**Academic Journal of Applied Sciences**
ISSN: 3105-0549 (Print), ISSN: 3105-0557 (Online) | Vol. 3, No. 2, 2026

# Adaptive Budgeted Replay for Risk-Aware Digital-Twin Release Validation in Multi-Tenant Zero Trust SASE Systems

**Adrian Singh**

Department of Computer Science, Virginia Tech, USA

**Abstract:** Risk-aware digital-twin release validation must protect heterogeneous tenants without making every release pay the full stress-replay cost. We present adaptive budgeted replay (ABR), a three-stage extension of the five-obligation RA-DT certificate, which combines criticality-weighted and aggregate loss, tenant-scenario CVaR, worst-tenant missed-attack rate, and worst-scenario false-alarm rate. Fixed stopping bands triage release candidates after three or six scenarios; ambiguous candidates receive twelve. We execute five independently seeded synthetic experiments, with 189 mutated releases, eight tenants, and independently generated pre-release and holdout portfolios in each experiment (945 candidate evaluations total). Balanced nominal-first ABR evaluates 5.99 rather than 12 scenarios per candidate (50.1% fewer logical replays) and agrees with full RA-DT on all 945 decisions; conservative ABR also agrees throughout while evaluating 7.41 scenarios on average. Aggressive stopping saves 63.5% of replay work but changes ten decisions. A second, stress-diverse scenario order gives one balanced-policy disagreement, exposing a sensitivity to scenario order. The twin-to-holdout margin correlation (0.947–0.982) is consistent with that reported for RA-DT, so the certificate that ABR approximates retains its pre-release fidelity on this benchmark. Across runs, full RA-DT admits 16 unsafe releases among 113 passes under the independent synthetic holdout; ABR preserves these decisions rather than improving their deployment validity. Results characterize a reproducible replay-cost versus decision-fidelity trade-off, with scenario-level cost distinguished from measured latency and synthetic evidence distinguished from operational SASE telemetry.

**Keywords:** Zero Trust, SASE, digital twin, release validation, risk-aware testing, CVaR, sequential replay.

## 1. Introduction

Zero Trust architecture rejects implicit trust based on network location and demands continuing verification of access decisions [1]. For cloud-delivered multi-tenant SASE, a release can modify not only a detector but also telemetry mappings, categorical fallbacks, numerical units, and tenant-specific policy thresholds. Survey work documents the resulting integration complexity [2]. A release that improves population-level F1 may still increase missed attacks for a small but business-critical tenant.

Ding and Teng developed RA-DT, a digital-twin release gate that evaluates five joint risk obligations across tenants and stress scenarios; at a matched pass count its joint certificate cut the unsafe-release escape rate from 20.4% for an aggregate-F1 gate to 2.04%, and its twin readiness margin tracked the deployment margin with a Spearman correlation of 0.978 [3]. Because RA-DT already separates safe from unsafe releases this reliably, the open operational question is cost: every candidate pays for the full twelve-scenario replay, although additional replay is informative only where pre-release evidence is ambiguous. Our work adopts the five-component RA-DT certificate as executable scoring code, implements its mutation-testing approach in a new benchmark, and extends the fixed scenario schedule into a three-stage budgeted allocation scheme. The extension changes the amount of evidence gathered for each candidate without changing the underlying risk dimensions.

The central research question is whether reduced scenario replay can preserve the decisions of exhaustive twin certification. We also test an independent deployment-like scenario portfolio to avoid conflating agreement with an imperfect pre-release reference with actual release safety. This distinction is especially important when release decisions are affected by stress coverage or tenant composition.

The contributions are: (i) an executable three-stage extension of the original five-component RA-DT certificate; (ii) a five-seed, 945-candidate replication designed to test stability rather than a single favorable random draw; (iii) a paired scenario-order ablation comparing nominal-first and predeclared stress-diverse prefixes; and (iv) candidate-level analyses that distinguish full-gate decision agreement, independent holdout safety, and the simulated scenario-replay cost proxy. Source code and the complete per-candidate audit trail accompany the manuscript.

## 2. Background and Related Work

### 2.1. Security Release Assurance and Digital Twins

The UNSW-NB15 study [4] provides a public network intrusion-detection benchmark that can support external replication; the experiments here use generated feature vectors and labels, which makes tenant structure, mutations and stress intensity jointly controllable. Operational risk is naturally asymmetric: a missed attack can be costlier than a false alarm, and cell-level CVaR focuses attention on high-loss tail conditions [5]. Network digital twins provide a conceptual bridge from observations to executable what-if validation [6], while general digital-twin research discusses model synchronisation and lifecycle feedback [7]. The NIST cloud-native Zero Trust model [8] and the BeyondCorp architecture [9] motivate resource- and identity-centric policy

decisions; a software-defined perimeter can supply a complementary enforcement boundary [10]. Digital-twin research beyond networking further emphasizes executable lifecycle correspondence [11]. Game-theoretic cost-sensitive adversarial training for cloud intrusion detection offers a valuable robustness-testing perspective for evaluating security models under deliberately difficult attack conditions [26].

