Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

The Cost of Waiting in Early Log Anomaly Recognition: Stopping Rules and Evaluation Populations

Jufeng Li

q3529416243@gmail.com

China University of Petroleum, East China

Wenjie Li

China University of Petroleum, East China

Junhao Zhan

China University of Petroleum, East China

Research Article

Keywords: Log anomaly detection, Early classification, Sequential decisions, Benchmarking, Reproducibility, Evaluation population

Posted Date: November 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11535197/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

The Cost of Waiting in Early Log Anomaly Recognition: Stopping Rules and Evaluation Populations

Wenjie Li[1], Jufeng Li[2*], Junhao Zhan[3]

[1]College of Mechanical and Electronic Engineering, China University of Petroleum (East China), Qingdao, 266580, China.
[2]College of Science, China University of Petroleum (East China), Qingdao, 266580, China.
[3]Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, 266580, China.

*Corresponding author(s). E-mail(s): q3529416243@gmail.com;
Contributing authors: dvien5162@gmail.com; zelos0216@gmail.com;

Abstract

A log monitor must decide whether another event is worth observing. We study this decision using a common prefix scorer and explicit false-alarm, missed-anomaly, waiting and scoring charges. On 575,061 Hadoop Distributed File System (HDFS) sessions, a tuned two-threshold rule has reference costs of 0.1305, 0.2343 and 0.4186 at false-negative to false-positive cost ratios of 5, 10 and 20. A grouped multi-step anticipation rule costs an additional 0.01150, 0.000068 and 0.01202 per session. At ratio 5, its missed-anomaly savings are outweighed by waiting and scoring charges. However, exact-sequence cluster intervals include zero on all five reference splits, independent group-disjoint draws change the comparison sign, and alternative population weights can reverse the ratio-5 ordering. We reconstruct the saved decisions, perform a fresh raw-data rerun, and add validation-budgeted false-alarm controls. The contribution is a reproducible analysis of when stopping-rule comparisons depend on the scorer, price and evaluation population, rather than a new detector or a general claim against anticipation. Completed-session replay and corpus-derived templates limit deployment interpretation.

Keywords: Log anomaly detection, Early classification, Sequential decisions, Benchmarking, Reproducibility, Evaluation population

MSC Classification: 68M20 , 68M15

1 Introduction

Logs are an important source of evidence about faults in distributed storage and computing systems. A complete sequence can often be classified accurately, but a monitor encounters that sequence incrementally. After observing a prefix, it may alert, stop with a normal decision, or obtain another event. Additional evidence can reduce errors while consuming monitoring effort and delaying a decision. An early recognizer should therefore be evaluated by the decisions it makes and the evidence it purchases, as well as its final classification accuracy.

This setting is a binary instance of early classification with paid observations. It does not introduce a new action space: a binary early classifier that stops and emits its predicted class already has two terminal outcomes and a continuation action. The open empirical question is whether an explicit forecast of future classification cost improves on a tuned reactive stopping band for a specified log scorer and target population. A two-threshold rule is itself a variable-time wait/commit policy; comparing it with an anticipatory rule measures the incremental value of anticipation.

We investigate that question on Hadoop Distributed File System (HDFS) block sessions from Loghub [1, 2]. The corpus contains 575,061 labeled sessions, including 16,838 anomalies. A calibrated count-based scorer supplies a probability at each causal prefix. Several stopping rules consume the same score trajectory, and all fitted choices use training or validation information. We explicitly price missed anomalies, false alerts, additional events and scorer evaluations. The protocol is an offline replay of completed sessions with an observable end-of-session assumption. It does not measure time before a fault or real fault-onset delay.

The reference comparison favors a tuned two-threshold rule over the tested grouped-anticipation adaptation in mean cost. That observation is informative, but its interpretation depends on the evaluation unit. In the reference session-random split, 97.61% of seed-11 test sessions have an identical full event sequence in training. The same corpus contains 18,373 distinct sequence groups with highly concentrated frequency weights. Treating those groups as the resampling unit substantially widens uncertainty, and changing the group draw or population weights changes some comparison signs. We therefore study the conditional result and its boundaries together.

The paper makes three contributions. First, it provides a reproducible cost ledger and a comparison of stopping rules on shared causal prefix scores. Second, it reconstructs the reference result from saved per-session decisions and separates that audit from a fresh raw-data rerun and new, explicitly exploratory, validation-budget controls. Third, it quantifies how exact-sequence composition and the target weighting affect the apparent value of anticipation. The practical output is a set of checks for

evaluating an early log decision rule, rather than a recommendation that one rule should be deployed in every system.

2 Related work

2.1 Log scoring and evaluation

DeepLog uses next-event prediction for log anomaly detection [3]; LogAnomaly combines sequential and quantitative signals [4]; LogRobust targets unstable logs [5]. LogBERT and LAnoBERT use masked-language objectives [6, 7], while PLELog estimates training labels in a semi-supervised setting [8]. NeuralLog reduces dependence on explicit parsing [9], and recent work combines larger language models with log representations [10, 11]. These studies motivate possible scoring functions. Their published complete-sequence or forecasting results do not establish the behavior of a causal prefix scorer under our stopping costs.

Critical evaluations are particularly relevant. Le and Zhang show that data selection and evaluation choices affect deep log detector results [12]. Landauer et al. demonstrate that simple event, length and count features explain many anomalies in common benchmarks [13]; their survey also describes recurring evaluation problems [14]. SemPCA demonstrates the value of comparatively simple representations [15], and Khan et al. distinguish parsing accuracy from representation distinguishability [16]. Xiao et al. examine split, visibility and component-cost sensitivity on HDFS and BGL [17]. We build on these lessons but evaluate per-session stopping cost, rather than claiming that protocol sensitivity is a new discovery.

2.2 Early classification and sequential decisions

ECONOMY formulates non-myopic early classification using anticipated misclassification and delay costs [18, 19]. Calimera learns future cost differences [20], while TEASER uses a learned reliability gate [21]. The early classification of time series (ECTS) survey and benchmark distinguishes classifiers from triggers, evaluates asymmetric misclassification and delay settings, and identifies a simple probability-threshold trigger as a strong competitor [22]. Our grouped-anticipation rule is an adaptation of this literature. Its failure to improve a particular log benchmark does not falsify the original algorithms, which are not faithfully reimplemented here.

The sequential probability ratio test (SPRT) and change-detection theory provide established sequential decision frameworks [23, 24]. A learned log score does not automatically satisfy the likelihood or independence assumptions under which those methods are optimal. Observation control across multiple data streams is also established [25]; our unit is one completed log session.

Selective prediction studies terminal rejection rather than buying the next temporal observation [26–28]. Ren et al. apply conformity-score p-values to complete HDFS count vectors [29]; that procedure should not be described as strict split-conformal calibration with a disjoint calibration set. Xin studies conformal log alert control on complete windows [30]. Neither cited procedure directly validates the probability of ever alerting during the repeated prefix decisions evaluated here. DQNLog uses binary

sequence-classification actions [31]; BranchyNet reduces network depth for a complete input [32]. Those are different decision units, without implying that the present literature review exhausts all stopping methods.

3 Decision problem and replay protocol

3.1 Observable information and outcomes

A session is an event sequence $x_{1:L}$ with eventual session label $y \in \{0, 1\}$, where 1 denotes anomaly. At prefix t, a scorer computes

$$p_t = f(x_{1:t}). \tag{1}$$

The feature vector uses the counts of events already observed and the observed prefix length. Full length L is not a scorer feature. Labels are attached to sessions; they do not identify the event at which a fault began. Observation counts and the retrospective ratio t/L are therefore recognition measures, not time-to-failure or fault-onset delay.

Decisions begin at $s = \min(3, L)$ and are queried after each subsequent event. The replay cap is $T = \min(L, 60)$. Alert and Normal are terminal, and a terminal decision is forced at T. The use of the completed session to observe its end is explicit. In particular, the replay does not solve the problem of knowing online that no further event will arrive for a block. A production system would require an independently specified close marker, censoring rule or inactivity timeout. No pre-end decision should inspect L or the number of events still to come.

