TYPE Original Research
PUBLISHED 17 November 2026
DOI 10.3389/bjbs.2026.17568

Check for updates

OPEN ACCESS

*CORRESPONDENCE
Muhammad Haseeb Tariq,
muhammadhaseebtariq19@gmail.com

RECEIVED 08 August 2026
REVISED 30 October 2026
ACCEPTED 10 November 2026
PUBLISHED 17 November 2026

CITATION
Tariq MH, Price C, Laher A, Harrison M, Ammoun M, Amir A and Tariq MM (2026) Simulation-based comparison of unsupervised anomaly detection and conventional multi-rules for clinical biochemistry quality control.
Br. J. Biomed. Sci. 83:17568.
doi: 10.3389/bjbs.2026.17568

COPYRIGHT
© 2026 Tariq, Price, Laher, Harrison, Ammoun, Amir and Tariq. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Simulation-based comparison of unsupervised anomaly detection and conventional multi-rules for clinical biochemistry quality control

Muhammad Haseeb Tariq [1]*, Claire Price [2], Ahmed Laher [1], Melissa Harrison [1], Mariam Ammoun [3], Afreenish Amir [4] and Muhammad Muneeb Tariq [5]

[1]Department of Biochemistry, Mid Yorkshire Hospitals NHS Trust, Wakefield, United Kingdom, [2]Institute of Life Science, Swansea University Medical School, Swansea, United Kingdom, [3]Department of Blood Transfusion, Royal Marsden NHS Foundation Trust, London, United Kingdom, [4]Department of Microbiology, National Institutes of Health, Islamabad, Pakistan, [5]Department of Computer Sciences, Quaid-i-Azam University, Islamabad, Pakistan

Background: Internal quality control is commonly carried out using rule-based systems such as Westgard multirules and the exponentially weighted moving average. However, these methods may be slow to detect gradual drift and imprecision changes in current analyzers. Unsupervised anomaly detection algorithms, trained on in-control quality control data, provide an additional data-driven approach that does not require patient data.
Methods: We generate synthetic IQC data with four faults: a +2 SD step shift, a linear trend to +2 SD, an abrupt doubling of the SD, and intermittent ±4 SD outliers (3% of post-onset runs) on a standardized z-scale, including 54,000 QC-level series (12 analyte labels × 3 QC levels × 5 scenarios × 300 replications). We used Holm-corrected paired tests and bootstrap confidence intervals to summarise delay per replication (median across 12 analytes); outlier delay was calculated from the first injected outlier, and delay in fault-free streams served as the no-fault reference. A severity sweep changed the fault's trajectory and size.
Results: Step shifts had median delays of 0.0–2.0 runs, while sudden imprecision increases had median delays of 0.5–3.5 runs, with CUSUM being the slowest. Delays from the initial injected outlier were 1.0 for EWMA, 12.5 for CUSUM, and 0.0 runs for Westgard, Isolation Forest, and One-Class SVM. The median delays for the trend were 19.8 (EWMA), 23.5 (Isolation Forest), 30.0 (Westgard), 31.5 (One-Class SVM), and 44.0 runs (CUSUM). Except for CUSUM, the trend delays were only roughly 2–7 runs shorter than the time to first alert in fault-free streams, and the false-alarm rates (2.2–5.2 per 100 analyte-runs) did not match. Most of the machine-learning discrimination was explained by engineered temporal features (imprecision AUC for Isolation Forest 0.93 with all features versus 0.70 with z and |z| only).
Conclusion: No method was consistently the fastest; instead, performance varied by defect type and false-alarm operating point. Discrepancies in the false-alarm rate were a major factor in explaining apparent discrepancies

between techniques for slowly developing trends. Validation using actual laboratory QC data is necessary because these standardized simulations do not show clinical benefit.

KEYWORDS
CUSUM, exponentially weighted moving average (EWMA), internal quality control, isolation forest, one-class SVM

Introduction

Internal quality control (IQC) is a critical component of laboratory quality management, ensuring that analytical systems perform within acceptable limits before patient data is released [1]. Westgard multirules implement Shewhart charting by combining multiple criteria (1–3s, 2-2s, R-4s, 4-1s, and 10x) to detect random and systematic errors using standardized, interpretable logic [2–5]. Because of their simplicity and interpretability, these methods are still frequently employed in internal quality control operations in clinical biochemistry laboratories. However, fixed rule frameworks were created based on analytical assumptions that are not necessarily appropriate in modern high-throughput biochemistry laboratories [2].

Analytical deterioration is most common in quality control time series generated by modern analyzers. Intermittent outliers (such as transient aspiration incidents), Gradual drift, including reagent aging and calibration bias accumulation, and subtle fluctuations in imprecision, mainly caused by instrument wear or environmental instability. Westgard rules are particularly effective in detecting sudden step shifts. These patterns can stay inside normal control parameters for a long time, delaying remedial action and increasing the chance of false patient test findings [6].

Exponentially weighted moving average (EWMA) charts are often suggested for tracking drift since they gather evidence of minor, persistent variations over time [7, 8]. Cumulative sum (CUSUM) charts, as described in the CLSI C24-Ed4 IQC guideline [6, 9], use a similar concept of cumulative evidence accumulation. They are theoretically ideally suited for drift detection, but their use in UK clinical laboratories is restricted. However, neither EWMA nor CUSUM capture complicated multivariate temporal structures, such as run-to-run variability patterns, rolling variance changes, and paired deviation signatures, which may precede overt analytical failure.

Patient-based real-time quality control systems have complementary detection capabilities, but require continuous access to patient result streams, introducing privacy, permission, and legal issues that have limited their widespread deployment in most healthcare sectors and in NHS practice [10, 11]. Unsupervised anomaly detection, by contrast, is an alternative paradigm in which models learn the characteristics of typical QC behavior from in-control QC material and then evaluate new data based on deviations from this established baseline. This method avoids the necessity for patient data, accepts feature-rich multivariate inputs, and adjusts to complex nonlinear patterns in QC time series. The Isolation Forest technique identifies anomalies through random partitioning in feature space [12, 13] and the One-Class SVM sets the boundary of normal operation within a kernel-transformed high-dimensional space [14].

Currently, many clinical biochemistry laboratories use a single software for monitoring, mainly Westgard Multirules enhanced by EWMA & CUSUM. However, we observe that unsupervised anomaly detection remains largely limited to small pilot studies and is not used at scale.

This patchwork implementation reflects a real evidence gap: laboratories now have a choice of several competing IQC paradigms (Westgard, EWMA, CUSUM, and machine learning-based anomaly detection), but no study has systematically benchmarked their relative performance under a common set of realistic, modern analytical failure modes. As a result, it is unclear which method, or combination of methods, laboratories should prioritize when developing IQC strategies that detect drift, imprecision, outliers, and rapid shifts. This study directly addresses this issue by employing a large-scale, multi-analyte simulation benchmark; to our knowledge, it is one of the first studies to comprehensively assess the detection performance of five commonly accessible IQC methods across a shared set of genuine defect types. This study provides a 300-replication simulation benchmark comparing Isolation Forest and One-Class SVM with Westgard multirules, CUSUM, and EWMA.

We assess detection delay using replication-level uncertainty estimates, false-alarm behaviour at multiple operating points, the contribution of engineered features, an exploratory combination of Westgard and Isolation Forest alerts, and a no-fault reference that demonstrates the portion of a delay attributable to chance alerts. We hypothesized that the type of failure and the false-alarm operating point would determine relative performance.

This research provides three primary contributions. First, it compares machine-learning anomaly detection with multirule and statistical process-control methods for internal quality control across four simulated fault types, utilizing replication as the unit of analysis. Second, it reports detection delay relative to the realized false-alarm rate, rather than at a single nominal calibration target, enabling comparison of methods at equivalent alarm burdens. Third, it quantifies the extent to which apparent detection delay for slowly developing faults results from false-alarm timing rather than actual fault detection. We do not assess clinical outcomes or workload; the findings are intended to guide prospective evaluation rather than current laboratory practice.

Materials and methods

This study uses only synthetic QC values and provides a controlled, pre-deployment simulation benchmark. The study was designed to inform subsequent local verification rather than to establish compliance with ISO 15189:2022. This strategy enabled the systematic comparison of internal quality control strategies under defined analytical fault patterns before clinical implementation [15].

This simulation design aligns with established internal quality control practices, where Monte Carlo-generated quality control series are used instead of patient-derived data to benchmark statistical control procedures against Westgard multirules [16, 17].

