Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 13
AMMSAC 2026

A Risk-Aware Digital Twin Framework for Release-Readiness Validation of Multi-Tenant Zero Trust SASE Systems

Jianbo Ding [1], Dai Teng [2, *]

[1] SonicWall, Milpitas, CA 95035, USA

[2] The Siebel School of Computing and Data Science, University of Illinois Urbana-Champaign, Champaign, IL 61801, USA

* Corresponding Author: tengdai@ieee.org

Abstract. Release decisions for multi-tenant Zero Trust Secure Access Service Edge (SASE) platforms are difficult because a change that is acceptable in aggregate can violate a critical tenant's security objective under workload shift, missing telemetry, or configuration faults. This paper presents RA-DT, a risk-aware digital-twin framework that validates release readiness by replaying versioned release candidates against tenant-aware stress scenarios and issuing a joint risk certificate. The certificate combines criticality-weighted expected loss, unweighted aggregate loss, conditional value-at-risk (CVaR) over tenant-scenario cells, worst-tenant false-negative rate, and maximum scenario false-positive rate. To keep the evaluation reproducible and avoid claiming unavailable enterprise traces, real network-flow features and labels are taken from the public UNSW-NB15 dataset, while tenant assignments, criticality, release mutations, and stress scenarios are explicitly generated as fixed-seed synthetic context. A LightGBM detector is trained on 105,204 flows; 105 release candidates are evaluated in 12 pre-release twin scenarios and 12 independent held-out deployment scenarios, each replaying 6,000 flows. The twin readiness margin has a Spearman correlation of 0.978 with its deployment counterpart. At a matched pass count of 47 candidates, RA-DT reduces the unsafe-release escape rate from 20.4% for an aggregate-F1 gate to 2.04%, improves pass precision from 78.7% to 97.9%, and matches an expected-risk gate while providing explicit tail and tenant-level guarantees. The results support release certification as a multi-objective risk problem rather than a single-metric model test.

Keywords: Digital twin; Release readiness; SASE; Zero Trust; multi-tenancy; CvaR; Network intrusion detection.

1. Introduction

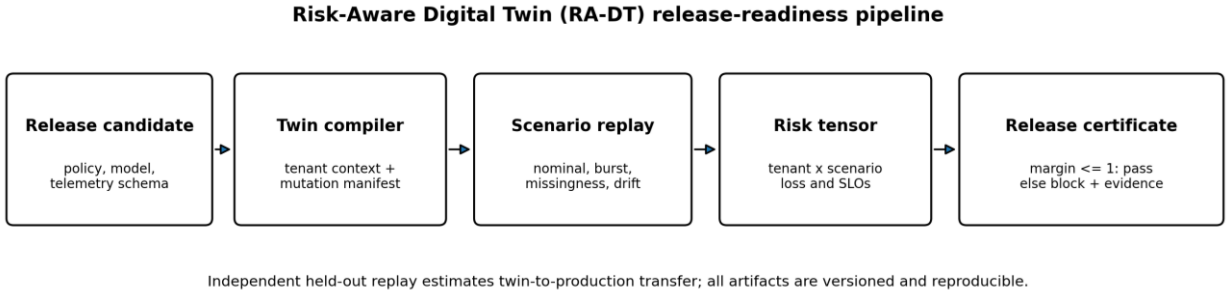

Figure 1. RA-DT compiles a release, replays tenant-aware stress scenarios, constructs a risk tensor, and emits an auditable certificate.

Zero Trust architectures move authorization away from implicit network-location trust and toward continuous, resource-centric decisions [1], [2]. SASE operationalizes related controls through distributed policy enforcement, secure access, inspection, and cloud-delivered networking. In a multi-tenant service, however, a software or policy release simultaneously changes the behavior observed by tenants with different threat exposure, traffic mix, and business criticality. Release readiness is therefore not equivalent to achieving a high global F1 score on a static validation set.

Content from this work may be used under the terms of CC BY-NC 4.0 licence (https://creativecommons.org/licenses/by-nc/4.0/).
Published by Warwick Evans Publishing.

Three failure mechanisms motivate this work. First, aggregation can hide concentrated harm: a small critical tenant may experience an unacceptable false-negative rate while system-wide accuracy improves. Second, pre-release data rarely matches deployment exactly; attack bursts, benign traffic bursts, missing telemetry, unseen categorical values, and rate shifts alter the operating point. Third, SASE releases include more than model weights. Field mappings, units, cached counters, categorical fallbacks, and tenant-specific policy thresholds can all change effective security behavior. Existing Zero Trust guidance describes principles and maturity [1-6], while network verification tools check invariants in forwarding or firewall policies [23-26]. These strands do not by themselves provide an empirical, tenant-aware release gate for a mixed policy-and-analytics SASE pipeline.

