activity events	>100,000
Reference-window duration	Baseline period	30 days reported; exact split required
Threshold	Rule and direction	(
Alerts	Number and rate flagged	[unavailable]
Labelled anomalies	Independent ground truth	[unavailable]
False-positive rate	FP/(FP+TN)	[N/A]
False-negative rate	FN/(FN+TP)	[N/A]
Precision	TP/(TP+FP)	[N/A]
Recall	TP/(TP+FN)	[N/A]
F1-score	Harmonic mean of precision and recall	[N/A]
MTTD	Operational definition and calculation	[unavailable]
MTTR	Operational definition and calculation	[unavailable]

6.4 Technical Conclusion and Evidentiary Boundary

The available evidence supports the following limited conclusion: the framework can transform a behavioural

observation into an interpretable baseline deviation and can specify how that deviation would enter a SOC workflow. The available evidence does not yet support claims that the detector “reliably” identifies malicious activity, reduces MTTD or MTTR, or performs effectively at cloud scale. Those claims

require labelled events, a comparator, operational outcome measures, and reproducible calculations.

The observed technical result is that the framework can convert a behavioural observation into an interpretable deviation score and specify how that score is carried into a contextual SOC workflow. The worked example demonstrates the calculation and exposes the implementation requirement that the variance convention be fixed and reported. The synthetic dataset description provides a plausible laboratory basis for demonstrating the pipeline, but the supplied manuscript does not provide independently adjudicated attack labels, a complete confusion matrix, a comparative detector, verified alert counts, or reproducible MTTR/MTTD calculations.

The defensible conclusion is therefore not that the system has been proven to detect malicious activity or reduce response time. The defensible conclusion is that the proposed architecture provides a **transparent and auditable anomaly-triage mechanism whose detection performance and operational value remain to be established through a leakage-controlled, labelled, multi-source cloud evaluation**. This conclusion strengthens rather than weakens the contribution: it aligns the claim with the evidence and defines the exact experiment required for the next stage of validation.

6.5 Evaluation Status and Required Evidence

The present study is a computational demonstration rather than a completed supervised detection benchmark. The available materials establish the stated threshold, the worked calculation, the approximate dataset scale, and the candidate behavioural indicators. They do not establish independent labels or the counts required to calculate detection-performance metrics. The following distinctions are therefore maintained:

- an **alert** is an observation satisfying the statistical rule;
- a **labelled anomaly** is an observation independently adjudicated as malicious or otherwise anomalous;
- a **true positive** is an alert matching an independently labelled anomaly;
- a **false positive** is an alert not supported by the independent label;
- a **false negative** is a labelled anomaly not flagged by the detector.

The full evaluation should report:

$$\begin{aligned} [FPR &= \frac{FP}{FP + TN}, \quad \\ FNR &= \frac{FN}{FN + TP}, \quad \\ Precision &= \frac{TP}{TP + FP},] \end{aligned}$$

$$\begin{aligned} [Recall &= \frac{TP}{TP + FN}, \quad \\ F1 &= 2 \frac{Precision \cdot Recall}{Precision + Recall}.] \end{aligned}$$

Where the event rate is highly imbalanced, precision–recall analysis should accompany ROC-oriented summaries. Confidence intervals should be reported for the principal metrics, and performance should be stratified by entity type, feature, workload condition, and threshold. A future experiment should compare at least a pooled static baseline, an entity-specific baseline, a peer-group baseline, and one appropriately selected machine-learning comparator.

No numerical FPR, FNR, precision, recall, F1, MTTR, or MTTR value is inserted here because those values are not present in the supplied evidence. The absence is reported transparently and is part of the study’s limitation rather than concealed by qualitative language.

7. VALIDITY, REPRODUCIBILITY, AND TRIANGULATION

7.1 Internal validity

Internal validity depends on preventing leakage between baseline estimation and scored events, applying the same preprocessing and scoring rules across observations, and fixing thresholds before inspecting outcomes. The final study should include a temporal or entity-level holdout where possible and a sensitivity analysis for alternative baseline windows and thresholds.

7.2 Construct Validity

Construct validity requires clear operational definitions for “normal behaviour,” “anomaly,” “suspicious activity,” “malicious event,” and “incident.” A Z-score measures deviation from a reference distribution; it does not directly measure threat severity. Threat severity is a separate construct requiring contextual evidence or independent labels.