3.2 Cost ledger and stopping-time errors

For a stopping time τ, let $w = \tau - s$ be additional events waited. The charged cost is

$$\text{Cost}_i = C_{\text{FP}}\mathbf{1}\{\widehat{y}_i = 1, y_i = 0\} + C_{\text{FN}}\mathbf{1}\{\widehat{y}_i = 0, y_i = 1\} + c_d w_i + c_q q_i. \tag{2}$$

We set $C_{\text{FP}} = 1$, $C_{\text{FN}} \in \{5, 10, 20\}$, $c_d = 0.05$ and $c_q = 0.01$ in the reference comparison. These are declared design charges, not measured seconds, CPU time or economic losses. Waiting is charged for normal and anomalous sessions alike. The initial observation up to s is common and uncharged. A sequential rule makes $q = w + 1$ score evaluations; a fixed-prefix rule scores once, so $q = 1$, while still paying $c_d(\tau - s)$. Maintaining prefix features and evaluating an anticipation table are not separately measured by this ledger. These omissions limit hardware-cost interpretations.

Each session receives one terminal decision. We report

$$\text{FAR}_{\text{stop}} = \Pr(\text{the policy alerts} \mid y = 0), \tag{3}$$

estimated over normal sessions. A marginal false-positive rate (FPR) at one prefix is not the same quantity. The reference comparison minimizes validation cost and reports the achieved stopping-time FPR; it is not an FPR-matched comparison. Section 6 separately selects rules under a validation FPR budget. Such validation constraints do not guarantee that test FPR is below the same number.

3.3 Splits, representation and fitting

The reference seeds are 11, 23, 37, 42 and 101. Sessions are split by block identifier, stratified by label, into 60% training, 20% validation and 20% test. Validation is divided into equal tuning and reserved evaluation halves using seed 13; the main tuning and calibration use the tuning half. Seed 11 has 345,036 training, 115,012 validation and 115,013 test sessions. The five splits reuse one corpus and are not five independent deployment systems.

The reference representation uses the distributor's 29 event types. That vocabulary was produced upstream from the corpus and is a genuine preprocessing-transduction limitation. Sharing it controls an experimental condition; it does not prove that transduction has equal effects on every stopping rule. We therefore distinguish this representation from a training-only signature-parser sensitivity and avoid calling the primary evaluation leakage-free.

The common scorer is a stochastic-gradient-descent (SGD) logistic model with $\log(1 + \text{event count})$ features and $\log(1 + t)$, trained with balanced class weights on training prefixes. The historical implementation uses three training epochs, ℓ_2 regularization with $\alpha = 10^{-4}$, and batches of 8,192 sessions. A logistic Platt map is fitted on tuning prefixes. Calibration fitting and threshold selection on the tuning data are both development operations; neither uses test labels. A fitted map need not be calibrated within each time point, sequence group or shifted population.

Exact sequence groups are defined by the complete ordered event sequence, including its length. In the reference split, 112,269 of 115,013 seed-11 test sessions have an exact training duplicate: 98.22% of normal and 77.55% of anomalous test sessions. Block-disjoint splitting therefore differs substantially from sequence-group-disjoint splitting. Near-duplicate edit-distance overlap is not measured.

4 Stopping rules and analysis

4.1 Comparators

The fixed-prefix rule B1 selects a prefix count and threshold on tuning cost, then scores once. B2 alerts when $p_t \geq h$, stops normally when $p_t \leq \ell$, and otherwise waits. Its independent lower and upper thresholds are tuned jointly. B5 is a symmetric probability trigger: it alerts at $p_t \geq u$, stops normally at $p_t \leq 1 - u$, and waits otherwise. In the continuous parameterization B5 is a constrained B2 family; finite candidate grids need not have identical optima. B5 uses $u = 0.50, 0.52, \ldots, 0.98$.

A unigram SPRT comparator accumulates training-fitted event log-likelihood ratios. We do not claim its theoretical optimality on these learned, dependent event sequences. The original boundary grid and its subsequent extension are separated in the supplement. A rare-template rule alerts on the first training-normal-rare event and otherwise waits until the cap. Always-Normal and Always-Alert are useful feasibility and prevalence controls.

The v2 adaptation groups continuation trajectories by eight event-count buckets and empirical score bins. The buckets are $[3, 4)$, $[4, 6)$, $[6, 9)$, $[9, 13)$, $[13, 19)$,

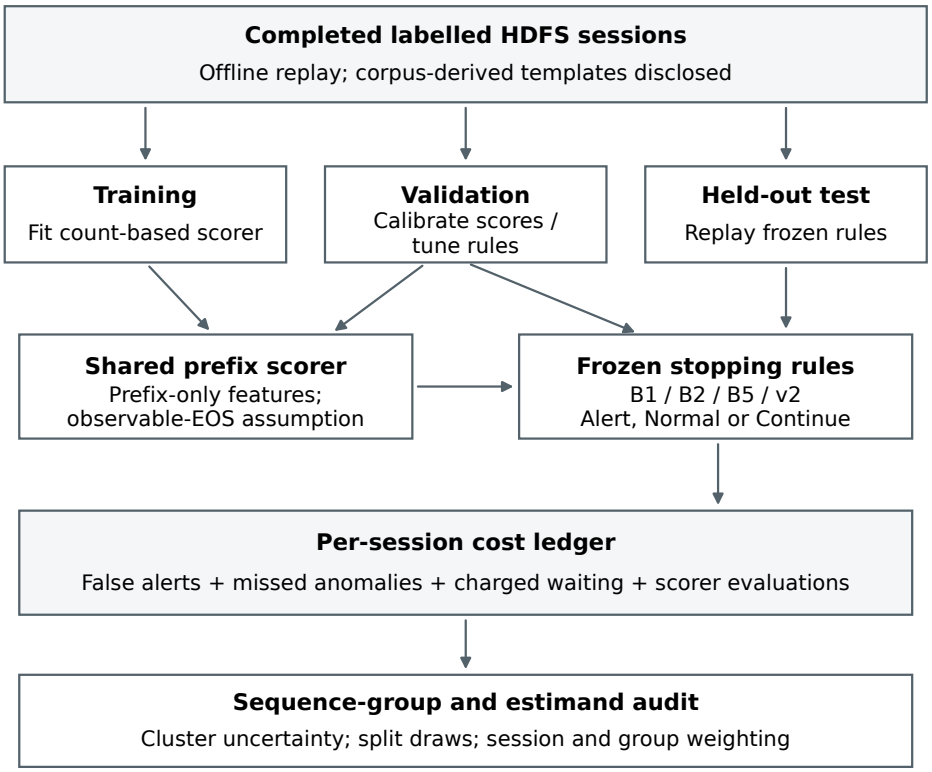


Fig. 1 Study framework for completed-session replay. Training and tuning produce a shared prefix scorer and fixed stopping-rule parameters. Held-out sessions produce a per-session cost ledger. Sequence-group composition and target weights are audited separately. The diagram describes an offline evaluation, not an implemented live alerting service.

$[19, 28)$, $[28, 41)$ and $[41, 61)$. Interior score quantiles are fitted from training-plus-tuning decision pairs, with duplicate cut points removed. For candidate horizons $a \in \{1, 2, 3, 5, 8, 13, 21\}$, the table estimates

$$A_g(a) = \frac{1}{|g|} \sum_{(i,t)\in g} \left[r\big(p_{i,\min(t+a,T_i)}\big) + (c_d + c_q)\big(\min(t + a, T_i) - t\big) \right], \tag{4}$$

where $r(p) = \min\{(1-p)C_{\mathrm{FP}}, pC_{\mathrm{FN}}\}$. Groups with fewer than 30 pairs fall back to the score-bin mean and then the global mean. The rule waits if some estimated anticipated cost is strictly below the current terminal Bayes risk; otherwise it takes the lower-risk terminal action. The current evaluation fee is common to the current choice; the forecast charges future evaluations while waiting. Its table is an approximation to future risk, not a certified value function.