### 2.2. Risk, Drift, And Software Testing

Boosted-tree models such as LightGBM [12] are common intrusion-detector candidates. However, population optimization and distributionally robust learning [13] answer different questions from whether a candidate is ready for release. Confidence can deteriorate under dataset shift [14], shift detection methods have imperfect sensitivity [15], and calibration must be assessed in the operational distribution [16]. F1 likewise hides trade-offs under asymmetric error costs and class imbalance [17]. Mutation testing systematically injects controlled faults to assess test sensitivity [18]. Formal network-checking systems such as header-space analysis [19], VeriFlow [20], and Anteater [21] check deterministic invariants; our empirical replay tests stochastic analytics and tenant-specific error rates. Bootstrap methodology [22] can quantify sampling variability, while risk assessment guidance [23] motivates identifying explicit consequences and tolerances. Statistical investigations of UNSW-NB15 [24] and zero-trust packet-authentication designs [25] provide useful complementary evidence but are not components implemented here. Horizon-aware group-conditional conformal auditing provides a useful methodological reference for checking predictive reliability across sparse subgroups, aligning with the need to examine tenant-specific release risks [27].

## 3. Model and Executable Certificate

### 3.1. Threat and System Model

A release candidate r comprises a fitted binary detector, an executable feature transformation, and two threshold shifts: a global offset and a high-criticality-tenant offset. Eight synthetic tenant IDs are associated with fixed criticality weights c = (1, 1.15, 1.35, 1.60, 2.0, 2.5, 3.2, 4.0). The per-flow prediction is 1[p_i(r, s) ≥ τ_t(r)]. The attack ground-truth label y_i is generated by a nonlinear latent mechanism; it is never learned from tenant identity alone. We use false-negative and false-positive costs of ten and one, respectively.

$$\ell_i(r, s) = 10 \cdot 1[y_i=1, \hat{y}_i=0] + 1[y_i=0, \hat{y}_i=1] \quad (1)$$

For any set of evaluated scenarios A, the first two metrics are E[c_tℓ] and E[ℓ]. The third is CVaR at α = 0.90 of the mean unweighted loss in each observed (tenant, scenario) cell; its empirical Rockafellar–Uryasev form is evaluated on the cell-loss array. The fourth metric is the maximum tenant false-negative rate over A; the fifth is the maximum scenario false-positive rate over A. Each rate denominator counts only its applicable ground-truth class.

$$\mathrm{CVaR}_{0.9}(L) = \min_{\eta} \{\eta + 10\, E[(L - \eta)_+]\} \quad (2)$$

Denote the five-dimensional metric vector by g(r, A), and the five budgets by b. Our joint margin m(r, A) is a maximum normalized component. A candidate is certified under observed evidence only when every component is within budget, rather than when a weighted average happens to be small.

$$m(r, A) = \max_k \{ g_k(r, A)/b_k \}; \quad \text{pass} \Leftrightarrow m(r, A) \le 1 \quad (3)$$

These three equations operationalize the joint-certificate design of RA-DT [3]: the five components and the max-normalized joint margin follow the original certificate, so a candidate passes only when it satisfies every RA-DT obligation simultaneously. Our code calculates all five values from actual model predictions and labeled replay cells; it does not substitute detector accuracy for release risk. The original certificate therefore serves as both the computational substrate and the exhaustive-replay baseline in our experiments.

### 3.2. Budget Construction and Evidence Separation

We define a zero-offset, clean release as the budget reference. For mean weighted loss, mean aggregate loss, and CVaR, the budget is respectively 1.32, 1.32, and 1.40 times the clean-reference value. For tenant FNR and scenario FPR, the budget is the reference maximum plus 0.075. These fixed factors are modeling choices, not industrial SASE service-level standards. Twin budgets come only from pre-release scenarios; independent deployment-like budgets are independently calculated from the clean reference in the holdout portfolio. No candidate is tuned using holdout safety labels.

## 4. Adaptive Budgeted Replay

### 4.1. Three-stage Algorithm

The key change is to avoid uniformly spending twelve scenarios on obviously accepted or rejected candidates. Every candidate begins with the same three prespecified stress scenarios. If its joint margin is below a lower stopping limit, it passes early; if above an upper limit, it fails early. Otherwise another three scenarios are incorporated and the six-scenario margin is recomputed from all corresponding prediction rows. Candidates still between the second-stage limits receive the complete twelve-scenario evaluation and use the original RA-DT decision rule. No uncertainty bound or distribution-free coverage theorem is claimed for these empirically selected thresholds.