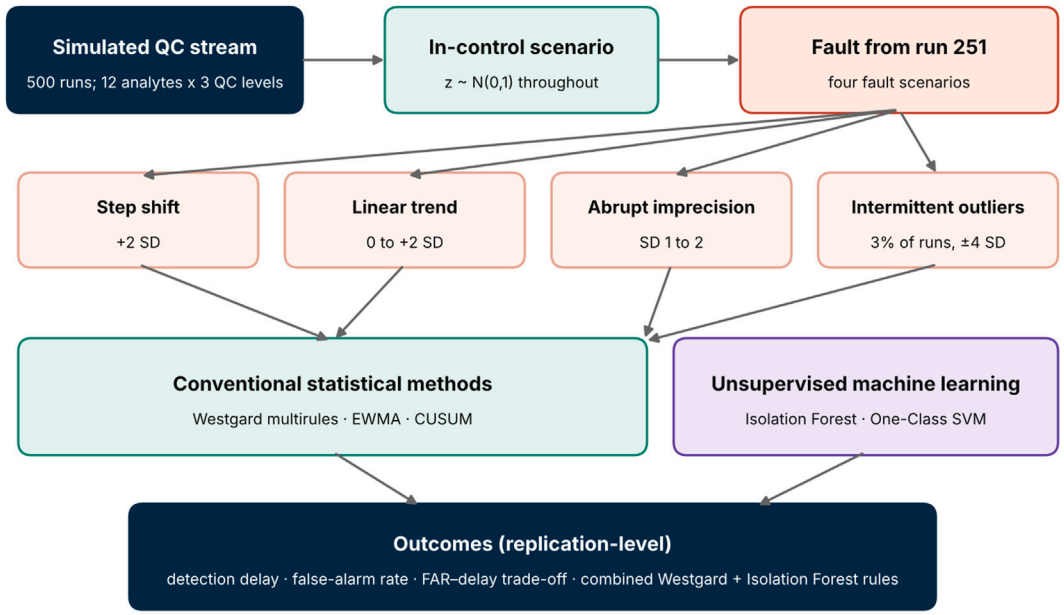


FIGURE 1
Study design. Simulated quality control (QC) streams (500 runs; 12 analytes × 3 QC levels) remain in control until run 250, after which one of four faults is introduced from run 251. We apply Westgard multirules, exponentially weighted moving average (EWMA), and cumulative sum (CUSUM) methods (conventional approaches), as well as Isolation Forest and One-Class Support Vector Machine (SVM) (unsupervised machine learning methods), to each series. Replication-level outcomes include detection delay, false-alarm rate, the false-alarm-delay trade-off, and exploratory combinations of Westgard and Isolation Forest. Clinical outcomes are not modeled.

Twelve common biochemistry analytes were modelled: ALT, glucose, creatinine, urea, sodium, calcium, albumin, total cholesterol, total protein, phosphate, potassium and bilirubin. For each analyte, we simulated three QC concentration levels (L1–L3) to represent low, normal, and high clinical concentrations. The target means (μ) and standard deviations (σ) for each analyte and QC level, given in Supplementary Table S9, are illustrative simulation parameters. They were set with reference to typical concentrations and analytical coefficients of variation in the EFLM biological variation database (https://biologicalvariation.eu/) [18] and the Ricós/EFLM literature [19], and they were not derived analyte by analyte from biological variation estimates, which describe within and between subject variation rather than analytical quality-control performance. Each evaluation series comprised 500 consecutive quality control (QC) runs, with fault onset introduced at run 251, resulting in 250 pre-onset observations (runs 1–250) and 250 post-onset observations (runs 251–500). We generated separate in-control training streams, independent of the evaluation streams, for model fitting and threshold calibration. We applied detection methods to standardized z-scores and simulated all series directly on the z-scale. Supplementary Table S9 defines concentration units only; the 12 analytes and 3 QC levels represent statistically exchangeable replicates of the same process rather than 12 independent performance domains.

Features were computed from run 11 onward, with runs 11 to 250 representing the pre-onset period. We evaluated all five methods over identical windows: Westgard, EWMA, and CUSUM statistics were accumulated from run 1, and we counted alerts for all methods from run 11 for false alarms (runs 11–250) and from run 251 for delay. We conducted 300 independent replications. In each replication, 36 quality control (QC)-level series (12 analytes by 3 QC levels) were generated for each of five scenarios, resulting in 54,000 simulated QC-level series (36 × 5 × 300). Delay and detection outcomes were defined for the 12 analyte streams per scenario, with alerts aggregated over the three QC levels. No formal sample-size calculation was required for the simulation; we used 300 replications, and the bootstrap confidence intervals indicate the resulting precision. These data represent simulated series rather than independent laboratories or patients, and the 36 series within each replication share the trained models. Figure 1 summarizes the experimental design, and Supplementary Figure S1 provides an example series with the alerts generated by each method. In all five scenarios, we drew pre-onset observations from a standard normal distribution, N(0,1).

a. In-control, where z_t follows N (0,1) for all 500 runs,
b. Step shift where z_t follows N (+2,1) starting at run 251, representing a standardized step increase of +2SD [18, 19].
c. Linear trend, where z_t follows N(δ_t,1) and δ_t increases linearly from 0 at run 251 to +2 at run 500
d. Abrupt imprecision increase, where z_t follows N(0,2^2) from run 251 onward an unchanged mean but a doubled standard deviation i.e., an abrupt doubling of the standard deviation, (not a progressive increase).
e. Outliers, where z_t follows N(0,1) throughout, except that 3% of post-onset runs (selected independently for each series) are replaced by an excursion of +4 or −4 SD (equal probability, small random jitter, SD 0.15). The 3% rate

and ±4 SD size are modeling assumptions for gross errors (cf. 20), not estimates from a specific analyzer or from that [20].

Raw QC values were converted to analyte-specific z-scores: $z = (x - \mu)/\sigma$. A six-element feature vector characterized each QC run:

a. z (standardized z-score);
b. |z| (absolute z-score);
c. Δz ($z_t - z_{t-1}$, run-to-run change);
d. roll_mean (10-run rolling mean of z);
e. roll_SD (10-run rolling SD of z);
f. ewma (EWMA of z, λ = 0.2).

Rolling features were computed from run 11 onward; observations before run 11 were excluded from all analyses to ensure feature stability.

Westgard multirules were applied to each analyte at each run as follows: 1–3s (|z| > 3 at any QC level); 2-2s (two consecutive results of the same QC level beyond the same 2 SD limit); R-4s (within one run, one QC level above +2 SD and another below −2 SD); 4-1s (four consecutive results of the same level beyond the same 1 SD limit); and 10x (ten consecutive results of the same level on the same side of the mean). The 1–3s and R-4s rules utilize the three QC levels of the same run jointly, while 2-2s, 4-1s, and 10x are evaluated within each QC level across consecutive runs. A single rule violation generated an alert for the corresponding analyte-run. Each rule used the standardized values of the three QC levels directly, without pooling the levels into a single ordered sequence. We verified rules using hand-constructed positive and negative test cases. For example, two levels of the same run at +2.1 and −2.2 SD (or +2.1 and −2.1 SD) constitute an R-4s violation. In contrast, two levels of the same run at +2.9 and −1.2 SD, although their range exceeds 4 SD, are not on opposite sides beyond ±2 SD and therefore do not represent a violation. Similarly, a single level at +2.5 SD followed by −2.5 SD in the next run represents a between-run excursion that is not an R-4s violation (Supplementary Table S6). The initial submission described R-4s as a range between consecutive runs, whereas the submitted code flagged any run in which the range across the three QC levels exceeded 4 SD without requiring opposite sides beyond ±2 SD. Neither approach aligns with the conventional rule. We corrected the implementation and provided a sensitivity analysis evaluating all rules jointly across levels in Supplementary Table S3.

EWMA statistic: $E_t = \lambda z_t + (1-\lambda)E_{t-1}$, with $E_0 = 0$ and $\lambda = 0.2$. Control limits calibrated per analyte-level series at the 99th percentile of $|E_t|$ from the pre-fault in-control segment (runs 11–250), targeting a nominal false alarm rate of ~1 per 100 in-control runs [21, 22].

Tabular CUSUM statistics: $C_t^+ = max(0, z_t - k + C_{t-1}^+)$ and $C_t^- = max(0, -z_t - k + C_{t-1}^-)$, applied to standardized z-scores (process SD = 1) where k = 0.5σ per CLSI C24-Ed4 recommendations [3, 9]. Alert generated when C^+_t or C^-_t exceeded the decision interval h = 5σ; settings fixed in advance and not tuned to these scenarios. The statistics were not reset after an alert.

Isolation Forest (100 trees) and One-Class SVM (RBF kernel, ν = 0.01, γ = 'scale') models were fitted to standardized features derived from in-control training streams (runs 11–250), pooled across the 36 series within each replication. Model fitting utilized a random sample of the pooled rows, selecting either 70% of the rows or 1,500 rows, whichever was smaller. We fitted the scaler exclusively on these training rows. The remaining in-control rows were reserved for decision-threshold calibration, where the threshold corresponded to the anomaly-score quantile exceeded by a fraction q of calibration observations (q = 0.5%, 1%, 2%, and 5% per QC-level observation). We pooled the 36 standardized series for model fitting because these series share the same in-control distribution by design, resulting in a larger, more stable training set and enabling one model per instrument rather than per analyte level. We did not evaluate per-series models, and a pooled model does not capture analyte-specific behavior. Table 1 reports results for q = 1%. We retrained the scaler, fitted on the training data, for all evaluation streams, and we retrained both models independently in each of the 300 replications. An analyte-run triggered an alert if any of its three QC levels exceeded the threshold. The data split was performed over individual feature rows (via random permutation of the pooled rows), rather than over series or time blocks. Of the 8,640 pooled rows (36 series × 240 rows), the 70% selection exceeded the 1,500-row cap, so we used 1,500 rows for fitting and the remaining 7,140 for threshold calibration. Because rolling-mean, rolling-SD, and EWMA features of adjacent runs share observations, fitting and calibration rows are not independent in the time-series sense, whereas the evaluation streams remain independent of both. The AUC analysis refitted models for each feature configuration using the same split procedure in each replication, and the permutation analysis employed the main Isolation Forest of each replication. Reporting of the machine-learning components in this study followed the TRIPOD-AI recommendations for studies that develop or evaluate prediction models incorporating artificial intelligence. The present analysis is exploratory and simulation-based and does not constitute a full TRIPOD-AI-compliant validation study, which would require external, real-world laboratory data, as described under Limitations [23].