This is an ECONOMY-inspired adaptation with coarse continuation grouping, rather than a faithful implementation of ECONOMY or Calimera. Its Bayes terminal action is also less freely tunable than B2's band in the reference experiment. Consequently, calibration error and continuation-table approximation can contribute to the observed difference. We do not identify all of that difference as an inherent cost of anticipation.

An earlier one-step prototype, v1, is retained only in the implementation audit and supplementary reference table. Its terminal-length-dependent tie heuristic is audited separately in the raw rerun. The core B2–v2 contrast does not use that heuristic.

4.2 Tuning and uncertainty

Reference search budgets per cost ratio differ: B1 searches 285 candidates, B2 64 and the original SPRT 72. The anticipatory rule fits its continuation table rather than using an identical threshold search. Common tuning data do not imply equal computation or equal model flexibility. Full grids, tie handling and forced actions are specified in the released implementation and supplement.

Let $d_i = \mathrm{Cost}_i(\mathrm{B2}) - \mathrm{Cost}_i(\mathrm{v2})$; positive means v2 is cheaper. We report paired effects per split and exact-sequence cluster-resampling intervals, with the fitted scorer and rule parameters held fixed. The latter resample the observed sequence groups and recompute the session-weighted statistic; they describe sensitivity to the sampled pattern composition under that resampling model. Identical event text does not by itself prove that separate physical sessions are dependent, and distinct sequences need not be independent. Thus group resampling is a stated target-population sensitivity, rather than a universal correction for every possible data-generating process. Historical session-resampling analyses condition on the frequency distribution of the observed sequences; the revised supplement reports the newly reconstructed group-resampling intervals.

A sequence-weight concentration measure is

$$G_{\mathrm{conc}} = \left(\sum_{g=1}^{G} s_g^2 \right)^{-1}, \qquad s_g = n_g/N. \tag{5}$$

Its value of 19.8 is the number of equally weighted groups with the same concentration. It is not an estimated statistical effective sample size or proof that only twenty independent observations exist.

4.3 Population-weight sensitivity

A session-weighted mean addresses the frequency-weighted traffic represented by the corpus. Equal sequence-group weights address pattern coverage; square-root weights and bounded group weights define still other targets. None is intrinsically correct for all deployments. With per-group mean effect d_g and covariance defined using denominator G,

$$\Delta_{\text{equal}} - \Delta_{\text{session}} = \sum_g \left(\frac{1}{G} - s_g \right) d_g = -G \operatorname{Cov}_G(s_g, d_g). \tag{6}$$

This identity explains the difference between two weighting targets; it is not a new algorithm or a generalization guarantee.

We distinguish pre-normalization truncation, $w_g = \min(n_g, cN) / \sum_h \min(n_h, cN)$, from a true final-weight bound, $w_g = \min(\lambda n_g, c)$ with $\sum_g w_g = 1$. The former can exceed nominal c after normalization. The latter requires $c \geq 1/G$. Cap values and all signs are reported as exploratory sensitivity; no application in this study establishes a preferred cap.

5 Reference reconstruction and sensitivity results

5.1 Costs and the price of evidence

All 90 original method–ratio–seed costs are reconstructed from saved per-session predictions, with maximum absolute discrepancy 2.3×10^{-16} against their accompanying CSV, at floating-point precision. This is an audit of saved decisions, not independent validation of their training history. The extended SPRT is additionally replayed from saved likelihood trajectories; its amended mean costs are 0.3780, 0.4898 and 0.7135, with 5.268 evaluations per session. The revised figures and supplementary table use the amended rule consistently.

Table 1 Reference costs (mean±standard deviation across five overlapping splits) and achieved errors at ratio 10. Cost units follow Eq. 2. These are reconstructed saved decisions, not independent systems.

Method	Cost @5	Cost @10	Cost @20	Recall @10	FPR @10	Evals @10
B1 fixed prefix	0.1314±0.0014	0.2442±0.0023	0.4285±0.0046	0.3710	0.00031	1.00
B2 two thresholds	0.1305±0.0015	0.2343±0.0024	0.4186±0.0046	0.3708	0.00010	1.67
B5 symmetric	0.1306±0.0015	0.2343±0.0023	0.4185±0.0046	0.3708	0.00009	1.67
v2 anticipation	0.1420±0.0013	0.2343±0.0023	0.4306±0.0147	0.3708	0.00020	1.67

B2 has reference mean costs of 0.13053, 0.23428 and 0.41861 at ratios 5, 10 and 20. Relative v2 excesses are 0.0115037, 0.0000680 and 0.0120181. Ratio 10 is a practical tie. B5 also nearly coincides with B2: their pooled costs differ by at most 1.2×10^{-4}. These are properties of the evaluated implementations and population, not a claim of unique superiority of an independently parameterized B2.

At ratio 5, v2 raises recall from 0.1780 to 0.3708 at approximately the same achieved false-alarm rate. It reduces false-negative cost by 0.0282316 per session, but adds 0.0330881 in waiting and 0.0066176 in scoring charges, and adds 0.0000296 in false-positive cost. Its net extra cost is therefore 0.0115037. Under the declared objective, this is a price-of-evidence result rather than a failure to recognize additional anomalies.

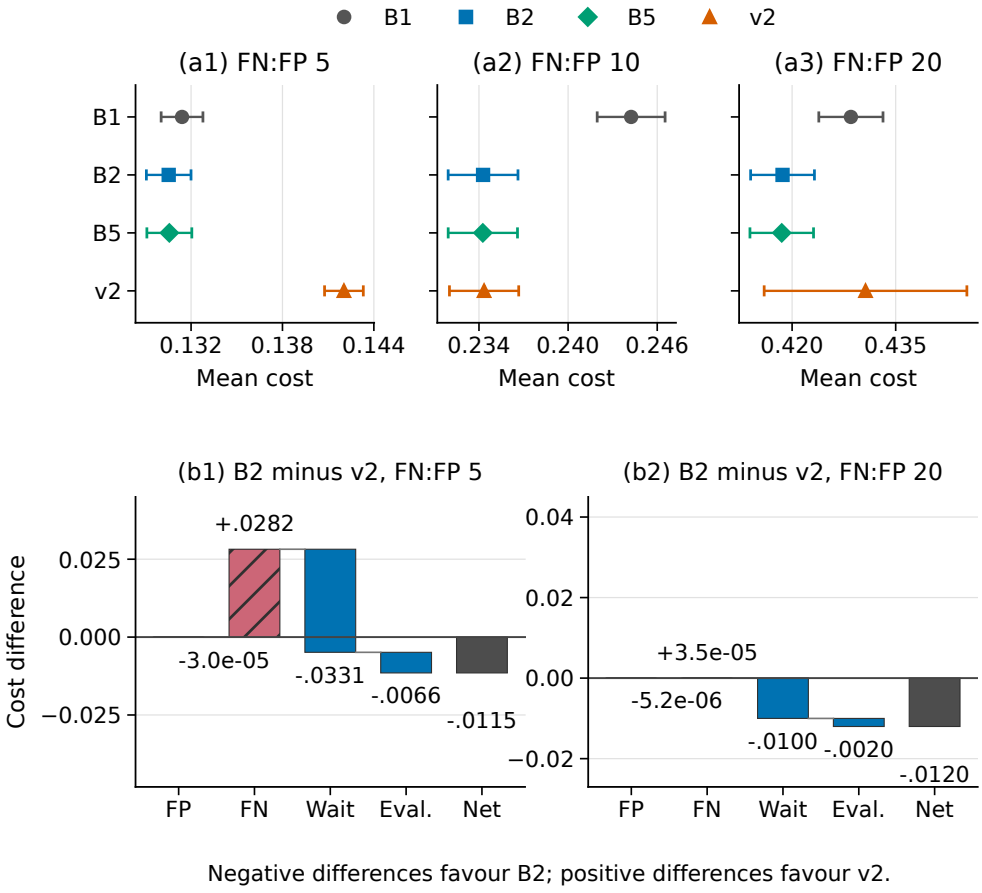


Fig. 2 Reference cost comparison and cost decomposition from saved per-session decisions. Bars summarize the five overlapping splits; variability across seeds is not variability across independent systems. The component comparison uses B2 minus v2, so positive values favor v2. At ratio 5 the missed-anomaly saving is outweighed by additional waiting and scoring charges.