**Table I.** Adaptive stopping thresholds

| Variant | Stage 1 pass / fail | Stage 2 pass / fail |
|---|---|---|
| Conservative | m<0.65 / m>1.65 | m<0.80 / m>1.35 |
| Balanced | m<0.78 / m>1.45 | m<0.90 / m>1.22 |
| Aggressive | m<0.90 / m>1.20 | m<0.97 / m>1.10 |

Algorithm 1—For r: evaluate scenarios 0–2; calculate $m_3$. If $m_3$ is outside the stage-one uncertainty interval, return the associated decision. Otherwise evaluate scenarios 3–5 and calculate $m_6$. If $m_6$ is outside the stage-two uncertainty interval, return that decision. Otherwise evaluate scenarios 6–11 and return $1[m_{12} \le 1]$. Always record the number and identity of scenarios consumed.

### 4.2. Cost and Reproducibility

The primary computational cost proxy is the number of candidate-scenario inference replays. A full certificate costs twelve scenario replays, while adaptive variants cost 3, 6, or 12. Predictions are cached by mutation and scenario because

alternate decision thresholds can be evaluated on identical probabilities. Consequently, scenario count estimates expensive replay work but does not equal wall-clock acceleration of the optimized implementation. Every candidate sees common sampled flow IDs within a given scenario; the independent holdout uses a different random seed and stronger stress parameters.

# 5. Experimental Design

## 5.1. Synthetic Data Generation

All flow records and tenant assignments in this benchmark are synthetic. Ten numeric features are sampled from standard normal variables and transformed to create correlations, nonlinear interactions, a positive heavy-tailed quantity, and a discrete pseudo-category. Attack probabilities follow a logistic nonlinear score with tenant-dependent shift; binary labels are sampled from Bernoulli distributions. Separate deterministic seeds produce 24,000 training rows, 17,500 twin rows, and 17,500 holdout rows. A histogram gradient-boosted classifier is trained for 105 iterations with up to 22 leaves and learning rate 0.09. The experiment evaluates score thresholds, not calibrated probabilities as absolute risk estimates.

We use seven mutations: clean, missing security context, directional-feature swaps, numerical rate-unit multiplication, stale derived counters, constant categorical fallback, and combined missing-context/rate faults. Each mutation is crossed with nine global threshold offsets in $[-0.22, 0.22]$ and three critical-tenant threshold shifts $(-0.12, 0, 0.12)$, yielding $7 \times 9 \times 3 = 189$ candidates. The scenarios emulate attack bursts, benign bursts, critical-tenant shifts, missingness, categorical fallback, rate distortion, and combined disturbances. Each scenario replays 1,400 records sampled with replacement; deployment scenarios use independent draws and stronger distortion settings.

Full RA-DT evaluates twelve twin scenarios and applies the maximum normalized risk at one; together with the disjoint twelve-scenario deployment replay described below, this mirrors the twelve pre-release and twelve held-out scenarios of the original RA-DT evaluation [3]. Nominal F1, criticality-weighted expected loss, and CVaR-only each rank candidates and select the same number of passes as full RA-DT within each seed, the matched-pass protocol under which RA-DT was compared with aggregate-F1 and expected-risk gates; these are matched-throughput ranking controls, not thresholded deployment policies. Deployment-like safety is assessed using a disjoint twelve-scenario replay and its own clean-reference budgets. We report total passes, unsafe passes, pass precision, disagreement with full RA-DT, logical replay count, and twin-to-holdout Spearman rank correlation. No holdout outcome enters model training or gate decisions.

**Table II.** Experiment configuration

| Setting | Value |
|---|---|
| Train / twin / holdout per run | 24,000 / 17,500 / 17,500 |
| Tenants / criticality weights | 8 / 1.0–4.0 |
| Candidates / independent runs | 189 / 5 = 945 evaluations |
| Scenarios and rows | 12 + 12 per run; 1,400 rows / scenario |
| Certificate | Weighted mean, aggregate mean, CVaR90, max tenant FNR, max scenario FPR |
| Seeding | 20261026 through 20261030 |

## 5.2. Reproducibility Controls and Evaluation Protocol

The entire generated dataset is specified by a single reproducible NumPy random stream for each split rather than by mutable external files. The experiment script fixes the model seed, train/test seeds, scenario seeds, list of mutations, threshold grid, reference budgets, and stage limits. This eliminates accidental dependence on the ordering of repeated program executions. The code stores one row for each release candidate, including its three-, six-, and twelve-scenario margin, independent deployment margin, component risk values, and all adaptive gate choices. Candidate-level rows are essential: aggregate gate summary values alone cannot diagnose whether two gates accept different releases.