Detection delay was defined as the number of runs from fault onset (run 251) to the first alert; thus, an alert in run 251 corresponded to a delay of 0. In the outlier scenario, we measured delay from the first injected outlier at any of the analyte's three quality control (QC) levels. We excluded analytes without an injected outlier, although none did. Series without an alert during the remaining observation window were treated as right censored at the window length. We calculated the false-alarm rate as the number of runs with an alert per 100 analyte-runs in runs 11–250 of the in-control scenario streams, which were independent of the training and calibration streams. This metric count runs in an alert state, so persistent exponentially weighted moving average (EWMA) or cumulative sum (CUSUM) states are counted in every run. The false-alarm rate therefore measures the proportion of runs spent in an alert state, rather than the number of distinct false-alarm events, and we use this definition throughout. Because the EWMA and CUSUM statistics were not reset, any method already in an alert state at run 251 was assigned a delay of 0 in both fault scenarios and the no-fault reference. An alert already active at fault onset was therefore counted as a detection with zero delay to match the predefined metric; however, such an alert does not necessarily indicate that the newly introduced fault was detected. The

TABLE 1 Replication-level detection delay by method and scenario.

Scenario	Method	Median delay, runs (IQR)	95% CI of median	Alert within 50 runs, % (95% CI)
Step shift	Westgard	1.0 (0.0–1.0)	1.0–1.0	100.0% (100.0–100.0)
Step shift	EWMA	1.0 (1.0–1.0)	1.0–1.0	100.0% (100.0–100.0)
Step shift	CUSUM	2.0 (1.0–2.0)	1.5–2.0	100.0% (100.0–100.0)
Step shift	Isolation forest	0.0 (0.0–0.5)	0.0–0.0	100.0% (100.0–100.0)
Step shift	One-class SVM	1.0 (0.5–1.0)	1.0–1.0	100.0% (100.0–100.0)
Linear trend	Westgard	30.0 (25.0–37.5)	29.0–31.5	74.2% (72.9–75.7)
Linear trend	EWMA	19.8 (16.0–25.1)	18.8–21.0	91.1% (90.2–92.0)
Linear trend	CUSUM	44.0 (40.0–48.5)	43.5–45.5	59.7% (58.3–61.1)
Linear trend	Isolation forest	23.5 (17.5–29.5)	22.0–25.2	85.4% (84.1–86.6)
Linear trend	One-class SVM	31.5 (24.5–38.5)	30.2–32.8	75.2% (73.7–76.8)
Abrupt imprecision increase	Westgard	1.0 (0.5–1.0)	1.0–1.0	100.0% (100.0–100.0)
Abrupt imprecision increase	EWMA	1.5 (1.0–2.0)	1.5–1.5	100.0% (100.0–100.0)
Abrupt imprecision increase	CUSUM	3.5 (2.5–4.0)	3.0–3.5	100.0% (100.0–100.0)
Abrupt imprecision increase	Isolation forest	0.5 (0.0–1.0)	0.5–1.0	100.0% (100.0–100.0)
Abrupt imprecision increase	One-class SVM	1.0 (1.0–1.5)	1.0–1.0	100.0% (100.0–100.0)
Intermittent outliers	Westgard	0.0 (0.0–0.0)	0.0–0.0	100.0% (100.0–100.0)
Intermittent outliers	EWMA	1.0 (0.0–2.5)	0.5–1.0	98.1% (97.6–98.5)
Intermittent outliers	CUSUM	12.5 (8.0–17.5)	11.5–13.2	84.9% (83.8–85.9)
Intermittent outliers	Isolation forest	0.0 (0.0–0.0)	0.0–0.0	100.0% (100.0–100.0)
Intermittent outliers	One-class SVM	0.0 (0.0–0.0)	0.0–0.0	100.0% (100.0–100.0)

Reported metrics include the median detection delay (measured in runs from fault onset at run 251, or from the first injected outlier for outlier scenarios), interquartile range (IQR), bootstrap 95% confidence interval (CI) of the median across 300 replications, and the percentage of analyte streams (300 replications × 12 analytes) with an alert within 50 runs (95% CI, from resampling replications).

percentage alerting within 50 runs represents the alerting proportion in simulated data, not a sensitivity estimates against real fault labels, and the first alert is not necessarily attributable to the fault. We obtained a no-fault reference delay by applying the delay definition to the in-control streams from run 251, representing the time to the first false alarm.

Each replication contributed one summary per method and scenario, defined as the median delay across the 12 analytes. We reported the median, interquartile range (IQR), and a bootstrap 95% confidence interval (10,000 resamples of replications) for the 300 replication-level values. The proportion of analyte streams with an alert within 50 runs was calculated as the mean over all 3,600 streams for each scenario and method, with a bootstrap interval that resampled entire replications to account for clustering of the 12 streams within a replication. We compared the methods using a Friedman omnibus test and two-sided paired Wilcoxon signed-rank tests across replications, with Holm correction applied within each scenario. Because medians over 12 analytes take values in steps of 0.5 runs, very small p-values indicate consistency in ordering across replications rather than the magnitude of a difference. We defined excess detection for a method in the trend scenario as its proportion of alerts within 50 runs minus the same proportion in the in-control scenario, paired by replication. Westgard OR Isolation Forest and Westgard AND Isolation Forest were evaluated as exploratory analyte-run-level combinations.

Severity sweep. To assess the impact of fault magnitude and trajectory, we simulated additional streams in each replication, independent of the primary evaluation streams, using the same trained models and 1% thresholds. These included step shifts of +1, +1.5, +2, and +3 standard deviations (SD); linear trends ending at +1, +1.5, +2, and +3 SD at run 500; abrupt increases in SD by factors of 1.5, 2, and 3; a progressive imprecision scenario in which SD increased linearly from 1 at run 251 to 2 or 3 at run 500; and outlier magnitudes of ±3, ±4, and ±5 SD (3% of runs from run 251, with delay measured from the first injected outlier). The +2 SD step, +2 SD trend, ×2 abrupt, and ±4 SD outlier scenarios repeated the primary scenarios with new random streams. We summarized delay and the proportion alerting within 50 runs as described above.

Three feature configurations were assessed for both Isolation Forest and One-Class SVM to separate algorithmic contribution from feature-engineering contribution: (1) all six features; (2) raw features only (z and |z|); and (3) no EWMA feature. For each replication, we calculated the area under the ROC curve. (AUC), which is independent of any threshold, separating post-onset runs (runs 251–500) of each fault scenario from runs 251–500 of the in-

control scenario (for the outlier scenario, only runs containing an injected outlier). We assessed the descriptive sensitivity of the Isolation Forest score to each feature by permuting that feature in 128 held-out in-control rows per replication; this is a model-sensitivity measure, not an attribution of alerts or a root-cause analysis.

We provide an open-access, browser-based companion application with this study to support reproducibility of the simulation analyses and offer an educational environment for exploring various internal quality-control (IQC) methods. This application also provides access to the benchmark tables, figures, and detection-delay calculations presented in this manuscript. The application has four functional sections. First, incoming QC data are shown on a Levey-Jennings chart with overlays for Westgard, EWMA, CUSUM (C^+/C^-), Isolation Forest, and One-Class SVM using a live IQC monitoring suite (Dashboard, QC Data Entry, and QC Chart & Results). This creates a Next QC, Resolution & Audit Log, and an Active Alert Log. Based on the laboratory's historical inter-run gap (median cadence), the due panel estimates the anticipated date of the next test for each analyte/level and marks runs as past due or soon due. This approach estimates due dates based on the local testing cadence rather than a fixed schedule, which may help identify overdue QC measurements or shifts in testing patterns.

The dashboard flags run comments, computes the mean z-score shift for the impacted analyte/level before and after the change, and marks those runs on the Levey-Jennings chart.

The application is available at https://muhammadhaseeb823.github.io/Clinqc-Pro/.

Results

We evaluated fault-detection speed across four scenarios using 300 replications, treating each replication as an independent case. Table 1 presents the median detection delays, interquartile ranges (IQRs), bootstrap 95% confidence intervals, and the percentage of series that alerted within 50 runs. A delay of 0 indicates that an alert was present at the first evaluated run following fault onset, whether newly triggered in that run or already active at onset. In the outlier scenario, delay was measured from the run in which the first outlier was introduced. An alert already active at onset does not necessarily indicate detection of the newly introduced fault.