Holding all actions fixed, the component ledger implies a waiting-price crossing of 0.0326166 per event at ratio 5. This is a counterfactual repricing of fixed decisions; it is neither a measured operational price nor the optimum after retuning both policies. Earlier per-price retuning controls are described separately in the supplement. We do not extrapolate their finite grid to every possible price regime.

5.2 Sequence composition and uncertainty

Figure 3 shows cluster-resampling intervals from the five reference splits. At ratio 5 all include zero; at ratio 20 the seed-37 interval also includes zero. At ratio 10 effects are small. Narrow session-level intervals therefore do not establish superiority over a population of unknown sequence patterns. The observed corpus supports a descriptive conditional ordering and a warning about evaluation units.

Two sequence-group-disjoint split constructions give an important boundary. The earlier reported construction favors B2 over v2 in 13 of 15 seed–ratio cells; the other two cells differ by no more than 1.1×10^{-4}. Its per-seed effects are available as rounded historical reports rather than a complete prediction set, and Figure 4 distinguishes their provenance. A second construction, available as saved predictions, favors v2 on three of five seeds at ratios 10 and 20. Its mean B2–v2 effects are -0.01601, $+0.00498$ and $+0.04182$ at ratios 5, 10 and 20. Large within-family variability accompanies the reversal. These are different held-out populations, not independent replications of one fixed test set.

The mechanism is visible in the independent grouped draws: on two seeds B2's validation-selected band has almost no test recall, while v2 retains appreciable recall. The validation and test prevalence and pattern mix differ substantially. This does not establish that v2 will generalize better; it shows that selecting a band on one pattern mixture can fail on another. Shared preprocessing or shared duplication cannot logically be assumed to cancel such interactions.

5.3 Changing the target weights

The corpus contains 18,373 exact sequence groups. The largest holds 94,794 sessions, or 16.48% of the corpus; $G_{\mathrm{conc}} = 19.78$ describes their concentration. Figure 5 relates that concentration to target-weight sensitivity.

On the reference random splits, session, equal-group and square-root weighting favor B2 at all three cost ratios. At ratio 5, true hard caps of 0.5%, 1% and 2% instead make v2 cheaper by 0.01655, 0.01353 and 0.00467. Using the unrounded bounded-weight B2 costs, these are reductions of 5.303%, 5.091% and 2.086%. Tighter 0.1%–0.2% or looser caps of at least 5% do not reverse that comparison. The dependence is therefore non-monotonic, and it does not justify choosing a favorable cap after viewing the outcomes. Equation 6 is verified numerically in every reference seed and ratio.

5.4 End-of-session and representation sensitivities

Only a small fraction of B1/B2 decisions occur exactly at the observed session end, but that does not eliminate the observability assumption. Charging a five-event time-out surcharge to such decisions raises reference mean cost by about 0.0013–0.0027 per

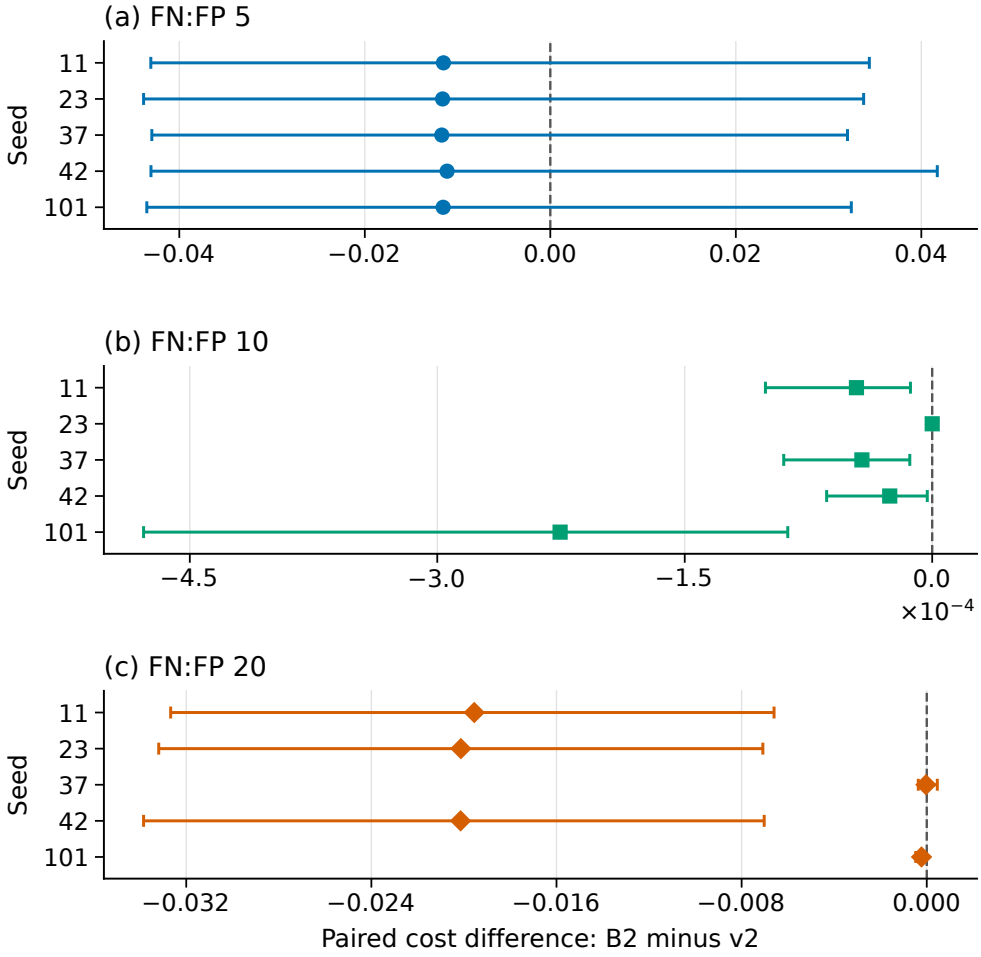


Fig. 3 B2–v2 effects and 95% exact-sequence cluster-resampling intervals by reference seed. Positive favors v2. Scorers and tuned policies are held fixed during resampling. The resampling model targets changes in observed pattern composition; it does not establish independent-system uncertainty.