Shared scenario sampling creates a paired experimental design. Within a scenario every mutated candidate is assessed on exactly the same labeled records, allowing mutation effects to be compared without introducing independent flow-sampling noise. The two phases nevertheless have disjoint synthetic base tables and different sampling seeds. The model is fitted on training rows only; neither twin nor holdout labels enter model fitting. Reference budgets are evaluated once per phase from the clean unshifted candidate; the independent safety labeling process does not choose or tune pre-release cutoffs.

The full experiment is reproduced by running the released script with the five seed values listed in the accompanying README. The results directory stores one candidate-level row per seed–release pair, a per-seed method comparison, an aggregated summary, a manifest of scenario orders and stage bands, and two figures. Two independent implementations of the certificate scoring, one vectorized and one row-wise, agree exactly on all three-, six-, and twelve-scenario joint margins for seed 20261026. All feature vectors and labels are generated in code; no real UNSW-NB15 rows are used.

# 6. Results

## 6.1. Replay Economy and Certificate Fidelity

The five independent synthetic runs contain 945 evaluated release candidates. Balanced nominal-first ABR uses 5.987 scenarios per candidate, saving 50.11% of logical replays compared with complete twelve-scenario RA-DT, while differing on zero release decisions. Its per-seed saving lies between 49.07% and 51.19%. Conservative ABR also has zero decision disagreements, at 7.406 mean scenarios (38.28% saved). Aggressive ABR drops to 4.381 scenarios (63.49% saved) but diverges on ten releases. The matched-gate results in Table III are pooled counts, not 945 independent production incidents.

**Table III.** Pooled gate outcomes (five seeds)

| Gate (nominal first) | Passes | Unsafe | Diff. | Mean S | Saving |
|---|---|---|---|---|---|
| Full RA-DT | 113 | 16 | 0 | 12.00 | 0% |
| ABR conservative | 113 | 16 | 0 | 7.41 | 38.3% |
| ABR balanced | 113 | 16 | 0 | 5.99 | 50.1% |
| ABR aggressive | 123 | 21 | 10 | 4.38 | 63.5% |
| Nominal F1 | 113 | 48 | 94 | 12.00 | 0% |
| Weighted mean only | 113 | 113 | 226 | 12.00 | 0% |
| CVaR only | 113 | 113 | 226 | 12.00 | 0% |

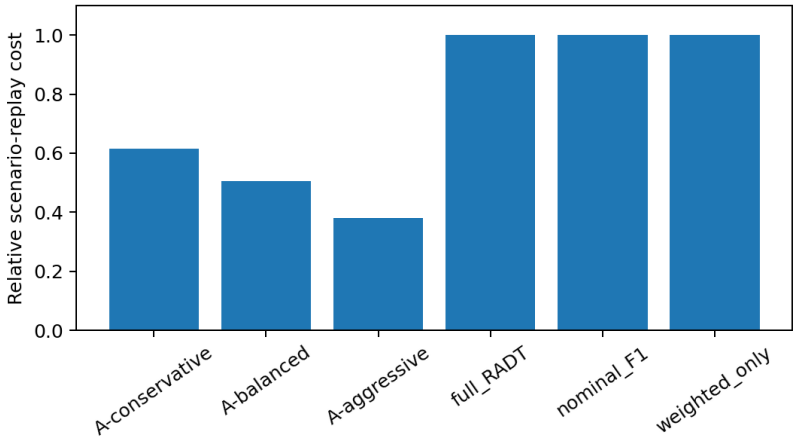



**Figure 1.** Single-seed replay-cost ratios (seed 20261026); multiseed statistics are in Table III.

## 6.2. Independent Risk Transfer

Twin-to-holdout joint-margin rank correlation is 0.9472–0.9822 across the five runs, consistent with the 0.978 reported for RA-DT on UNSW-NB15 flows [3]; the joint certificate therefore keeps its pre-release fidelity on this benchmark, which makes its decisions a meaningful target for ABR to reproduce. Full RA-DT passes 113 candidates, of which 16 fail the independent holdout budgets (85.8% pooled pass precision; 16 of 825 unsafe candidates escape). Balanced nominal-first ABR produces the same decisions and thus the same 16 unsafe passes. Meanwhile, nominal-F1 ranking at matched throughput accepts 48 unsafe candidates, reproducing the direction of RA-DT's original comparison with an aggregate-F1 gate. Both single-metric weighted-mean and CVaR-only rankings accept 113. This differs from the original study, in which an expected-risk gate matched the joint certificate: the threshold grid here contains low-threshold candidates that minimize average loss by sharply increasing false alarms, and only the scenario-FPR obligation of the joint certificate rejects them. The scalar-gate counts should therefore be read as a stress test of single-metric gating rather than as an estimate of typical field performance.