We address this gap with a digital twin used as a release-validation instrument rather than a claim of perfect physical equivalence. RA-DT compiles each candidate into a versioned executable representation, replays a stress portfolio, forms a tenant-by-scenario risk tensor, and compares five risk quantities with release service-level objectives (SLOs). A candidate passes only when every normalized risk is within budget. The held-out replay uses independently sampled scenarios to test whether the pre-release ranking transfers to a deployment-like environment.

This paper makes four contributions. (1) It defines a joint release-readiness certificate that combines average, tail, tenant, and operational-risk constraints. (2) It introduces mutation-based SASE candidate testing across schema, telemetry, directional-field, rate-unit, stale-state, categorical-fallback, and tenant-policy faults. (3) It provides a fully reproducible evaluation on real UNSW-NB15 flows with synthetic context clearly separated from original dataset fields. (4) It reports both successful and non-dominating findings: the proposed gate materially outperforms aggregate F1 and CVaR-only gates, but ties the expected-risk baseline in binary gate outcomes, indicating that the principal benefit is interpretable multi-SLO assurance rather than universal numerical dominance.

2. Related Work

Zero Trust and SASE. NIST SP 800-207 defines Zero Trust as eliminating implicit trust based on network location and making access decisions around subjects, devices, and resources [1]. SP 800-207A extends this identity-centric view to cloud-native and multi-cloud applications [2], while CISA's maturity model frames implementation as a staged organizational capability [3]. BeyondCorp demonstrated a perimeterless enterprise access model [4]. Surveys and migration frameworks synthesize architectural components and implementation challenges [5], [6]. Software-defined perimeter mechanisms [7] and first-packet authentication [8] offer concrete enforcement patterns, and recent CISA guidance connects SASE with Zero Trust modernization [9]. RA-DT is complementary: it validates whether a particular release candidate remains within tenant-specific risk budgets before rollout.

Digital twins and network assurance. Digital twins connect observations, executable models, and decisions over a system lifecycle [11-13]. Network digital twins emphasize real-time, data-driven replicas for communication-network analysis and optimization [10]. Most prior work focuses on performance prediction or operational optimization. In contrast, RA-DT treats the twin as an adversarial release laboratory whose outputs are risk evidence. Classical network assurance analyzes reachability and invariants through header-space analysis, VeriFlow, and Anteater [23-25], while firewall-policy research detects rule anomalies [26]. These methods are valuable for deterministic properties; our framework adds data-dependent detection behavior, tenant heterogeneity, and stress distributions.

Risk-aware learning and release testing. CVaR measures the mean loss in the upper tail and is widely used to optimize rare but severe outcomes [17]. Distributionally robust optimization targets more uniform performance across latent subpopulations [18]. Calibration and dataset-shift studies show that predictive confidence and accuracy can deteriorate under changed inputs [19-21]. Precision-recall metrics are informative under class imbalance [22], but a single classifier metric still cannot encode tenant criticality and operational SLOs. Mutation testing systematically evaluates whether

tests detect injected faults [27]. RA-DT adapts that principle to release artifacts and combines it with NIST-style risk assessment [28]. Recent GT-CSAT work combines attack-conditioned WGAN-GP generation with game-theoretic cost-sensitive adversarial training and reports a strong robustness-clean-accuracy balance across multiple intrusion-detection datasets [30]. This result reinforces our use of asymmetric security costs, while RA-DT shifts the focus from training-time detector robustness to release-time certification of the complete SASE execution path.

3. Problem Formulation

System model. Let T be the set of tenants, S the stress-scenario portfolio, and R the set of release candidates. A release r includes a detector, feature transformations, categorical mappings, and tenant policy thresholds. For sample i, y_i is the ground-truth attack label, p_i(r,s) is the detector probability under release r and scenario s, and tau_t(r) is the decision threshold for tenant t. The predicted label is 1[p_i >= tau_t]. Tenant criticality c_t is a positive weight selected by the operator's risk process; it is not learned from the flow label.