7.3 External Validity

The simulated 30-day dataset and laboratory workflow support methodological illustration but not generalisation to all cloud environments. External validity should be assessed through datasets spanning different sectors, identity architectures, cloud providers, regions, workload patterns, and attack scenarios.

7.4 Triangulation of Data

A stronger evaluation should triangulate four sources:

1. **Telemetry:** authentication, session, file-access, network, privilege, and cloud audit records.
2. **Ground truth:** red-team injections, incident-response labels, adjudicated analyst decisions, or a documented attack simulation.
3. **Operational outcomes:** alert volume, analyst handling time, MTTR, MTTR, escalation rate, and false-alert burden.
4. **Expert interpretation:** blinded analyst assessment of alert explanations and usefulness.

Agreement across these sources would support operational validity. Disagreement should be reported because it may reveal benign-but-unusual behaviour, label noise, or a limitation of the statistical baseline.

7.5 Evidence Chain and Triangulation Protocol

The framework’s validity depends on alignment among four evidence layers. First, telemetry establishes what activity occurred. Second, the statistical baseline establishes how unusual the activity is relative to a defined reference population and time window. Third, independent incident or analyst labels establish whether the event was security-relevant. Fourth, operational records establish whether the alert changed investigation or response performance. These layers must not be collapsed into a single anomaly flag.

A future validation study should therefore triangulate raw telemetry, independently adjudicated attack and benign events, SOC analyst decisions, and operational measures such as alert volume, analyst handling time, escalation rate, MTTR, and

MTTR. Agreement across layers would strengthen construct and operational validity; disagreement would be analytically useful because it may reveal benign behavioural change, label noise, baseline contamination, or contextual features omitted from the model.

The current synthetic demonstration establishes the first two layers in limited form and illustrates the operational hand-off. It does not establish the third and fourth layers because independent labels and reproducible response-time calculations are not supplied. The manuscript consequently presents the architecture as evaluation-ready rather than fully validated. claim.

8. SOC ARCHITECTURE, FINDINGS, AND DISCUSSION

8.1 SOC Operating Model

The proposed operating architecture has six layers: organisational infrastructure; telemetry collection; normalisation and feature extraction; statistical detection;

dashboard and alert correlation; and investigation, containment, and forensics. The statistical engine provides a transparent score. The SIEM provides aggregation and correlation. UEBA provides entity-oriented behavioural context. SOAR may automate bounded actions, but automated containment should be subject to policy, confidence, asset criticality, and analyst oversight. **Figure 1 presents the high-level SOC architecture.** Behavioural telemetry is collected from identity, cloud-audit, endpoint, network, file, and privilege systems. A normalisation and quality-control layer creates a common event representation. The statistical detection plane maintains reference baselines, produces candidate anomaly scores, and preserves the evidence needed to reproduce each alert. A contextual correlation plane adds privilege, resource, temporal, access-latency, and threat-intelligence context. The SOC decision plane routes the evidence bundle to investigation, containment, documentation, or controlled automation. Governance feedback controls baseline recalibration; alerts are not silently incorporated into the normal-behaviour reference set.