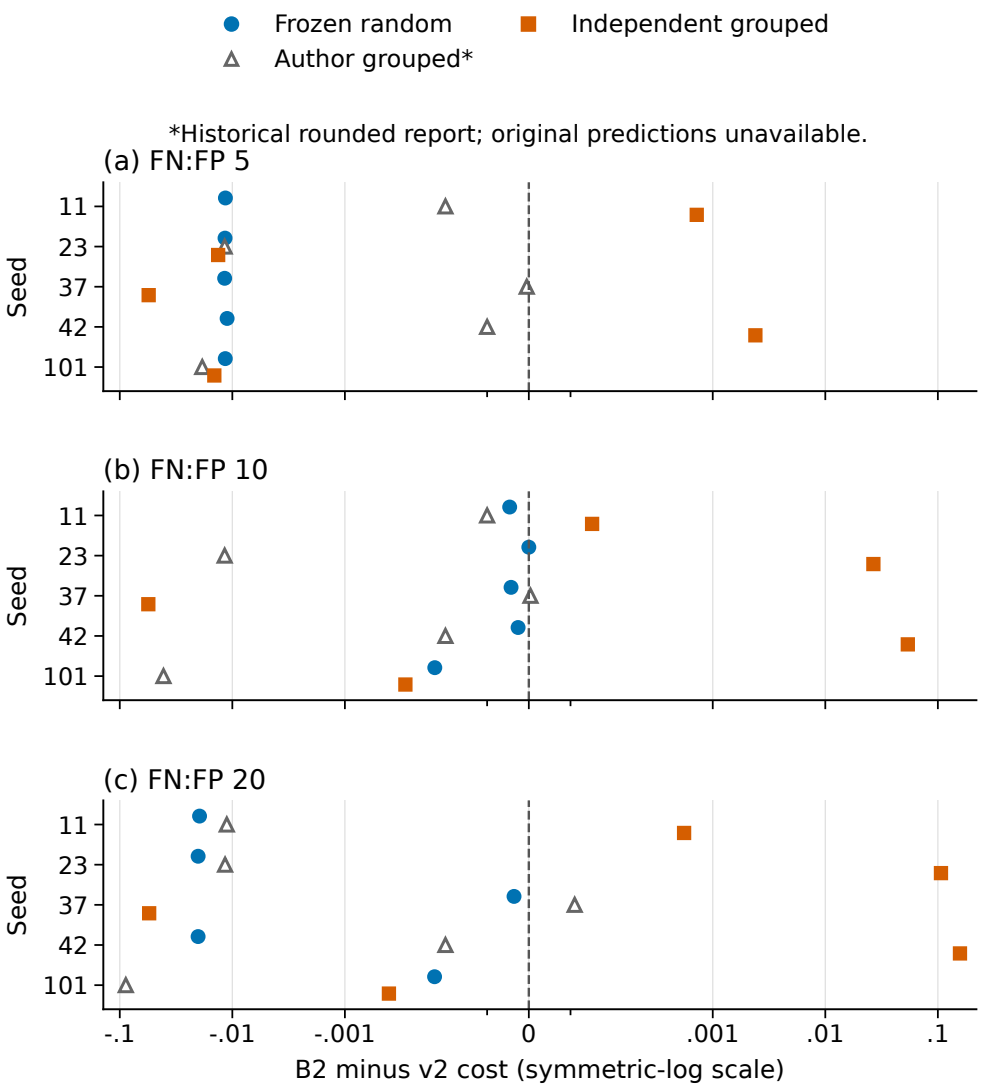


Fig. 4 Paired B2–v2 effects across the reference session split and two sequence-group-disjoint draw constructions. Each point is a seed. Open symbols for the earlier grouped draw use rounded values reported in the original manuscript; the other two families use reconstructible outputs. Positive favors v2. Changing the draw can change both prevalence and the distribution of difficult patterns, so a single group-disjoint split does not identify a population-invariant ranking.

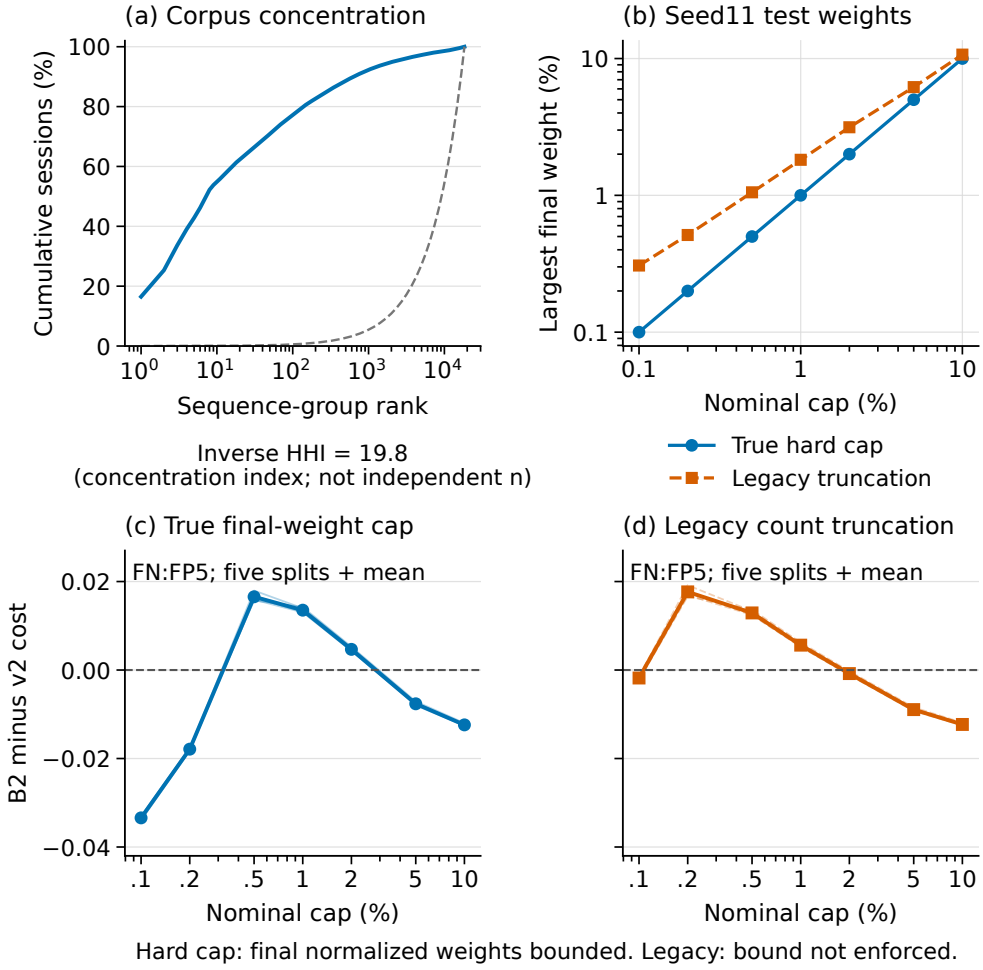


Fig. 5 Sequence-frequency concentration and population-weight sensitivity. The concentration statistic is not a statistical sample size. True final normalized caps and pre-normalization truncation are distinct weight families. Positive paired effect favors v2; cap values are exploratory target definitions, not recommended operating settings.

session; the ratio-5 and ratio-20 B2–v2 directions remain. This surcharge is an accounting sensitivity, not a simulated wall-clock timeout service. The supplement provides end fractions, charge definitions and limits.

A seed-11 signature parser fits its vocabulary on training data only, masks block identifiers, addresses and numeric literals, and retains 28 signatures plus an unknown slot. Relative to the distributor representation, v2–B2 cost differences are 0.0160580, 0.0030904 and 0.0120710 at the three ratios. B2 remains cheaper than v2 in that control, while B1 becomes cheaper than B2 at ratio 20. The control establishes representation sensitivity and one observed non-reversal of the core contrast; it does not prove that corpus-derived preprocessing never affects rankings. Additional training-only reruns, when completed, are separately identified in the supplement.

6 Fresh rerun and validation-budget controls

6.1 Provenance and raw-data execution

The fresh rerun starts from the public HDFS archive with SHA-256 `04f919f2185821f23f045dca611a7586429bdabc601bd7b43f30005f8e289b01`. It uses new output directories, records package versions and saves the tuning and test trajectories. It is performed after inspection of the reference results and is explicitly exploratory. Reusing the public corpus does not create a new external validation population, even if training is rerun on a different platform.

Table 2 Fresh raw-data rerun versus the archived reference outputs. All five seeds were refitted in new directories. The comparison covers 90 shared method–ratio–seed cells; new B4/B5 outputs are retained separately.