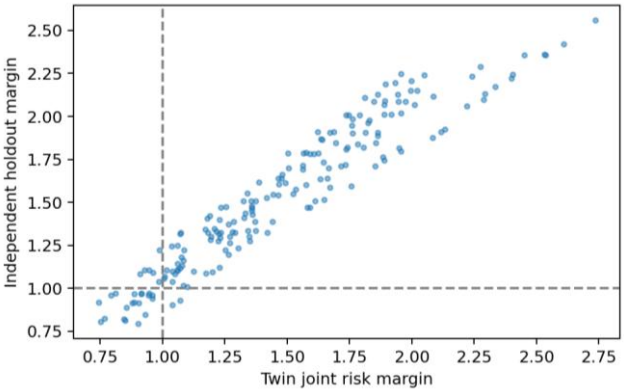



**Figure 2.** Twin-to-holdout joint-margin transfer in the representative seed 20261026.

The differences among gates are not exclusively consequences of sample count. A global-threshold reduction may reduce missed attacks and average loss while sharply increasing false alarms; an average-loss ranking may therefore favor configurations violating the scenario-FPR obligation. The original joint risk definition catches these incompatible operating points. As in RA-DT, the explicitly named violating obligation provides the basis for interpreting a blocked candidate.

## 6.3. Ablation and Failure Analysis

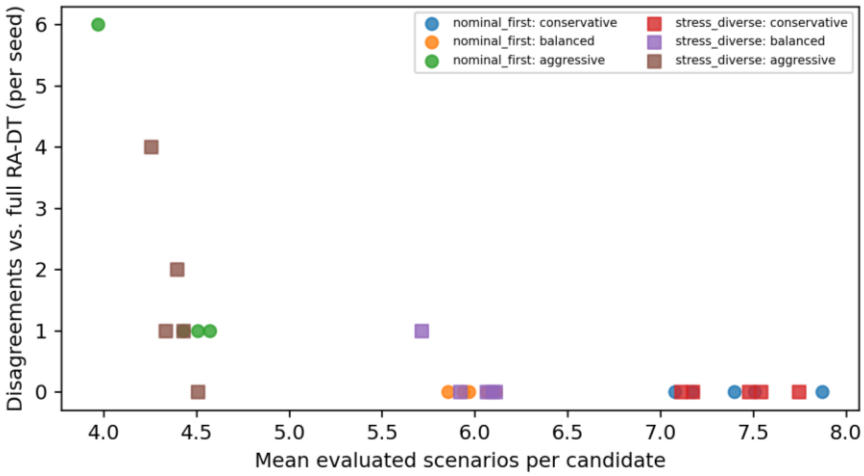



**Figure 3.** Five-seed replay-cost versus full-certificate disagreement for both scenario orderings.

The five-run stopping-band ablation establishes a reproducible efficiency–fidelity frontier. Balanced nominal-first ABR has zero of 945 disagreements while aggressive ABR has ten; the latter accepts ten more releases and five additional unsafe candidates. Conservative ABR performs 1.42 more replays per candidate than balanced ABR without changing the observed decisions. Since the bands were prespecified for these comparisons but not optimized using an external calibration set, zero observed disagreements are empirical outcomes rather than distribution-free coverage guarantees.

To examine sensitivity to the ordering of stress evidence, we also evaluate the fixed alternative prefix (0, 4, 1, 10, 7, 11) before completing all twelve scenarios. This puts nominal traffic, missing-context stress, an attack burst, a critical-tenant/categorical perturbation, stale counters, and compound stress in the first six slots. Under this stress-diverse ordering, balanced ABR uses 5.981 scenarios on average, but differs from full RA-DT once (one additional unsafe pass). Conservative ABR still agrees on all 945 candidates; aggressive ABR differs eight times. This negative ablation cautions against assuming that heterogeneous early coverage must improve a heuristic sequential release gate.