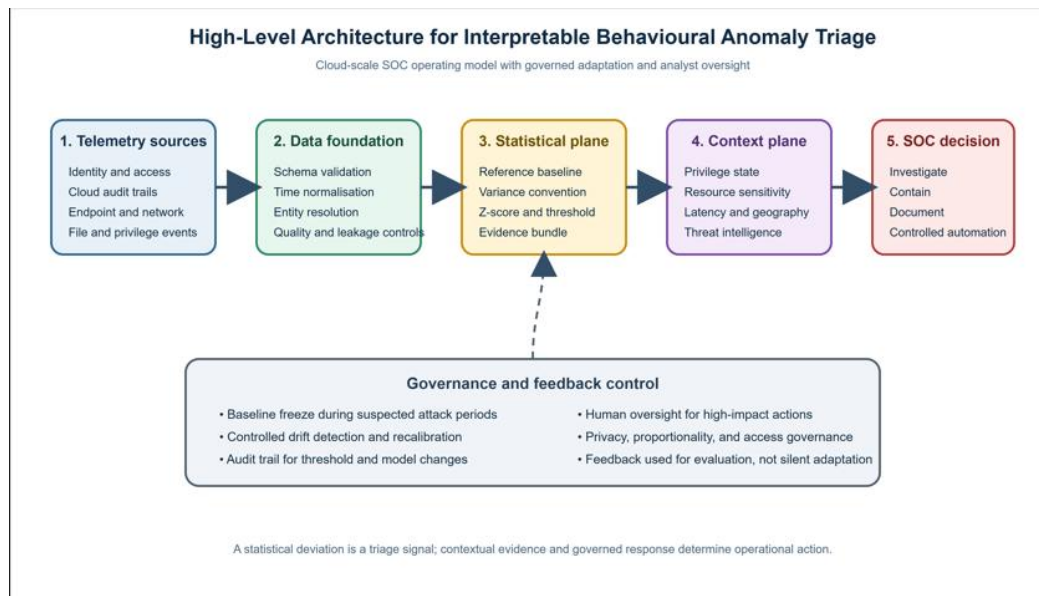


Figure 1: High-level architecture for interpretable behavioural anomaly triage in cloud-scale SOC operations. The architecture separates telemetry, statistical inference, contextual correlation, response, and governed feedback; a statistical anomaly is not treated as a confirmed security incident.

Figure 2 presents the low-level statistical processing path for one entity–feature–time observation. The event is parsed and validated, mapped to an entity and feature schema, and assigned to a reference or scoring partition. The baseline service selects an entity-specific, peer-group, or cold-start baseline according to the governance policy. It calculates the mean and the selected variance convention, produces the

standardised score, applies the threshold, and emits a structured evidence bundle. Context enrichment then produces an operational priority without changing the underlying statistical measurement. The alert is routed to SIEM/UEBA/SOAR interfaces, where human or policy-governed automated action is recorded.

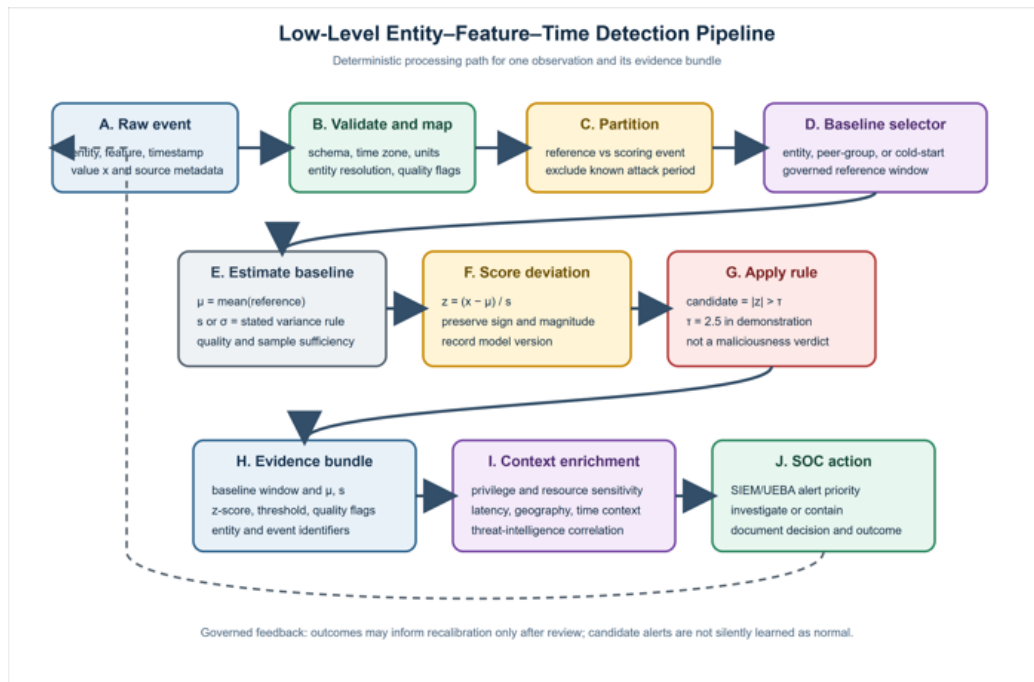


Figure 2: Low-level processing architecture for a single behavioural observation. The data path makes the unit of analysis, baseline selection, variance convention, thresholding, evidence bundle, contextual enrichment, and response hand-off explicit.