Step shift. All five methods detected the +2 SD shift in 100% of series within 50 runs, with median delays ranging from 0.0 to 2.0 runs (Table 1). Isolation Forest demonstrated the shortest delay (0.0, IQR 0.0–0.5), while CUSUM exhibited the longest (2.0, IQR 1.0–2.0). Westgard, EWMA, and One-Class SVM each had a median delay of 1.0 run. Statistically significant differences were observed (Friedman $p < 0.001$), although the maximum difference was 2 runs.

Abrupt imprecision increase. Median delays were 0.5 runs (IQR 0.0–1.0) for Isolation Forest, 1.0 (IQR 0.5–1.0) for Westgard, 1.0 (IQR 1.0–1.5) for One-Class SVM, 1.5 (IQR 1.0–2.0) for EWMA, and 3.5 (IQR 2.5–4.0) for CUSUM. All methods generated alerts within 50 runs in every replication (Friedman $p < 0.001$). The maximum difference between the fastest and slowest methods was 3 runs.

Outliers. When measured from the first injected outlier, Westgard, Isolation Forest, and One-Class SVM all generated alerts in the same run (median 0.0 runs, IQR 0.0–0.0; 100% within 50 runs), with no differences observed among them (all paired differences were zero). EWMA exhibited a median delay of 1.0 run (IQR 0.0–2.5; 98.1% within 50 runs), while CUSUM had a median delay of 12.5 runs (IQR 8.0–17.5; 84.9% within 50 runs and 99.97% by the end of the window). The immediate response of the first three methods is expected, as a ±4 SD excursion exceeds the 1–3s limit by design.

For the linear trend scenario, the median detection delays were 19.8 (IQR 16.0–25.1) for EWMA, 23.5 (IQR 17.5–29.5) for Isolation Forest, 30.0 (IQR 25.0–37.5) for Westgard, 31.5 (IQR 24.5–38.5) for One-Class SVM, and 44.0 (IQR 40.0–48.5) for CUSUM. The proportions of series alerting within 50 runs were 91.1% for EWMA, 85.4% for Isolation Forest, 74.2% for Westgard, 75.2% for One-Class SVM, and 59.7% for CUSUM. The Friedman test indicated a significant overall difference ($p < 0.001$), and all pairwise comparisons were significant after Holm correction ($p < 0.001$), except for Westgard versus One-Class SVM ($p = 0.092$; Supplementary Table S1). Interpret these rankings alongside the false-alarm rates presented below.

Figure 2A presents time-to-detection curves, showing the proportion of analyte streams without an alert as a function of runs since onset (300 replications × 12 analytes, totaling 3,600 streams per method, with alerts aggregated over three QC levels; Supplementary Table S7). After 50 runs from the onset of the trend, 8.9% of EWMA, 14.6% of Isolation Forest, 24.8% of One-Class SVM, 25.8% of Westgard, and 40.3% of CUSUM series had not alerted. By run 100, all curves were below 0.5%. Under the applied censoring convention, no series in the trend scenario remained undetected at the end of the observation window. In contrast, the corresponding curves for fault-free streams (Figure 2B) demonstrate rapid accumulation of false alarms: after 50 runs, 28% of EWMA, 27% of Isolation Forest, 39% of Westgard, 42% of One-Class SVM, and 70% of CUSUM series had not yet raised an alert, and 18% of CUSUM series remained without an alert after 250 runs.

False-alarm rates. In fault-free streams, the number of false alarms per 100 analyte-runs (median; 2.5th–97.5th percentile across replications) was 2.26 (1.56–3.04) for Westgard, 2.15 (0.88–3.80) for CUSUM, 2.99 (2.07–4.10) for Isolation Forest, 3.12 (1.91–5.21) for One-Class SVM, and 5.17 (3.44–7.24) for EWMA (Table 2). Since each analyte-run aggregates three QC levels, a 1% per-observation target equates to approximately 3 alerts per 100 analyte-runs, rather than the 1 per 100 targeted in the initial submission. EWMA exhibited a higher false-alarm rate than the other methods. Consequently, the methods were not matched on false-alarm rate. The mean contributions of individual Westgard rules across replications (per 100 analyte-runs; the individual rates sum to 2.38, exceeding the mean of 2.29 for any rule, as a run can violate multiple rules simultaneously; the median for any rule is 2.26 as noted above) were 0.80 for 1–3s, 0.32 for 2-2s, 0.29 for R-4s, 0.38 for 4-1s, and 0.59 for 10x (2.29 for any rule; Supplementary Table S2).

No-fault reference. To distinguish fault detection from random alerts, we applied the delay definition to fault-free streams beginning at run 251 (Table 2). For step shift, imprecision, and outliers, all observed delays were substantially lower than this reference. In the case of trend faults, the delays for Westgard (30.0 versus 36.0 runs),

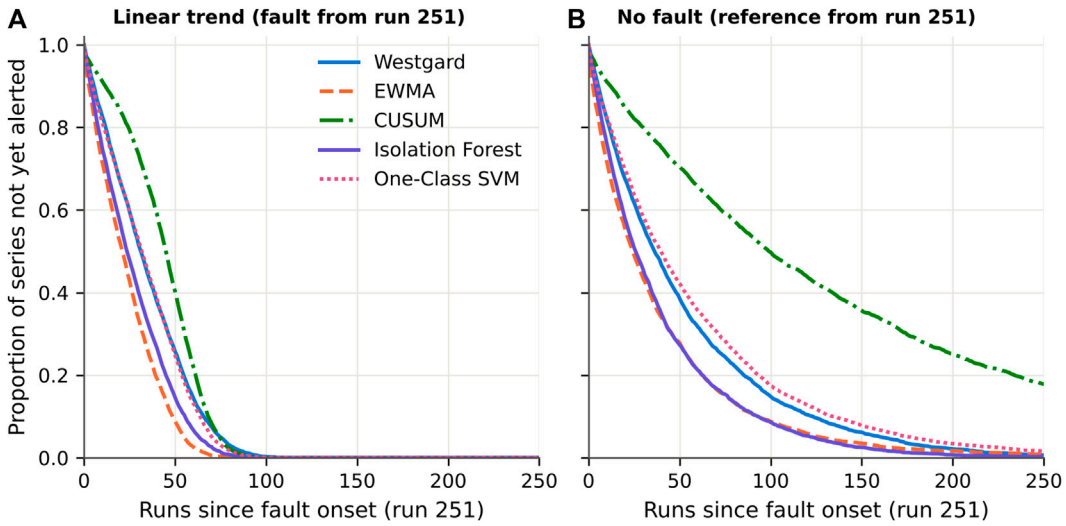


FIGURE 2
Time-to-detection curves. The proportion of analyte streams (300 replications × 12 analytes = 3,600 streams per method, each aggregating alerts across three QC levels) without an alert is plotted as a function of runs since fault onset. **(A)** Linear trend (fault from run 251). **(B)** No fault (reference from run 251), showing the time to the first false alarm. Streams without an alert by run 250 are right-censored at the end of the observation window.

TABLE 2 **False-alarm rates, no-fault reference, and excess detection in the trend scenario.**

Method	False alarms per 100 analyte-runs (2.5th–97.5th pct)	No-fault reference delay, runs (IQR)	No-fault: alert within 50 runs, %	Trend: alert within 50 runs, %	Excess, percentage points (95% CI)
Westgard	2.26 (1.56–3.04)	36.0 (28.0–46.5)	61.4	74.2	12.8 (10.8–14.9)
EWMA	5.17 (3.44–7.24)	24.0 (17.5–32.0)	72.1	91.1	19.0 (17.3–20.8)
CUSUM	2.15 (0.88–3.80)	98.0 (76.0–131.1)	29.6	59.7	30.1 (28.1–32.1)
Isolation forest	2.99 (2.07–4.10)	25.5 (19.0–33.5)	72.7	85.4	12.7 (10.8–14.6)
One-class SVM	3.12 (1.91–5.21)	38.8 (29.0–52.6)	57.8	75.2	17.4 (15.3–19.5)

False alarms are reported per 100 analyte-runs in the in-control scenario, presented as the median and 2.5th–97.5th percentile across replications. The no-fault reference includes the median delay (IQR) to the first alert from run 251 in fault-free streams and the percentage of alerts occurring within 50 runs. Excess detection is the percentage of alerts within 50 runs under the trend scenario minus the corresponding percentage without a fault, expressed in percentage points, with 95% bootstrap confidence intervals across replications.

EWMA (19.8 versus 24.0), Isolation Forest (23.5 versus 25.5), and One-Class SVM (31.5 versus 38.8) were only 2–7 runs shorter than the time to the first alert in the absence of any fault. In contrast, CUSUM alerted after 44.0 runs under trend conditions but required 98.0 runs without a fault (Figure 3). The proportion of streams alerting within 50 runs exceeded the fault-free proportion by 12.7 percentage points for Isolation Forest, 12.8 for Westgard, 17.4 for One-Class SVM, 19.0 for EWMA, and 30.1 for CUSUM (Table 2).