Quantity	Maximum absolute difference
Expected cost	0
Recall	0
Stopping-time FPR	0
Mean decision step	0
Mean evaluations	0
Mean observed fraction	2.2×10^{-16}

The cost, confusion counts, recall, stopping-time FPR, decision steps and evaluation counts match the archived 90 cells exactly in this rerun. The largest difference among the reported observation-ratio means is 2.3×10^{-16}, attributable to floating-point aggregation. The environment used Python 3.13.5, NumPy 2.1.3 and scikit-learn 1.6.1. The match is a measured outcome, not a portability requirement. An implementation audit found that the historical one-step prototype used final session length to break exact continuation-risk ties before EOS. The revised implementation stops on equality without inspecting remaining length. A synthetic same-prefix/different-future test detects the old dependency and passes for the corrected rule. No exact tie occurred in the 30 real validation/test–ratio audit cells, and the correction changes no

observed decisions or costs. The v1 reference row is therefore retained for provenance, without treating the historical code as generally causal.

6.2 A common validation false-alarm budget

For budgets $\alpha \in \{0.001, 0.01\}$, we select the lowest validation-cost candidate satisfying $\widehat{\mathrm{FAR}}_{\mathrm{stop_{val}}} \leq \alpha$ within each declared family. B2 and B5 use their stated threshold grids. For v2 we additionally search a declared alert-guard grid, which also binds forced terminal alerts. Always-Normal is reported separately as a feasible zero-FPR floor. Candidates and tie handling are specified before computing these new test summaries; they were not selected to improve test cost. An infeasible family is marked as such, rather than silently relaxed.

Table 3 Validation-budgeted controls on the fresh raw rerun, averaged over five seeds. The budget constrains tuning stopping-time FPR; achieved test FPR is reported separately and is not guaranteed.

Budget	FN:FP	Method	Cost	Recall	Test FPR	Extra events
0.001	5	B2	0.13053	0.1780	0.00007	0.002
0.001	5	B5	0.13057	0.1781	0.00016	0.001
0.001	5	v2 + guard	0.14204	0.3708	0.00010	0.664
0.001	10	B2	0.23428	0.3708	0.00010	0.666
0.001	10	B5	0.23426	0.3708	0.00009	0.666
0.001	10	v2 + guard	0.23434	0.3708	0.00020	0.665
0.001	20	B2	0.41852	0.3708	0.00010	0.666
0.001	20	B5	0.41850	0.3708	0.00009	0.666
0.001	20	v2 + guard	0.43063	0.3710	0.00030	0.866
0.01	5	B2	0.13053	0.1780	0.00007	0.002
0.01	5	B5	0.13057	0.1781	0.00016	0.001
0.01	5	v2 + guard	0.14204	0.3708	0.00010	0.664
0.01	10	B2	0.23428	0.3708	0.00010	0.666
0.01	10	B5	0.23426	0.3708	0.00009	0.666
0.01	10	v2 + guard	0.23434	0.3708	0.00020	0.665
0.01	20	B2	0.41861	0.3710	0.00034	0.665
0.01	20	B5	0.41850	0.3708	0.00009	0.666
0.01	20	v2 + guard	0.43063	0.3710	0.00030	0.866

All selected validation operating points are feasible in the declared grids. Across the 90 algorithm seed–ratio–budget cells, no achieved test FPR exceeds the corresponding budget. Always-Normal is separately retained as a zero-FPR feasibility floor in the released data. The selected v2 alert guard is zero in these controls, so the constraint does not create different v2 trajectories. At ratio 20, the stricter budget changes the B2 operating point slightly; the other pooled comparisons are effectively unchanged. B2 retains lower pooled cost at ratios 5 and 20, while ratio 10 remains a practical tie. This is the observed result at two validation budgets, not dominance over an entire controlled-FPR frontier.

These controls compare finite candidate families, not globally optimal policies. In particular, an alert guard can change the decisions and observations purchased by v2,

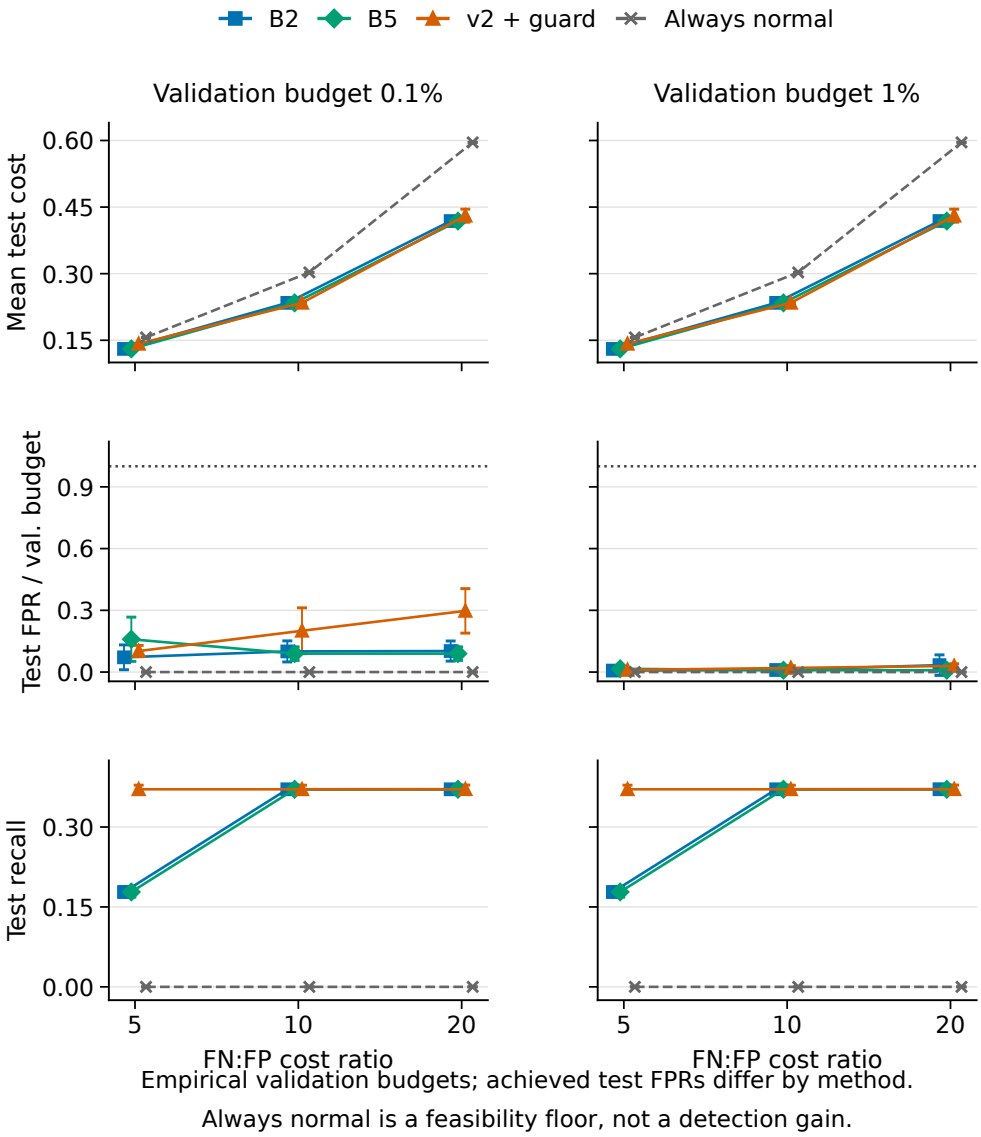


Fig. 6 New validation-budget controls. Parameters minimize tuning cost subject to the indicated tuning stopping-time FPR budget and are then frozen. Test cost, recall and achieved stopping-time FPR are descriptive outcomes; a tuning budget does not confer a finite-sample or anytime-valid test guarantee.

so its result is a separately tuned implementation rather than a guarantee about its original Bayes threshold.

6.3 Calibration diagnostic

A shared fitted score is not a guarantee of shared calibration error. We report a post-hoc time-specific diagnostic on the fixed retrospective cohort $L \geq 20$, retaining the same sessions at every plotted prefix. This selection uses eventual length only for analysis and is not an implementable admission rule for live sessions. It avoids conflating a change in the eligible population with changes in prefix information.