8.2 Analytical Discussion

The results should be interpreted at the level supported by the design. The experiment demonstrates that a behavioural observation can be transformed into a transparent baseline deviation and then attached to a structured SOC evidence path. This is a meaningful systems contribution because it links statistical inference to operational accountability. It is not, however, equivalent to proving that the event is malicious, that the detector outperforms alternatives, or that response time has been reduced. Those stronger propositions require independent labels, leakage-controlled evaluation, comparative baselines, and measured operational outcomes.

The resulting claim is deliberately precise: **the proposed architecture is a reproducible and auditable foundation for behavioural anomaly triage in cloud-scale security operations; its production detection accuracy and incident-response impact remain empirical questions for the next validation stage.** This formulation closes the evidentiary gap while preserving the manuscript’s practical and scholarly contribution.

The framework’s value is interpretability under operational constraints. An analyst can inspect the baseline, deviation, threshold, and contextual evidence rather than receiving an opaque score. This may reduce the explanation burden associated with complex models, but interpretability does not guarantee accuracy. Statistical baselines are vulnerable to nonstationary, sparse histories, peer-group mismatch, seasonal behaviour, and contamination by attack activity.

The most consequential design requirement is therefore **governed adaptation**. Cloud-scale systems create high-cardinality entities, ephemeral identities, horizontal scaling, and multi-region movement. A baseline engine must adapt without absorbing malicious drift. The final implementation should specify change detection, freeze conditions, rollback, peer-group selection, and analyst approval for material recalibration.

8.3 Regulatory and Practitioner Implications

A transparent anomaly score can support audit trails, incident investigation, and control monitoring under cybersecurity and resilience regimes. It does not itself demonstrate NIS2 or DORA compliance. Practitioners should use the method as one layer within defence in depth, combined with identity assurance, endpoint protection, network controls, threat intelligence, incident response, and governance documentation.

9. RECOMMENDATIONS, POLICY ADVICE, AND PRACTITIONER FORESIGHT

9.1 Research recommendations

- Release the synthetic dataset or a statistically equivalent generator and the complete analysis code.
- Add independently labelled attack and benign events.
- Compare pooled, peer-group, entity-specific, and dynamically recalibrated baselines.
- Report FPR, FNR, precision, recall, F1, alert rate, calibration, and confidence intervals.
- Evaluate sensitivity to thresholds, baseline windows, missingness, drift, and contaminated reference data.
- Compare the statistical engine with at least one transparent benchmark and one machine-learning comparator.
- Measure analyst utility and operational effects rather than inferring them from anomaly counts.

9.2 Enterprise Recommendations

Enterprises should begin with an inventory of telemetry, entity identifiers, retention periods, and data-quality constraints. Each alert should carry an explanation bundle: feature, baseline, window, deviation, threshold, contextual signals, and

recommended next step. Dynamic recalibration should be governed through change control, not treated as an automatic technical convenience.

9.3 Policy and Regulatory Advice

Regulators and standards bodies should encourage auditable monitoring processes without prescribing a single detection algorithm. Evidence expectations should focus on coverage, traceability, tuning, incident escalation, false-alert management, and continuous improvement. Organisations should avoid presenting statistical alerts as legal findings or automated proof of individual wrongdoing.

9.4 Practitioner Foresight

Future cloud SOC's will combine statistical baselines with graph, temporal, and language-model-assisted investigation. The strategic requirement will be to retain a verifiable evidence chain as automation increases. Statistical methods can remain valuable as a transparent control layer, drift monitor, and explanation reference even when more complex models are added.

10. RESEARCH IMPLICATIONS, ORIGINALITY, AND CONTRIBUTION

10.1 Originality

The originality claim is architectural and operational: the paper connects a transparent statistical baseline engine to explicit cold-start governance, contextual risk enrichment, and SOC response workflows. The Z-score, confidence interval, and UEBA concepts themselves are established and should not be presented as novel inventions.