False-alarm–delay trade-off. Figure 4 presents the relationship between median delay and realized false-alarm rate. Increasing the threshold for Isolation Forest and One-Class SVM from a 0.5% to a 5% per-observation target increased the realized false-alarm rate from approximately 1.6 to 14.3 -14.5 per 100 analyte-runs, while reducing the median trend delay from 37.5 to 5.5 runs for Isolation Forest and from 45.5 to 6.0 runs for One-Class SVM (see Supplementary Table S4). At the realized rate for EWMA (5.2 per 100), linear interpolation of median delay against the logarithm of the realized false-alarm rate between the four threshold points for each curve yields trend delays of approximately 15.1 and 21.0 runs for Isolation Forest and One-Class SVM, respectively, which bracket the 19.8 runs observed for EWMA. At Westgard's rate (2.3 per 100), the interpolated values (29.6 and 38.3) similarly bracket its 30.0 runs. CUSUM (2.2 per 100) exhibited a slower response (44.0 runs) than both interpolated curves (30.6 and 39.3). In the remaining three scenarios, Westgard, Isolation Forest, and One-Class SVM already achieved median delays of at most 1.0 run at the 1% target, so further threshold adjustments could reduce delays by at most 1 run.

Table 3 presents exploratory combinations of Westgard and Isolation Forest alerts. When either method signaled (OR), the false-alarm rate was 4.27 (3.28–5.45) per 100 analyte-runs, with a trend delay of 18.0 (IQR 14.0–22.1) compared to the no-fault reference of 19.5. Requiring both methods to signal (AND) resulted in a false-

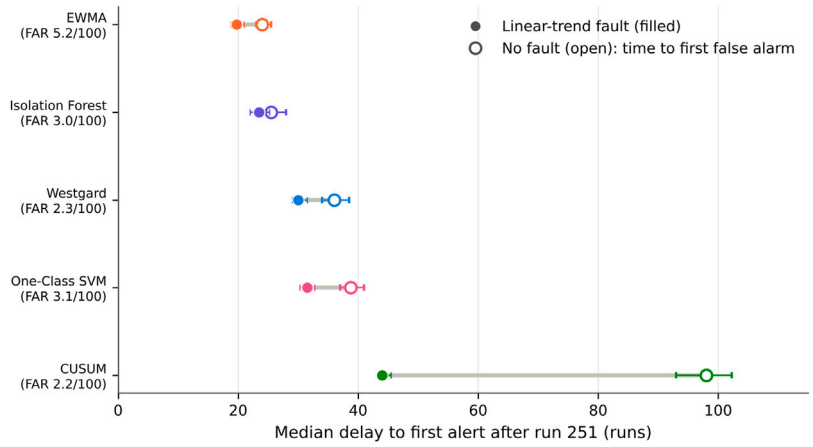


FIGURE 3
Linear trend: comparison of detection delay with and without a fault. Filled symbols indicate the median delay to the first alert after run 251 under the trend scenario. At the same time, open symbols represent the median time to the first alert after run 251 in fault-free streams (no-fault reference). Error bars denote bootstrap 95% confidence intervals over 300 replications. Methods are labelled with their false-alarm rate per 100 analyte-runs.

alarm rate of 1.01 (0.69–1.48) per 100 and a trend delay of 48.5 (IQR 40.4–56.5) compared to the no-fault reference of 69.0. Delays for other scenarios varied by at most one run. In the imprecision scenario, Isolation Forest achieved an AUC of 0.932 with all features (Table 4) and 0.697 with only z and |z| (One-Class SVM: 0.925 and 0.652). Engineered features also improved performance for trend (Isolation Forest 0.640–0.836) and step shift (0.843–0.987). All configurations separated outlier runs almost perfectly (AUC ≥0.999), reflecting the constructed ±4 SD excursions rather than real-world outlier detection. Omitting the EWMA feature altered the AUC by at most 0.03. Permutation of single features affected the Isolation Forest score most for z (median 0.0105), ewma (0.0087), and |z| (0.0060), and least for roll_SD (0.0011; Supplementary Table S5).

Table 5; Figure 5 report delays for various fault sizes, with confidence intervals provided in Supplementary Table S8. We generated the severity-sweep streams independently of the primary evaluation streams, using the same models and thresholds but new random draws. As a result, estimates at the same nominal fault size differed slightly. For an abrupt doubling of the SD, the CUSUM median delay was 3.0 runs (95% bootstrap CI 3.0–3.5) in the severity sweep and 3.5 runs (95% bootstrap CI 3.0–3.5) in the primary analysis. For a +2 SD trend, the Westgard median was 29.0 runs (95% bootstrap CI 28.5–31.0) in the sweep and 30.0 runs (95% bootstrap CI 29.0–31.5) in the primary analysis. For ±4 SD outliers, the CUSUM median was 12.0 runs (95% bootstrap CI 10.5–12.5) in the sweep and 12.5 runs (95% bootstrap CI 11.5–13.25) in the primary analysis. The confidence intervals overlapped in all three comparisons. All methods detected larger step shifts and abrupt increases in imprecision more quickly. For a +3 SD shift, Westgard, Isolation Forest, One-Class SVM, and EWMA had median delays of 0.0 runs, while CUSUM had a delay of 1.0 run. For a +1 SD shift, median delays ranged from 2.5 to 4.0 runs, with detection occurring within 50 runs in every replication. For a ×1.5 increase in SD, delays were 2.0 (Isolation Forest), 3.0 (Westgard), 3.5 (One-Class SVM), 4.0 (EWMA), and 8.5 runs (CUSUM). For trends, the final size had minimal effect on delay for most methods (EWMA 22.0 to 19.5, Isolation Forest 24.0 to 21.0, Westgard 33.2 to 28.0 runs from a final size of +1 to +3 SD), as the deviation remains small during the initial runs. Only CUSUM showed a marked change (63.0–35.0 runs). When the SD increased progressively, delays were substantially longer than after an abrupt increase (to ×2: 17.5 EWMA, 18.5 Isolation Forest, 24.0 Westgard, 24.5 One-Class SVM, and 47.5 CUSUM). Only 52% of CUSUM series alerted within 50 runs, and the ordering resembled that observed for trends. These slowly developing scenarios face the same false-alarm caveat as the primary trend scenario. For isolated outliers, delay depended on their size: at ±3 SD, the medians were 0.0 (Isolation Forest), 0.0 (Westgard), 3.0 (One-Class SVM), 7.0 (EWMA), and 33.2 runs (CUSUM), with 63% of CUSUM series alerting within 50 runs. At ±5 SD, all methods except CUSUM (2.0 runs) had a median delay of 0.0 runs.

Discussion

The main conclusion is that, contrary to initial expectations, machine learning-based anomaly detection does not replace traditional IQC monitoring; instead, the methods showed complementary detection profiles across fault types, supporting a layered, multi-method IQC strategy rather than reliance on any single approach. The ClinQC Pro interactive tool offers a thorough, 300-replication simulation benchmark of five IQC detection techniques spanning twelve biochemistry analytes, three QC levels, and five error scenarios. (https://muhammadhaseeb823.github.io/Clinqc-Pro/).

In this standardized simulation, no method consistently demonstrated the fastest detection across all scenarios. For step shifts, abrupt increases in imprecision, and isolated gross errors, Westgard multirules, Isolation Forest, and One-Class SVM responded within approximately one run (median 0.0–1.0), with the largest difference in median delay among these methods being one run. EWMA exhibited equal or slightly slower response times, while CUSUM, configured with a reference value of k = 0.5 and a decision interval of h = 5 applied to standardized z-scores, was the slowest in each scenario. Since a ±4 SD excursion inherently exceeds the 1–3s