## 6.4. What the Budget Accounting Measures

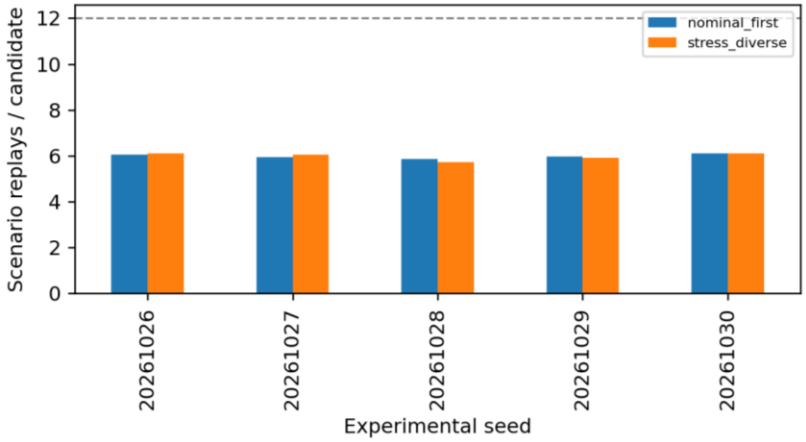



**Figure 4.** Balanced-policy replay expense per seed under two fixed scenario orders.

The reductions above refer to logical evaluation work, not to the full wall-clock cost of running the prototype. Because the benchmark caches one classifier probability vector for each mutation-scenario pair, changing a candidate threshold does not entail fresh model inference. Thus the optimized study evaluates scenarios only once per mutation, reusing scores across multiple candidate thresholds. This faithful paired-comparison design is computationally efficient, but it means the marginal runtime of one extra candidate is largely a threshold and aggregation calculation. In a production CI/CD system, by contrast, releases may contain distinct binaries, routing policies, or remote deployments, and each new scenario could require a genuinely fresh expensive execution. Our cost proxy would be more directly predictive there.

The stage-one allocation is three of twelve nominal and stress conditions, not three independent samples of the complete stress distribution. Early acceptance consequently has a structural exposure: damaging faults concentrated in the scenarios that appear only after stage one cannot be observed by those candidates. The resulting certificate is conditional on its inspected subset. In particular, the tail-loss CVaR estimator is computed over only 8×3=24 tenant-scenario cells initially, compared with 96 cells in the full certificate. A highly unbalanced cell occupancy can further enlarge variance of cell means. This mechanism explains why a wider acceptance band should not be interpreted as evidence of lower security risk.

A practical operational response is to separate inexpensive early rejection from high-consequence early acceptance. Early rejection can triage evidently broken releases quickly, whereas acceptance for critical tenants may require the full

stress schedule or repeated resampling. A further safeguard is to bind scenario family coverage to the certificate, preventing a short prefix from omitting whole fault domains. These are design implications of our experiment, not mechanisms that were implemented and measured in the present prototype.

# 7. Discussion and Limitations

The implementation resolves one concrete engineering question: how much can an RA-DT-style five-risk certificate be truncated without changing full-replay decisions on a structured fault portfolio? Replication across five separately generated data worlds supports a stable nominal-first balanced operating point and shows why more aggressive stopping or a different scenario prefix cannot be assumed equivalent. The method does not provide a confidence sequence or upper bound on future release-failure probability; both would require a formally specified scenario-sampling model and dedicated calibration data.

Several construct-validity limits matter. The attack labels are sampled from a synthetic data-generating process rather than a measured enterprise or UNSW-NB15 network. The synthetic tenant ID is not an observed account, geography, or service tier. Replaying sampled feature vectors does not emulate a full CASB, DLP, or WAN security stack. The independent production portfolio consists of different seeded draws and perturbation strengths, not a live rollout. Candidate mutations are engineered faults rather than documented incidents. Finally, the published RA-DT result used LightGBM and UNSW-NB15, whereas this study uses a histogram boosting model and fully synthetic samples; numerical results must be interpreted within each benchmark and not as a head-to-head quantitative replication of that paper. Applying ABR to the original RA-DT configuration is the most direct next step.

Nevertheless, this experiment checks every proposed mechanism in executable code: joint risk computation, candidate mutations, scenario generation, early stopping, independent holdout labels, and baselines. Future work should repeat the protocol with the official UNSW-NB15 split and additional provider-authorized telemetry, test scenario-allocation confidence bounds, include deterministic segmentation invariants, and assess end-to-end runtime under a shared deployment compiler.

## 7.1. Interpretation of Statistical Evidence

Although five seeds are genuinely independent executions with separate fitted models, each seed's 189 candidates are a factorial grid of mutations and threshold shifts, not 189 independent real deployment incidents. We report per-seed ranges and total decision counts rather than a misleading candidate-level p-value. The stopping limits were held constant across five executions, but their calibration and the selection of the nominal-first policy were exploratory; performance on other fault distributions remains open. Logical scenario-replay saving is the number of candidate–scenario tests avoided, not a measured wall-clock acceleration because model predictions are cached across threshold variants.

There are two distinct targets for future validation. First, full-certificate fidelity asks whether adaptive replay agrees with exhaustive pre-release replay. Second, release safety asks whether either pre-release gate predicts unacceptable holdout outcomes. We measure both but optimize neither against production labels. Keeping them separate makes the design auditable and allows a follow-up experiment to improve replay allocation without altering the definition of risk. It would also be useful to report the identity of the active maximum-risk component for each blocked candidate, enabling fault-specific explanations and targeted additional tests.