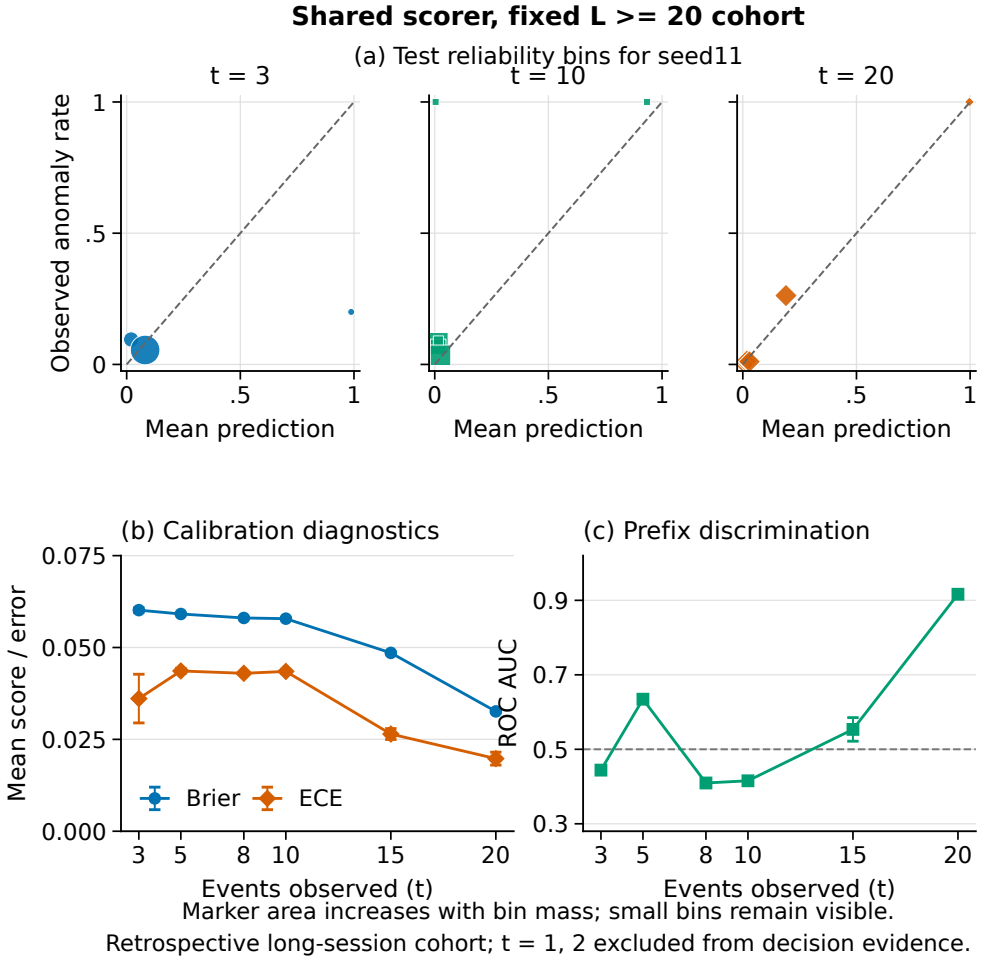


Fig. 7 Time-specific calibration diagnostic on a fixed retrospective long-session cohort. Each displayed estimate uses observed labels at a declared prefix. Sparse score bins are reported rather than interpolated as evidence. The cohort is selected using eventual length for analysis only; the plot is not a live triggering rule or evidence that Platt fitting yields perfect conditional calibration.

On this fixed cohort, mean test ROC-AUC across the five splits is 0.444 at event 3, 0.415 at event 10 and 0.916 at event 20. Mean Brier score falls from 0.0602 at event 3 to 0.0326 at event 20. The non-monotonic early discrimination and remaining calibration error caution against interpreting pooled prefix probabilities as ideal time-conditional posteriors. These are properties of this scorer and selected cohort, rather than limits on all available log information.

The diagnostic evaluates the scoring signal on which every stopping comparison depends. It does not isolate all sources of approximation error in the anticipation table or establish that recalibration would improve the policy. That would require a separate training-only or validation-only intervention followed by untouched evaluation.

7 Interpretation and limitations

The stable finding in the reference replay is that extra detections can cost more evidence than they save under a declared charge vector. The unstable finding is the ranking across different evaluation populations. A simple band is a necessary baseline to tune at the same prices; it is not a universally preferred operational rule. A method claim should state its scorer, decision cadence, attainable false-alarm region and target pattern mixture.

Several boundaries matter. First, the headline data come from one labeled HDFS corpus. The five seeds are overlapping partitions, and group-disjoint draws remain draws from that corpus. Historic BGL and GRU controls are separated in the supplementary provenance record and are not used here as independently reproduced cross-system support. Second, the reference template vocabulary is corpus-derived; a limited training-only parser control does not remove all preprocessing concerns. Third, HDFS session labels do not provide fault onset, and the observed end of a completed session is assumed. No measured live-system lead time or calibrated wall-clock delay is claimed.

Fourth, the continuation estimator uses coarse groups and score-derived Bayes risks. A flexible terminal threshold can compensate for score miscalibration differently from a Bayes terminal action. Thus a negative result for this estimator is not evidence against anticipation as a whole. Fifth, the computation charge counts scorer calls and omits table lookup, feature maintenance, training and storage costs. Hardware timings would require an additional benchmark. Sixth, alternative population weights are exploratory questions, with no single cap justified by an operator in this study. Unknown future patterns are not covered by reweighting known groups.

The retrospective calibration and raw rerun strengthen traceability but are not new confirmatory tests. Future evaluations should reserve genuinely unseen systems or sequence populations, select a causal end protocol, and specify whether the target is traffic frequency, rare-pattern coverage or a resource-constrained mixture. Reporting both achieved errors and observation costs is necessary for interpreting a purported early-recognition gain.

8 Conclusion

On a shared count-based HDFS prefix scorer, the evaluated grouped-anticipation rule has higher reference mean cost than a tuned two-threshold rule under the base charges; one cost ratio is an effective tie. At the lowest ratio it detects more anomalies, but the additional waiting and scoring charges outweigh the missed-anomaly saving. The saved ledger is reproducible, while sequence-cluster uncertainty, independent group draws and bounded-weight targets show why the comparison is conditional.

The contribution is a reproducible study of this dependence, including a fresh raw-data rerun and explicitly validation-budgeted controls. It does not establish a new detector, a population-wide stopping-policy ranking or a live failure-prediction system. The evidence supports evaluating the price and population of an early decision before attributing a gain or loss to anticipation.

Supplementary information. Online Resource 1 contains method grids, per-seed reference results, uncertainty and weight definitions, replay-boundary sensitivities, provenance and the revision audit. The accompanying reproducibility archive contains code, derived tables, prediction artifacts and execution instructions.

Statements and Declarations

Funding

This research received no external funding.

Competing interests

The authors declare that they have no competing interests.

Ethics approval and consent

Not applicable. The study uses public system-log benchmark data and does not recruit human participants.

Data availability

The public HDFS dataset is available from the Loghub archive at https://zenodo.org/records/8196385. The accompanying reproducibility archive provides the derived per-session decisions, split identifiers, analysis tables and scripts used for the results reported here. The original public log archive is identified by its checksum and is not redistributed in the manuscript source package.

Code availability

Analysis, figure generation, saved-decision reconstruction and raw-data rerun scripts are included in the accompanying reproducibility archive. Scripts distinguish saved-evidence verification from raw-data training and write outputs to a user-selected working directory.

Author contributions

Wenjie Li: Conceptualization, Methodology, Writing—original draft. Jufeng Li: Project administration, Validation, Writing—review and editing, Corresponding author. Junhao Zhan: Software, Data curation, Visualization.