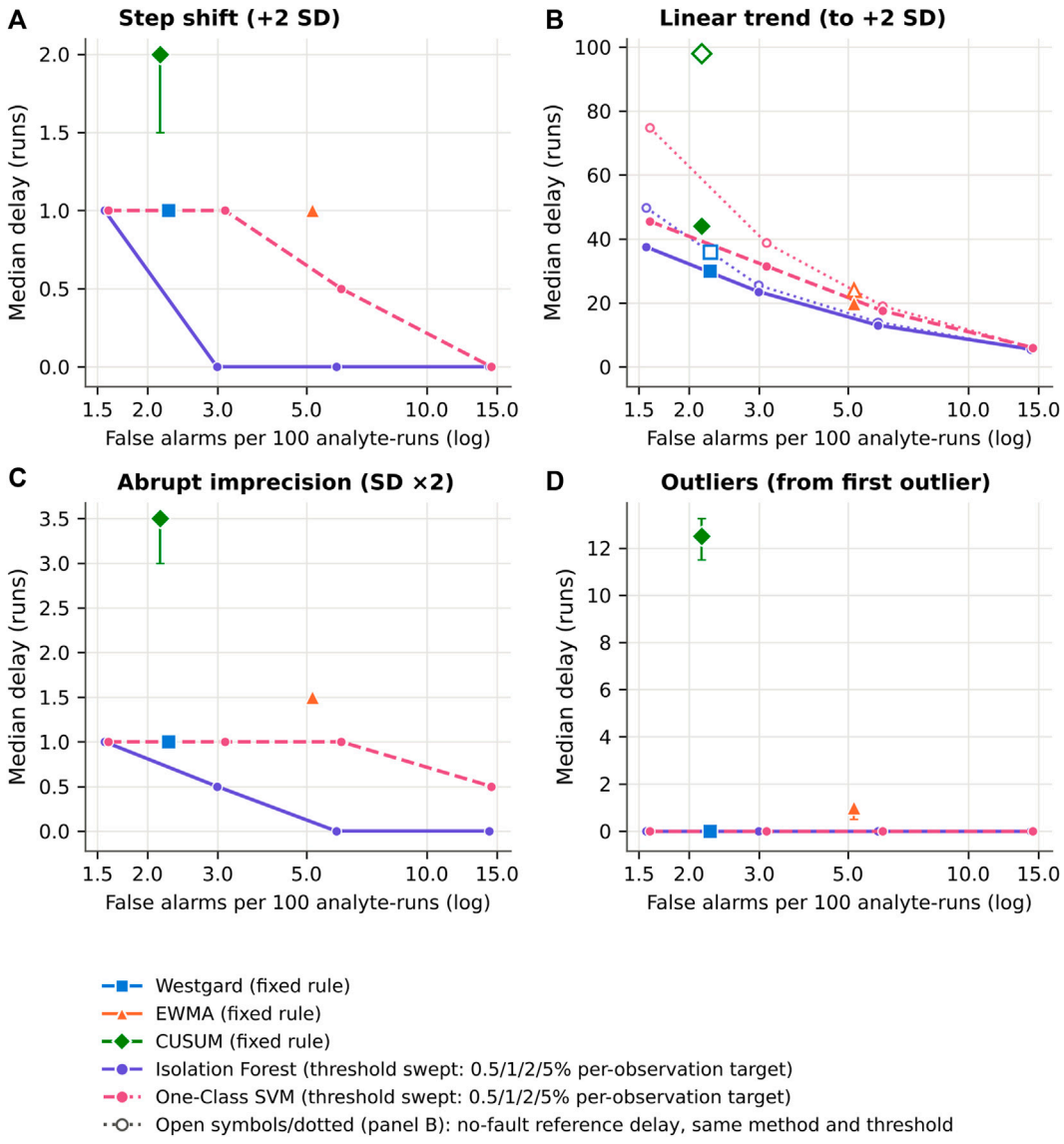


FIGURE 4
Median detection delay versus realized false-alarm rate. Panels display: **(A)** step shift, **(B)** linear trend, **(C)** abrupt imprecision increase, and **(D)** outliers (from the first injected outlier). Isolation Forest and One-Class SVM curves are plotted across four thresholds (0.5%, 1%, 2%, and 5% per-observation target). Westgard, EWMA, and CUSUM are shown as fixed rules, represented as single points with bootstrap 95% confidence intervals for the median. In Panel B, open symbols and dotted lines indicate the no-fault reference delay (time to the first alert from run 251 in fault-free streams) for each method and threshold. The x-axis represents the realized false-alarm rate per 100 analyte-runs (logarithmic scale).

TABLE 3 **Exploratory combinations of Westgard and Isolation Forest alerts.**

Strategy	False alarms per 100 analyte-runs	Shift	Trend	Imprecision	Outliers	No-fault reference
Westgard	2.26 (1.56–3.04)	1.0	30.0	1.0	0.0	36.0
Isolation forest	2.99 (2.07–4.10)	0.0	23.5	0.5	0.0	25.5
Westgard OR isolation forest	4.27 (3.28–5.45)	0.0	18.0	0.5	0.0	19.5
Westgard AND isolation forest	1.01 (0.69–1.48)	1.0	48.5	1.0	0.0	69.0

Reported metrics include median delay (runs) for each scenario, false-alarm rate per 100 analyte-runs (median; 2.5th–97.5th percentile), and the no-fault reference delay. OR, indicates an alert if either method triggers; AND, indicates an alert only if both methods trigger.

TABLE 4 Area under the ROC curve (AUC) by feature configuration, model, and scenario.

Model and features	Step shift	Linear trend	Abrupt imprecision	Outliers
Isolation forest: all six features	0.987 (0.977–0.994)	0.836 (0.819–0.852)	0.932 (0.916–0.946)	0.999 (0.999–1.000)
Isolation forest: z and |z| only	0.843 (0.816–0.865)	0.640 (0.614–0.664)	0.697 (0.688–0.705)	0.999 (0.997–1.000)
Isolation forest: no EWMA feature	0.971 (0.955–0.984)	0.808 (0.783–0.831)	0.938 (0.926–0.949)	0.999 (0.999–1.000)
One-class SVM: all six features	0.997 (0.995–0.998)	0.806 (0.770–0.831)	0.925 (0.904–0.943)	0.999 (0.994–1.000)
One-class SVM: z and |z| only	0.744 (0.692–0.806)	0.606 (0.576–0.633)	0.652 (0.633–0.672)	1.000 (1.000–1.000)
One-class SVM: no EWMA feature	0.996 (0.995–0.998)	0.803 (0.768–0.827)	0.922 (0.902–0.940)	0.999 (0.993–1.000)

Median (2.5th–97.5th percentile) values are reported over 300 replications. The AUC, distinguishes post-onset runs (251–500) of each fault scenario from runs 251–500 of the in-control scenario. For outliers, only runs with an injected outlier are included.

TABLE 5 Severity sweep.

Fault and size	Westgard	EWMA	CUSUM	Isolation Forest	One-Class SVM
Step shift, +1 SD	4.0 (3.0–4.5); 100.0%	2.5 (2.0–3.0); 100.0%	4.0 (4.0–5.0); 100.0%	3.0 (2.5–4.0); 100.0%	4.0 (3.5–5.0); 100.0%
Step shift, +1.5 SD	1.5 (1.0–2.0); 100.0%	1.5 (1.0–2.0); 100.0%	2.5 (2.0–3.0); 100.0%	1.0 (1.0–1.5); 100.0%	2.0 (1.0–2.5); 100.0%
Step shift, +2 SD	1.0 (0.5–1.0); 100.0%	1.0 (1.0–1.0); 100.0%	2.0 (1.4–2.0); 100.0%	0.0 (0.0–0.5); 100.0%	1.0 (0.5–1.0); 100.0%
Step shift, +3 SD	0.0 (0.0–0.0); 100.0%	0.0 (0.0–0.5); 100.0%	1.0 (1.0–1.0); 100.0%	0.0 (0.0–0.0); 100.0%	0.0 (0.0–0.0); 100.0%
Linear trend, to +1 SD	33.2 (25.0–42.5); 65.8%	22.0 (17.5–29.6); 79.4%	63.0 (53.4–72.0); 36.9%	24.0 (18.5–31.0); 77.3%	35.5 (26.5–46.5); 64.3%
Linear trend, to +1.5 SD	33.0 (26.0–39.6); 69.6%	21.0 (16.0–26.5); 86.3%	50.0 (44.0–56.5); 48.3%	23.5 (17.5–31.0); 80.5%	33.5 (25.5–41.0); 69.9%
Linear trend, to +2 SD	29.0 (23.5–36.0); 76.5%	20.8 (15.5–25.6); 91.3%	42.5 (38.0–48.0); 62.0%	23.5 (18.5–28.6); 86.3%	31.2 (26.0–37.5); 75.7%
Linear trend, to +3 SD	28.0 (22.5–33.5); 85.5%	19.5 (15.5–23.0); 97.6%	35.0 (31.9–38.5); 84.8%	21.0 (16.5–25.5); 94.2%	28.0 (22.5–33.5); 88.9%
Abrupt imprecision, SD ×1.5	3.0 (2.0–4.0); 100.0%	4.0 (3.0–5.5); 99.9%	8.5 (7.0–10.5); 98.2%	2.0 (1.5–3.0); 100.0%	3.5 (2.5–4.5); 100.0%
Abrupt imprecision, SD ×2	1.0 (0.5–1.0); 100.0%	2.0 (1.0–2.5); 100.0%	3.0 (2.5–4.0); 100.0%	1.0 (0.0–1.0); 100.0%	1.0 (1.0–1.5); 100.0%
Abrupt imprecision, SD ×3	0.0 (0.0–0.0); 100.0%	0.5 (0.0–1.0); 100.0%	1.0 (1.0–1.5); 100.0%	0.0 (0.0–0.0); 100.0%	0.0 (0.0–0.0); 100.0%
Progressive imprecision, SD to ×2	24.0 (20.0–29.5); 84.8%	17.5 (13.5–22.6); 88.8%	47.5 (40.5–55.0); 52.1%	18.5 (14.0–22.5); 94.1%	24.5 (19.5–31.0); 85.0%
Progressive imprecision, SD to ×3	20.0 (16.9–23.5); 97.1%	16.5 (13.0–20.0); 96.1%	35.0 (29.9–39.5); 76.9%	15.5 (12.5–18.5); 99.4%	20.0 (16.0–24.0); 97.8%
Outliers, ±3 SD	0.0 (0.0–2.5); 98.4%	7.0 (3.5–10.5); 92.4%	33.2 (22.9–43.0); 63.3%	0.0 (0.0–0.0); 99.9%	3.0 (1.0–6.5); 96.9%
Outliers, ±4 SD	0.0 (0.0–0.0); 100.0%	1.0 (0.0–2.5); 98.3%	12.0 (7.0–16.5); 85.6%	0.0 (0.0–0.0); 100.0%	0.0 (0.0–0.0); 100.0%
Outliers, ±5 SD	0.0 (0.0–0.0); 100.0%	0.0 (0.0–0.0); 99.8%	2.0 (1.0–3.5); 97.2%	0.0 (0.0–0.0); 100.0%	0.0 (0.0–0.0); 100.0%

Median delay in runs (IQR) and percentage of analyte streams with an alert within 50 runs, reported by fault size and detection method (300 replications; Isolation Forest and One-Class SVM, at the 1% per-observation threshold). Fault sizes are expressed in standard deviation units on the standardized scale. The sweep utilized newly generated streams, with a distinct random-number stream for each replication; therefore, values at the primary fault size may differ slightly from the primary results (see text).

limit, the equivalence of the three methods for outlier detection is expected and not particularly informative; this scenario primarily demonstrates that smoothing and cumulative statistics delay the response to isolated gross errors. Under the simulated conditions and parameter settings evaluated, CUSUM exhibited longer detection delays than the other methods for these fault scenarios. Its performance may vary with alternative settings and operating points, and reducing the decision interval would decrease delay at the expense of increased false alarms.