10.2 Contribution to Knowledge

The paper contributes a structured way to separate four questions that are often conflated: what behaviour is observable; what behaviour is normal for the chosen reference population; how large is the deviation; and what operational context determines its priority. This separation creates a clearer basis for empirical comparison and reproducibility.

10.3 Contribution to Academia, Practice, and Society

For academia, the framework provides a testable pipeline and identifies the evaluation artefacts required to move from statistical demonstration to detection-performance evidence. For practice, it offers an interpretable alert explanation structure that can be integrated with SIEM, UEBA, and SOAR. For society, improved detection of compromised accounts and abnormal access may support digital resilience, but automation must respect proportionality, privacy, access governance, and human oversight.

11. CONCLUSION AND FUTURE WORK

This paper presents an interpretable statistical framework for behavioural anomaly triage in SOC-driven cloud security operations. It specifies how telemetry is prepared, baselines are constructed, deviations are standardised, cold-start conditions are managed, and candidate anomalies are passed into contextual SOC workflows. The worked example demonstrates the arithmetic and exposes an important implementation requirement: the manuscript must state consistently whether the population or sample standard deviation is used.

The current evidence supports a narrower conclusion than the original manuscript claimed. The framework is

computationally transparent and operationally plausible on the described synthetic laboratory data. It does not yet demonstrate production-scale detection accuracy, reduced MTTR or MTTR, or reliable identification of malicious activity because independent labels, comparative metrics, and complete operational calculations are not reported. The next study should use multi-source cloud telemetry, attack and benign ground truth, leakage-controlled temporal evaluation, threshold sensitivity analysis, comparator methods, and analyst-centred operational metrics.

Future work should investigate dynamic and peer-group baselines, temporal and graph models, hybrid statistical-machine-learning detectors, privacy-preserving analytics, adversarial baseline poisoning, cross-cloud identity resolution, and explainable automated response. The paper's durable contribution is therefore a reproducible and auditable foundation upon which those stronger evaluations can be built.

12. ACKNOWLEDGMENTS

We sincerely appreciate the JAAI reviewers for their invaluable feedback, which has significantly enhanced the quality and clarity of our manuscript. Their thoughtful and constructive comments have been instrumental in refining the content to meet both rigorous academic standards and industry relevance. We are grateful for the time and effort they dedicated to reviewing our work and helping us achieve a manuscript that is both academically sound and aligned with professional expectations.

13. REFERENCES

- [1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.
- [2] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.
- [3] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symposium on Security and Privacy*, pp. 305–316, 2010.
- [4] J. Opara-Martins, "Architecting Behavioural Anomaly Detection for Cloud-Scale Security Operations: A Statistical Framework for SOC-Driven Cyber Threat Analytics," EU Digital4Business Consortium, 2026.
- [5] IBM Security, "IBM QRadar SIEM: Security Information and Event Management," 2025. [Online]. Available: <https://www.ibm.com/security/security-intelligence/qradar>. [Accessed: 7 Mar. 2026].
- [6] European Commission, "Network and Information Security Directive 2 (NIS2)," *Official Journal of the European Union*, 2023.
- [7] European Parliament, "Digital Operational Resilience Act (DORA)," *Official Journal of the European Union*, 2022.
- [8] NIST, *Framework for Improving Critical Infrastructure Cybersecurity*, National Institute of Standards and Technology, 2018.
- [9] ISO/IEC, *ISO/IEC 27001:2013 Information technology—Security techniques—Information security management systems—Requirements*, 2013.
- [10] A. Oprea, Z. Li, A. Juels, M. K. Reiter, and M. Robinson, "Detection of early-stage insider attacks using behavioral

models,” in *Proc. ACM SIGSAC Conference on Computer and Communications Security*, pp. 109–120, 2015.

- [11] M. Ahmed, A. N. Mahmood, and J. Hu, “A survey of network anomaly detection techniques,” *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.
- [12] G. Kim, S. Lee, and S. Kim, “A novel hybrid intrusion detection method integrating anomaly detection with misuse detection,” *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2014.
- [13] Y. Zhang, N. Meratnia, and P. Havinga, “Outlier detection techniques for wireless sensor networks: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.
- [14] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *Proc. IEEE International Conference on Data Mining*, pp. 413–422, 2008..