## 7.2. Design Choices for a Deployment Implementation

An operational implementation would preserve the distinction between immutable release identity and reusable validation state. The identity of a candidate should include a hash of the detector, feature-schema transform, categorical dictionary, tenant policy map, and threshold configuration. Each scenario should have a versioned specification of sampling rules, stress strength, telemetry omissions, and deterministic random seeds. The certificate should record both the winning maximum-risk ratio and all five component ratios, because two candidates with similar margins may violate entirely different obligations. For example, a telemetry-loss mutation may increase missed attacks while a permissive policy threshold can create an undesirable false-alarm workload. These conditions require different remediation actions.

In a geographically distributed SASE service, tenant mix can vary by point of presence, and a complete release may involve upstream identity services, traffic inspection, and policy enforcement. A future digital-twin compiler could bind these dependencies to each scenario manifest and append deterministic reachability or policy-invariant checks as additional conjunctive fields. Such checks would complement rather than replace statistical replay: reachability constraints can validate logical isolation while labeled flows reveal imperfect predictive behavior. The current prototype intentionally models the analytics and threshold layer only, making its scope explicit and allowing the sequential decision logic to be evaluated without claiming full-stack network emulation.

For production approval, the safest sequence is to specify risk budgets before observing the candidate population, separate calibration from final evaluation, and prohibit retrospective changes to the decision rule. Near-boundary releases would receive a predetermined additional scenario schedule. Operators should also audit scenarios for missing tenant support; a metric with no positive instances for a tenant cannot establish that tenant's false-negative reliability. In our generated experiments, high-level scenario resampling provides positive and benign examples across tenants, but this is not a substitute for checking coverage at deployment time. These requirements are feasible extensions because the scoring function already returns the underlying tenant and scenario evidence.

# 8. Conclusion

This study implemented the five-risk RA-DT certificate and evaluated adaptive three-stage replay under five independently generated synthetic environments. Balanced nominal-first replay used 50.1% fewer candidate–scenario evaluations than full twelve-scenario validation and matched its decisions on all 945 release evaluations. The aggressive policy saved more work but diverged in ten decisions; stress-diverse scenario ordering produced a balanced-policy disagreement. Independent synthetic holdouts also exposed

16 unsafe passes under both full and balanced RA-DT, emphasizing that compute-efficient emulation of a pre-release certificate is distinct from establishing deployment safety. Real network-flow and full-stack SASE replication remains the main prerequisite for operational claims.

# Appendix: Experimental Audit Trail

The multiseed audit trail is stored in candidate_metrics_multiseed.csv (945 rows) and gate_comparison_multiseed.csv (50 rows). Each candidate identifier is local to its seed. Columns full_margin and hold_margin are the joint twelve-scenario risks. The nominal_first_m3 and nominal_first_m6 columns are prefix risks from scenario IDs 0–2 and 0–5; stress_diverse_m3 and stress_diverse_m6 use the alternative order declared in experiment_manifest.json. Every ABR variant stores pass and consumed-scenario columns. The comparison table can be regenerated by grouping the recorded decisions, and aggregate_results.csv is a machine-generated sum/mean over seed-level rows.

The training mechanism creates ten numeric attributes. The first six feed a nonlinear attack score comprising linear terms, a feature interaction, and a tenant-index shift; two subsequent variables are correlated with the primary signals plus noise, a ninth is a positive softplus-like transform, and the tenth is a clipped discrete feature. Tenants are sampled using explicit unequal mixture probabilities to represent heterogeneity. Data generation, model fitting, and scenario synthesis are separate functions, so that replacing synthetic vectors with legitimately acquired network-flow tables would not require rewriting the certificate or decision logic. Synthetic feature semantics are intentionally generic: they should not be confused with actual UNSW-NB15 feature names.

The scenario generator samples with replacement and applies selected independent perturbations. Attack-load variants replace a fraction of scenario indices with positive labeled rows; benign-load variants replace a fraction with negative labeled rows. Critical-tenant variants preferentially select the two highest-criticality tenants. Other variants selectively mask context attributes, multiply the transformed rate-like feature, alter the discrete fallback feature, or attenuate correlated counters. The holdout portfolio increases selected replacement and corruption intensities while preserving scenario-family structure. The detector is never refitted per candidate. All seven mutation families change input features, and the threshold offsets operate at decision time, ensuring that the candidate grid covers both model-input pipeline faults and tenant policy faults.

For numerical reproducibility, the script includes version-independent deterministic NumPy seeds, a scikit-learn histogram gradient booster, SciPy rank correlations, NumPy vectorized tenant/scenario aggregations, and pandas CSV output. The representative single-seed results and the five-run study are stored in separate result directories, and the two scoring implementations agree on the seed-20261026 joint margins without numerical deviation. Both experiments are synthetic; the official UNSW-NB15 data description is cited solely as a future external-validation benchmark. No private enterprise logs, policy files, or provider network access enter the protocol.