Interpretation of delay for slowly developing faults requires consideration of the false-alarm rate. All methods generated alerts in fault-free streams, with rates neither close to 1% nor equivalent: aggregating three QC levels converts a 1% per-observation target into approximately 3 alerts per 100 analyte-runs. EWMA, with limits estimated from 240 autocorrelated observations per series, exhibited the highest realized rate in this implementation (5.2 per 100). This result should not be interpreted as EWMA being inherently prone to false alarms, as limits estimated from longer training streams or set at matched false-alarm rates would yield different outcomes. In the trend scenario, the mean increased by only 0.008 SD per run (for a final size of +2 SD; the severity sweep indicated minimal dependence of delay on final size), resulting in frequent false alarms within the first few tens of runs. For Westgard, EWMA, Isolation Forest, and One-Class SVM, the time to first alert in fault-free streams ranged from 24.0 to 38.8 runs, within approximately 2–7 runs of their trend delays. The ranking in Table 1 (with EWMA as the fastest) primarily reflects EWMA's higher alarm rate. When delay is plotted against the realized false-

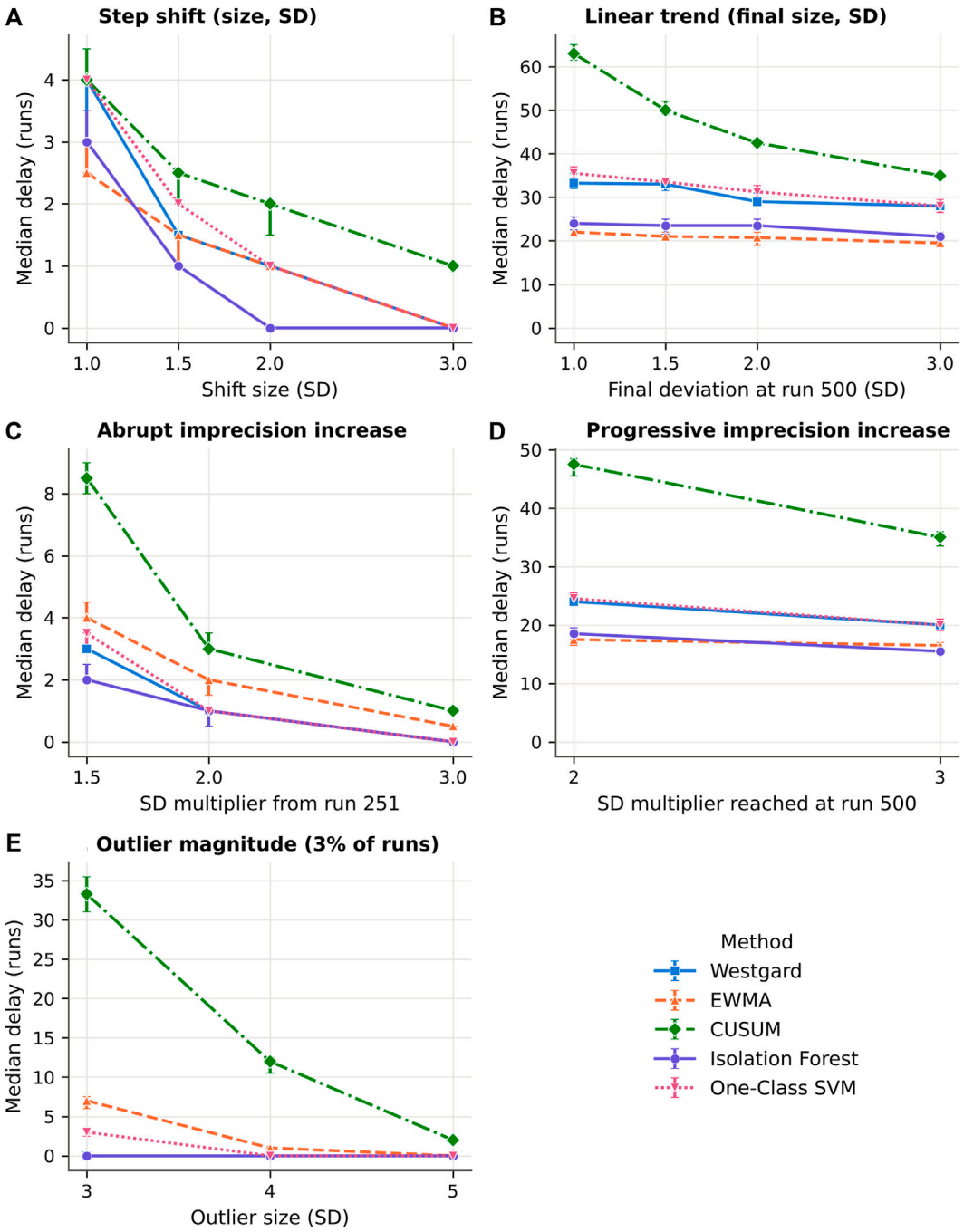


FIGURE 5
Severity sweep: median detection delay (95% bootstrap confidence interval over 300 replications) as a function of fault size for **(A)** step shift, **(B)** linear trend, **(C)** abrupt imprecision increase, **(D)** progressive imprecision increase, and **(E)** outlier magnitude. We evaluate Isolation Forest and One-Class SVM at the 1% per-observation threshold.

alarm rate (Figure 4B), EWMA and Westgard fall within the range defined by the two machine-learning methods, while CUSUM lies above this range. CUSUM's low alarm rate resulted in the largest excess detection within 50 runs, but also the longest delay. Future benchmarks should compare methods at matched realized false-alarm rates and include a no-fault reference.

We revised the Westgard comparator during the study: R-4s was initially described as a range between consecutive runs and coded as a range greater than 4 SD across QC levels within a run, neither of which aligns with the conventional rule. We now evaluate it within a run across QC levels. We applied rules to each QC-level stream, then aggregated alerts per analyte-run. A fully joint multilevel implementation, in which 2-2s was also applied across levels within a run and 4-1s and 10x to the chronological sequence of results from the three levels, produced a similar median false-alarm rate (2.29 versus 2.26 per 100 analyte-runs across replications) and similar delays (see

Supplementary Table S3). Laboratory rule sets vary, so these results pertain only to the configuration used in this study.

We evaluated combinations of Westgard and Isolation Forest alerts as exploratory analyses. Applying an OR rule reduced the median trend delay to 18.0 runs, but resulted in 4.27 alerts per 100 analyte-runs and a no-fault reference of 19.5 runs. Conversely, an AND rule decreased the false-alarm rate to 1.01 per 100, but increased the trend delay to 48.5 runs (no-fault reference 69.0). These outcomes primarily reflect changes in the alert rate. Staff workload, alert fatigue, and clinical consequences were not assessed; therefore, the findings indicate that alert layering is a testable strategy, but its benefit remains unestablished.

Engineered temporal features substantially improved discrimination in the machine-learning models: the area under the curve (AUC) for imprecision increased from approximately 0.65–0.70 with z and |z| alone to about 0.93 when all features were included. In contrast, the EWMA feature was largely redundant. The ablation analysis compared feature sets within two machine-learning models only and did not demonstrate an advantage of the algorithms over conventional methods. The permutation analysis provided a descriptive measure of the Isolation Forest's sensitivity to each feature in the in-control distribution; however, correlated features can cause single-feature permutations to leave the data manifold, and the results cannot identify reagent, calibration, or analyzer failure mechanisms. For these reasons, we removed the SHAP and counterfactual analyses from the initial submission, which used 20 replications.

A shadow-mode evaluation, in which machine-learning alerts are collected alongside routine internal quality control (IQC) and reviewed by senior staff before any action, represents a potential next step. The proposed duration for such an evaluation period (for example, approximately 30 working days) is a recommendation by the authors and not an ISO 15189 requirement or a direct outcome of this study; the duration should be determined by local risk assessment and workload considerations. Any implementation would necessitate local verification, documented threshold rationale, change control, staff competency assessment, and evidence that false alarms are manageable [15].