- [15] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, “Deep learning on graphs: A survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 249–270, 2019.

- [16] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier,” in *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

14. APPENDICES

Appendix A. Minimum data dictionary

Include field name, type, unit, source, aggregation window, entity key, missingness rule, retention rule, and privacy classification for every variable. The final data dictionary must state the unit, aggregation rule, entity key, and missing-data treatment for every feature; the current manuscript reports the feature concepts but not all schema-level definitions.

Feature	Operational role	Unit required for final release	Entity/time key	Supplied status
Authentication timestamp	Temporal and frequency behaviour	ISO-8601 timestamp or equivalent	Entity × event	Reported conceptually; exact schema NR
Login frequency	Authentication-volume deviation	Count per stated interval	Entity × interval	Reported; exact interval NR
Session duration	Persistence and session-behaviour deviation	Seconds/minutes per session	Entity × session	Reported conceptually; exact unit NR
File-access frequency	Access-volume deviation	Count per stated interval	Entity × interval	Reported conceptually; exact interval NR
Network-access event	Network-behaviour context	Event count/type	Entity × interval	Reported conceptually; schema NR
Access latency	Resource-access timing deviation	Milliseconds/seconds	Entity × event or interval	Reported conceptually; unit NR
Privilege escalation	Contextual risk signal	Binary event/count and role transition	Entity × event	Reported conceptually; coding NR
Failed authentication attempts	Credential-misuse signal	Count/rate per interval	Entity × interval	Reported conceptually; coding NR
Entity identifier	Baseline grouping key	Pseudonymous identifier	stable	User/service/workload/device Exact key NR
Label	Independent evaluation outcome	Attack/benign/uncertain	Entity × event	Not supplied

Appendix B. Detection pseudocode

for each entity e and feature f:

```

reference = observations(e, f, reference_window)
if quality(reference) < minimum_requirement:
    baseline = governed_peer_or_heuristic_baseline(e,
f)

```

```
status = "cold-start"
```

```
else:
```

```
    baseline = estimate_mean_and_sd(reference)
```

```
    status = "statistical"
```

```
for each new observation x(e, f, t):
```

```

z = (x - baseline.mean) / baseline.sd
candidate = abs(z) > threshold
context = enrich(x, privilege, latency, resource,
threat_intelligence)
emit_alert(e, f, t, z, candidate, context, status)

update_baseline_only_under_governed_drift_policy()

```

Appendix C. Required Evaluation Report

Report dataset counts; baseline windows; threshold choices; alert rates; confusion matrix; FPR; FNR; precision; recall; F1; PR-AUC or ROC-AUC where appropriate; confidence intervals; calibration; threshold sensitivity; drift sensitivity; computational latency; and analyst workload. Because the supplied record does not contain the complete preprocessing audit, the present study is reported as a computational demonstration rather than a leakage-controlled accuracy benchmark.

Processing control	Required camera-ready specification	Current supplied status	Camera-ready treatment
Timestamp normalisation	Time zone, daylight-saving rule, timestamp precision	NR	Insert exact rule or retain NR
Schema validation	Required fields and invalid-record criteria	NR	Insert validation rule
Entity resolution	Mapping from raw principal to monitored entity	NR	Insert keying procedure
Duplicate handling	Duplicate definition and removal rule	NR	Insert rule
Missing values	Field-level imputation, exclusion, or missingness flag	NR	Insert exact treatment
Malformed records	Rejection criteria and count rejected	NR	Insert audit count
Feature aggregation	Interval and aggregation function	NR	Insert exact interval and function
Reference baseline	Window preceding scored observation	Conceptual specified	rule Retain and implement
Self-inclusion control	Scored observation excluded from its own baseline	Conceptual specified	rule Retain
Attack contamination control	Known attack periods excluded or sensitivity-tested	Labels not supplied	Mark unavailable unless evidence exists
Reference/scoring split	Separate baseline and scoring partition	NR	Insert split or state demonstration only
Evaluation partition	Independent labelled/adjudicated outcomes	Not supplied	N/A for current unlabelled demonstration
Recalibration policy	Drift trigger, freeze rule, rollback, approval	Proposed conceptually	Insert operational thresholds when available