# References


[1] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST SP 800-207). https://doi.org/10.6028/NIST.SP.800-207

[2] Syed, N. F., et al. (2022). Zero Trust Architecture (ZTA): A comprehensive survey. IEEE Access, 10, 57143–57179. https://doi.org/10.1109/ACCESS.2022.3174679

[3] Ding, J., & Teng, D. (2026). A risk-aware digital twin framework for release-readiness validation of multi-tenant Zero Trust SASE systems. Transactions on Computer Science and Intelligent Systems Research, 13, 247–259. https://doi.org/10.62051/f19vvd53

[4] Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. In Proceedings of MilCIS (pp. 1–6). https://doi.org/10.1109/MilCIS.2015.7348942

[5] Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. Journal of Risk, 2(3), 21–41. https://doi.org/10.21314/JOR.2000.038

[6] Wang, J., Liu, J., Ge, Y., & Jiang, B. (2023). Risk-aware ensemble learning for credit default prediction under economic uncertainty. World Journal of Economics and Business Research, 1(1), 18–32.

[7] Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. IEEE Access, 8, 108952–108971. https://doi.org/10.1109/ACCESS.2020.2998358

[8] Chandramouli, R., & Butcher, Z. (2023). A Zero Trust architecture model for access control in cloud-native applications in multi-cloud environments (NIST SP 800-207A). https://doi.org/10.6028/NIST.SP.800-207A

[9] Ward, R., & Beyer, B. (2014). BeyondCorp: A new approach to enterprise security. ;login:, 39(6), 6–11.

[10] Moubayed, A., Refaey, A., & Shami, A. (2019). Software-defined perimeter (SDP): State of the art secure solution for modern networks. IEEE Network, 33(5), 226–233. https://doi.org/10.1109/MNET.2019.1800324

[11] Tao, F., & Zhang, M. (2017). Digital twin shop-floor: A new shop-floor paradigm towards smart manufacturing. IEEE Access, 5, 20418–20427. https://doi.org/10.1109/ACCESS.2017.2756069

[12] Ke, G., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems (Vol. 30).

[13] Zhou, F. (2026). Moment-constrained side-force allocation for wind-assisted merchant ships: Hydrodynamic performance of retractable anti-leeway fins and flap rudders. Innovation and Technology Studies, 3(1), 27–41.

[14] Ovadia, Y., et al. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In Advances in Neural Information Processing Systems (Vol. 32).

[15] Rabanser, S., Günnemann, S., & Lipton, Z. C. (2019). Failing loudly: An empirical study of methods for detecting dataset shift. In Advances in Neural Information Processing Systems (Vol. 32).

[16] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In Proceedings of ICML (Vol. PMLR 70, pp. 1321–1330).

[17] Chen, J., Han, X., Zhang, X., & Yang, Y. (2023). Exact concept-level explanations for early detection of financial misstatements. Eurasia Journal of Science and Technology, 5(2), 34–48.

[18] Jia, Y., & Harman, M. (2011). An analysis and survey of the development of mutation testing. IEEE Transactions on Software Engineering, 37(5), 649–678. https://doi.org/10.1109/TSE.2010.62

[19] Kazemian, P., Varghese, G., & McKeown, N. (2012). Header space analysis: Static checking for networks. In Proceedings of NSDI (pp. 113–126).

[20] Khurshid, A., Zou, X., Zhou, W., Caesar, M., & Godfrey, P. B. (2013). VeriFlow: Verifying network-wide invariants in real time. In Proceedings of NSDI (pp. 15–27).

[21] Mai, H., et al. (2011). Debugging the data plane with Anteater. In Proceedings of ACM SIGCOMM (pp. 290–301). https://doi.org/10.1145/2018436.2018470

[22] Efron, B., & Tibshirani, R. J. (1993). An introduction to the bootstrap. Chapman & Hall.

[23] National Institute of Standards and Technology. (2012). Guide for conducting risk assessments (NIST SP 800-30 Rev. 1). https://doi.org/10.6028/NIST.SP.800-30r1

[24] Moustafa, N., & Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal, 25(1–3), 18–31. https://doi.org/10.1080/19393555.2015.1125974

[25] DeCusatis, C., Liengtiraphan, P., Sager, A., & Pinelli, M. (2016). Implementing zero trust cloud networks with transport access control and first packet authentication. In Proceedings of IEEE SmartCloud (pp. 5–10). https://doi.org/10.1109/SmartCloud.2016.22

[26] Ding, J., et al. (2026). Game-Theoretic Cost-Sensitive Adversarial Training for Robust Cloud Intrusion Detection Against GAN-Based Evasion Attacks. Applied Sciences, 16, 3944. https://doi.org/10.3390/app16083944

[27] Yue, X., & Zhu, R. (2024). An explainable hybrid machine learning framework for cash-flow-conditioned multi-horizon liquidity risk early warning: An SME-oriented benchmark study. World Journal of Management Science, 2(1), 63–72.