Use of generative artificial intelligence

OpenAI Codex was used to assist with manuscript revision, reference checking, code review, analysis and figure-generation scripts, and preparation of submission materials. This assistance went beyond language editing. The named human authors retain responsibility for the scientific claims, code, numerical outputs and submitted work. No synthetic observations were added to the benchmark data.

References

[1] Zhu, J., He, S., He, P., Liu, J., Lyu, M.R.: Loghub: A large collection of system log datasets for AI-driven log analytics. In: 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), pp. 355–366. IEEE (2023). https://doi.org/10.1109/ISSRE59848.2023.00071

[2] Xu, W., Huang, L., Fox, A., Patterson, D., Jordan, M.I.: Detecting large-scale system problems by mining console logs. In: Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles, pp. 117–132. ACM (2009). https://doi.org/10.1145/1629575.1629587

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

[4] Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., Chen, Y., Zhang, R., Tao, S., Sun, P., Zhou, R.: LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, pp. 4739–4745. International Joint Conferences on Artificial Intelligence Organization (2019). https://doi.org/10.24963/ijcai.2019/658

[5] Zhang, X., Xu, Y., Lin, Q., Qiao, B., Zhang, H., Dang, Y., Xie, C., Yang, X., Cheng, Q., Li, Z., Chen, J., He, X., Yao, R., Lou, J.-G., Chintalapati, M., Shen, F., Zhang, D.: Robust log-based anomaly detection on unstable log data. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 807–817. ACM (2019). https://doi.org/10.1145/3338906.3338931

[6] Guo, H., Yuan, S., Wu, X.: LogBERT: Log anomaly detection via BERT. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE

(2021). https://doi.org/10.1109/IJCNN52387.2021.9534113

[7] Lee, Y., Kim, J., Kang, P.: LAnoBERT: System log anomaly detection based on BERT masked language model. Applied Soft Computing **146**, 110689 (2023) https://doi.org/10.1016/j.asoc.2023.110689

[8] Yang, L., Chen, J., Wang, Z., Wang, W., Jiang, J., Dong, X., Zhang, W.: Semi-supervised log-based anomaly detection via probabilistic label estimation. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), pp. 1448–1460. IEEE (2021). https://doi.org/10.1109/ICSE43902.2021.00130

[9] Le, V.-H., Zhang, H.: Log-based anomaly detection without log parsing. In: 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 492–504. IEEE (2021). https://doi.org/10.1109/ASE51524.2021.9678773

[10] Guan, W., Cao, J., Qian, S., Gao, J.: LogLLM: Log-Based Anomaly Detection Using Large Language Models. Version 1 (2024). https://arxiv.org/abs/2411.08561v1

[11] Hadadi, F., Xu, Q., Bianculli, D., Briand, L.: LLM meets ML: Data-efficient anomaly detection on unstable logs. ACM Transactions on Software Engineering and Methodology **35**(8), 1–48 (2026) https://doi.org/10.1145/3771283

[12] Le, V.-H., Zhang, H.: Log-based anomaly detection with deep learning: How far are we? In: Proceedings of the 44th International Conference on Software Engineering, pp. 1356–1367. ACM (2022). https://doi.org/10.1145/3510003.3510155

[13] Landauer, M., Skopik, F., Wurzenberger, M.: A critical review of common log data sets used for evaluation of sequence-based anomaly detection techniques. Proceedings of the ACM on Software Engineering **1**(FSE), 1354–1375 (2024) https://doi.org/10.1145/3660768

[14] Landauer, M., Onder, S., Skopik, F., Wurzenberger, M.: Deep learning for anomaly detection in log data: A survey. Machine Learning with Applications **12**, 100470 (2023) https://doi.org/10.1016/j.mlwa.2023.100470

[15] Yang, L., Chen, J., Gao, S., Gong, Z., Zhang, H., Kang, Y., Li, H.: Try with simpler - an evaluation of improved principal component analysis in log-based anomaly detection. ACM Transactions on Software Engineering and Methodology **33**(5), 1–27 (2024) https://doi.org/10.1145/3644386

[16] Khan, Z.A., Shin, D., Bianculli, D., Briand, L.C.: Impact of log parsing on deep learning-based anomaly detection. Empirical Software Engineering **29**(6), 139 (2024) https://doi.org/10.1007/s10664-024-10533-w

[17] Xiao, H., Sung, J., Li, Z., Qian, G., Xu, C.: Protocol-Sensitive Evaluation of Log Anomaly Detection: Component Costs and Target-Access Sensitivity on HDFS and BGL. Version 1; accepted at DASC 2026 (2026). https://arxiv.org/abs/2610.05807v1

[18] Dachraoui, A., Bondu, A., Cornuéjols, A.: Early classification of time series as a non myopic sequential decision making problem. In: Machine Learning and Knowledge Discovery in Databases. Lecture Notes in Computer Science, vol. 9284, pp. 433–447. Springer (2015). https://doi.org/10.1007/978-3-319-23528-8_27

[19] Achenchabe, Y., Bondu, A., Cornuéjols, A., Dachraoui, A.: Early classification of time series: Cost-based optimization criterion and algorithms. Machine Learning **110**(6), 1481–1504 (2021) https://doi.org/10.1007/s10994-021-05974-z

[20] Bilski, J.M., Jastrzębska, A.: CALIMERA: A new early time series classification method. Information Processing & Management **60**(5), 103465 (2023) https://doi.org/10.1016/j.ipm.2023.103465

[21] Sch"afer, P., Leser, U.: TEASER: Early and accurate time series classification. Data Mining and Knowledge Discovery **34**(5), 1336–1362 (2020) https://doi.org/10.1007/s10618-020-00690-z

[22] Renault, A., Bondu, A., Cornuéjols, A., Lemaire, V.: Early classification of time series: A survey and benchmark. Transactions on Machine Learning Research (2025)

[23] Wald, A.: Sequential Analysis. John Wiley and Sons, New York (1947)

[24] Tartakovsky, A., Nikiforov, I., Basseville, M.: Sequential Analysis: Hypothesis Testing and Changepoint Detection. Chapman and Hall/CRC (2014). https://doi.org/10.1201/b17279

[25] Tsopelakos, A., Fellouris, G.: Sequential Anomaly Identification with Observation Control under Generalized Error Metrics. Version 1 (2024). https://arxiv.org/abs/2412.04693v1

[26] Geifman, Y., El-Yaniv, R.: SelectiveNet: A deep neural network with an integrated reject option. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 2151–2159. PMLR (2019). https://proceedings.mlr.press/v97/geifman19a.html

[27] Franc, V., Prusa, D., Voracek, V.: Optimal strategies for reject option classifiers. Journal of Machine Learning Research **24**(11), 1–49 (2023)

[28] Feng, J., Sondhi, A., Perry, J., Simon, N.: Selective prediction-set models with coverage rate guarantees. Biometrics **79**(2), 811–825 (2023) https://doi.org/10.

1111/biom.13612

[29] Ren, Y., Gu, Z., Wang, Z., Tian, Z., Liu, C., Lu, H., Du, X., Guizani, M.: System log detection model based on conformal prediction. Electronics **9**(2), 232 (2020) https://doi.org/10.3390/electronics9020232

[30] Xin, Q.: Log anomaly detection with conformal alert control and evidence-grounded incident ticket generation. Aviation Electronics, Information Technology, Telecommunications, Electricals, and Controls (AVITEC) **8**(2), 247–264 (2026) https://doi.org/10.28989/avitec.v8i2.3974

[31] He, Y., Pei, X.: Semi-Supervised Learning via DQN for Log Anomaly Detection. Version 2 (2024). https://arxiv.org/abs/2401.03151v2

[32] Teerapittayanon, S., McDanel, B., Kung, H.T.: BranchyNet: Fast inference via early exiting from deep neural networks. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 2464–2469. IEEE (2016). https://doi.org/10.1109/ICPR.2016.7900006