This study utilized simulated quality control (QC) series standardized to z-scores and generated from common Gaussian distributions with prespecified faults. The 12 analytes and 3 QC levels were treated as exchangeable, and the study did not model analyte-specific coefficient of variation (CV) or bias, total allowable error, concentration-dependent variance, non-Gaussian noise, autocorrelation, reagent-lot effects, maintenance and calibration events, seasonal variation, instrument aging, correlated multi-analyte drift, or combined faults. The severity sweep addressed shift size, trend size, and the size and trajectory of the SD increase, but not combined faults; the outlier rate and magnitude were assumed. Thresholds were calibrated using simulated in-control data and do not establish generalizability to real data. False-alarm rates were not matched across methods, making delays operating-point specific, with trend results particularly dependent on the alarm rate. We report delays in QC runs; we did not convert them to patient-sample exposure or working days. We removed the SHAP and counterfactual analyses. We fitted the machine-learning models on pooled series and did not evaluate per-analyte models. The study employed a single implementation of each method and one Westgard rule configuration. Real laboratory QC data (such as LIMS logs from multiple instruments and sites) are necessary before concluding clinical or operational value.

In a standardized simulation, Westgard multirules, EWMA, CUSUM, Isolation Forest, and One-Class SVM exhibited fault-type- and threshold-dependent behavior, with no method being the fastest for every fault. Differences in delay for slowly developing trends were primarily attributable to differences in false-alarm rate, indicating that methods should be compared at matched false-alarm rates and with a no-fault reference. The findings support further evaluation of complementary monitoring strategies rather than replacement of validated IQC procedures, and highlight the need for prospective validation with real, analyte-specific laboratory QC data. This work advances biomedical science by demonstrating, through standardized simulation, that internal quality control (IQC) detection delay depends on both fault type and alerting rate. By systematically comparing five IQC methods across four simulated analytical fault types, the results indicate that no single method consistently achieves the fastest detection in all scenarios. These findings underscore the need to evaluate detection performance at equivalent false-alarm rates and establish a reproducible benchmark for comparing conventional and anomaly-detection approaches. The provided browser-based application enhances reproducibility and offers an educational platform for exploring the methods and results. Because the study uses synthetic data, further assessment with real laboratory quality-control data is needed to determine its applicability and potential clinical or operational benefits.

Summary table

What is known about this subject

- Both Westgard and EWMA are well-known internal quality control techniques.
- Although unsupervised anomaly detection shows promise for IQC, systematic benchmarking across several analytes is still lacking.
- Despite being advised in quality control guidelines, CUSUM methods are still not widely used in UK biochemistry labs.

What this work adds

- Five IQC techniques were benchmarked across twelve analytes with 300 replications.
- No single technique prevailed with all simulated types most rapidly.
- Engineered temporal features improved machine-learning discrimination for several faults.

Data availability statement

Raw simulation outputs and all model code are accessible via the ClinQC Pro application at https://muhammadhaseeb823.github.io/Clinqc-Pro/ and are additionally archived with a permanent DOI on Zenodo [DOI: 10.5281/zenodo.23351635]. Supplementary Tables S1–S9; Supplementary Figure S1 are provided in a separate Supplementary Material file accompanying this submission.

Author contributions

MHT and CP: analysis, writing. MA, AA, MH, and AL: critical review. MHT and MMT: machine learning development, model development.

Funding

The author(s) declared that financial support was not received for this work and/or its publication.

Conflict of interest

The author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declared that generative AI was used in the creation of this manuscript. Generative AI Disclosure: Generative AI tools were not used in the preparation of analytical code or model outputs. Standard word processing assistance was used for manuscript formatting.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontierspartnerships.org/articles/10.3389/bjbs.2026.17568/full#supplementary-material

References

1. Kinns H, Pitkin S, Housley D, Freedman DB. Internal quality control: best practice. *J Clin Pathol* (2013) 66(12):1027–32. doi: 10.1136/jclinpath-2013-201661

2. Ricós C, Fernández-Calle P, Perich C, Westgard JO. Internal quality control - past, present and future trends. *Adv Lab Med* (2022) 3(3):243–52. doi: 10.1515/almed-2022-0029

3. Buttner J, Borth R, Toronto C, Boljtwell Atlanta JH, Ga USA, Broughton PMG, et al. Approved recommendation (1978) on quality control in clinical chemistry. *Clinica Chim Acta*. (1979) 98(1–2):129–43. doi: 10.1016/0009-8981(79)90174-8

4. Ramírez BS, Ramírez JG. *Douglas Montgomery's Introduction to Statistical Quality Control: A JMP Companion*. Cary, NC: SAS Institute Inc (2018).

5. Westgard JO, Groth T, Aronsson T, Falk H, de Verdier CH. Performance characteristics of rules for internal quality control: probabilities for false rejection and error detection. *Clin Chem* (1977) 23(10):1857–67. doi: 10.1093/clinchem/23.10.1857

6. Parvin CA. What's new in laboratory statistical quality control guidance? The 4th edition of CLSI C24, statistical quality control for quantitative measurement procedures: principles and definitions. *J Appl Lab Med* (2017) 1(5):581–4. doi: 10.1373/jalm.2016.022269

7. Gruber L, Hausch A, Mueller T. Internal quality controls in the medical laboratory: a narrative review of the basic principles of an appropriate quality control plan. *Diagnostics* (2024) 14(19):2223. doi: 10.3390/diagnostics14192223

8. Javed H, Ismail M, Saeed N. New extended exponentially weighted moving average control chart for monitoring process mean. *Heliyon* (2024) 10(14):e34424. doi: 10.1016/J.HELIYON.2024.E34424

9. Clinical and Laboratory Standards Institute (CLSI). *Statistical Quality Control for Quantitative Measurement Procedures: Principles and Definitions*. 4th ed. Wayne, PA: CLSI. (2016).

10. Lorde N, Mahapatra S, Kalaria T. Machine learning for patient-based real-time quality control (PBRTQC), analytical and preanalytical error detection in clinical laboratory diagnostics. *Multidisciplinary Digital Publishing Inst (Mdpi)* (2024) 14:1808. doi: 10.3390/diagnostics14161808

11. Duan X, Zhang M, Liu Y, Zheng W, Lim CY, Kim S, et al. Next-generation patient-based real-time quality control models. *Ann Lab Med Korean Soc Lab Med* (2024) 44:385–91. doi: 10.3343/alm.2024.0053

12. Liu FT, Ting KM, Zhou ZH. Isolation forest. In: *Proceedings - IEEE International Conference on Data Mining, ICDM* (2008). p. 413–22. doi: 10.1109/ICDM.2008.17

13. Xu Y, Dong H, Zhou M, Xing J, Li X, Yu J. Improved isolation forest algorithm for anomaly test data detection. *J Computer Commun* (2021) 09(08):48–60. doi: 10.4236/jcc.2021.98004

14. Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC. Estimating the support of a high-dimensional distribution. *Neural Comput* (2001) 13(7):1443–71. doi: 10.1162/089976601750264965

15. Xie H, Jia Y, Liu S. Integration of artificial intelligence in clinical laboratory medicine: advancements and challenges. In: *Interdisciplinary Medicine*. John Wiley and Sons Inc (2024). doi: 10.1002/INMD.20230056

16. Wood R. A simulation study of the westgard multi-rule quality-control system for clinical laboratories. *Clin Chem* (1990). 36 (3):462–465. doi: 10.1093/clinchem/36.3.462

17. Poh DKH, Lim CY, Tan RZ, Markus C, Loh TP. Internal quality control: moving average algorithms outperform westgard rules. *Clin Biochem* (2021) 98:63–9. doi: 10.1016/j.clinbiochem.2021.09.007

18. Aarsand AK, Webster C, Fernandez-Calle P, Jonker N, Diaz-Garzon J, Coskun A, et al. *EFLM Biological Variation Database* (2026). Available online at: https://biologicalvariation.eu (Accessed November 13, 2026).

19. Cava CRF, García-Lario JV, Hernández A, Simón Va CVJJMCPM, Hernández A, Jiménez CV, et al. Current databases on biological variation: pros, cons and progress. *Scand J Clin Lab Invest* (1999) 59(7):491–500. doi: 10.1080/00365519950185229

20. Enderlein G, Hawkins DM. Identification of outliers. *Biometrical J* (1987) 29(2):198. doi: 10.1002/bimj.4710290215

21. Lu JH, Lin CY. A novel moving average–exponentiated exponentially weighted moving average (MA-Exp-EWMA) control chart for detecting small shifts. *Mathematics* (2025) 13(18):3049. doi: 10.3390/math13183049

22. Van Andel E, Henricks LM, Giliams APM, Noordervliet RM, Mensink WJ, Filippo D, et al. Moving average quality control of routine chemistry and hematology parameters - a toolbox for implementation. *Clin Chem Lab Med* (2022) 60(11):1719–28. doi: 10.1515/cclm-2022-0655

23. Collins GS, Moons KGM, Dhiman P, Riley RD, Beam AL, Van Calster B, et al. TRIPOD+AI statement: updated guidance for reporting clinical prediction models that use regression or machine learning methods. *BMJ* (2024) 385:e078378. doi: 10.1136/bmj-2023-078378