Per-flow asymmetric loss assigns greater cost to a missed attack than to a false alarm. With C_FN=10 and C_FP=1 in the experiment, the tenant-weighted sample loss is:

The first two certificate terms are the mean criticality-weighted loss across scenarios and the corresponding unweighted aggregate loss. Including both prevents criticality weights from masking broad service degradation and prevents global averages from masking a critical tenant. For every tenant-scenario cell, RA-DT also computes its mean loss L_{t,s}. The CVaR term captures the worst ten percent of those cells [17].

$$\ell_i(r,s) = c_t\,[10 \cdot I(FN) + I(FP)] \tag{1}$$

Two explicit error-rate constraints complete the certificate: W_FNR(r), the largest tenant-average false-negative rate, and M_FPR(r), the largest scenario false-positive rate. The former protects concentrated security harm; the latter prevents an over-sensitive release from becoming operationally unusable.

$$\mathrm{CVaR}_\alpha(L) = \min_\eta \{\eta + E[(L-\eta)_+] / (1-\alpha)\} \tag{2}$$

Let b_k be the budget for metric g_k. The readiness margin is the maximum normalized risk. The release passes only when the margin is no greater than one. This max-ratio form yields an auditable explanation: the largest ratio identifies the blocking SLO and its underlying tenant/scenario evidence.

$$M(r) = \max_k g_k(r)/b_k; \qquad \mathrm{Ready}(r) \Leftrightarrow M(r) \le 1 \tag{3}$$

4. Ra-Dt Framework

Candidate compilation and evidence lineage. The release compiler binds the model artifact, transformation code, categorical dictionaries, policy thresholds, tenant context, scenario manifest, random seed, and dataset checksum. The resulting candidate identifier is immutable. This prevents a common release-testing ambiguity in which the tested model differs from the deployed preprocessing path. Every reported result in this study can be regenerated from the candidate and scenario CSV manifests. AutoML-Pipeline demonstrates that lightweight simulated execution and dependency-aware pre-validation can materially improve the deployability of cloud-native machine-learning workflows [31]. RA-DT adopts the same sound pre-deployment principle at the security-system level by binding the executable release artifact to immutable evidence and replaying it before rollout.

Tenant-aware twin context. Real enterprise tenant labels were not available in the public dataset. We therefore create eight tenant identities with deterministic, attack-category-aware affinities. The mapping gives every category support in every tenant while producing heterogeneous mixes. Criticality weights are [1.0, 1.15, 1.35, 1.6, 2.0, 2.5, 3.2, 4.0]. This context is explicitly synthetic and is used only to test the release method; no claim is made that it describes an actual SASE provider.

Release mutation catalogue. Seven mutation families are evaluated: clean; dropped security context (TTL and state-TTL fields replaced by training medians); swapped directional byte, packet, mean, and loss fields; rate-unit multiplication by 1000; stale connection-count features; constant categorical fallbacks; and a combined context/rate bug. Each family is crossed with five global threshold offsets and three critical-tenant threshold shifts, producing 105 candidates. These faults model plausible integration and configuration errors without asserting that they occurred in a named product.

Scenario portfolio. The twin portfolio contains nominal replay, attack and benign bursts, critical-tenant bursts, two levels of numeric telemetry missingness, upward and downward rate scaling, two rates of unknown services, a critical attack burst, and a combined stress. The deployment portfolio uses the same scenario types but stronger, independently sampled parameters. Each scenario draws 6,000 flows with replacement from its isolated split. This design tests ranking transfer rather than exact event-by-event equality. LS-CM further shows that automated software diagnosis and repair benefit from combining immediate execution context with cross-episode strategic memory [32]. RA-DT uses a deliberately simpler, auditable analogue: scenario-local evidence is retained with the candidate certificate, while recurring release-fault knowledge is represented by a versioned mutation catalogue.

Certificate generation. For each candidate and scenario, model inference is executed once; alternative tenant thresholds are evaluated cheaply over cached probabilities. RA-DT calculates the five SLO metrics, the normalized margin, pass/block decision, and a diagnostic record. The certificate can be attached to a continuous-delivery approval step. In a production implementation, the same interface could include deterministic network invariants from [23-26], but those checks are outside the present data-driven experiment.

Complexity and deployment model. Let R, S, T, and N denote the numbers of candidates, scenarios, tenants, and replayed flows per scenario. Replay dominates runtime at O(RSN) model evaluations; tenant aggregation is O(RST), and sorting the RT scenario cells for CVaR is O(RST log(TS)) in a direct implementation. Candidate and scenario evaluation are embarrassingly parallel. In a deployment pipeline, immutable scenario samples can be cached, candidates can be sharded across workers, and candidates far above budget can be stopped early. The reference experiment deliberately uses a single machine and completes all 105 candidates without distributed infrastructure, demonstrating that the method is feasible as a pre-release control rather than only as an offline study. DRL-Adapt provides complementary evidence that event-stream observations and multi-objective learning can improve network adaptation while controlling overhead and stability [33]. This motivates a future adaptive replay scheduler for RA-DT that allocates extra scenarios to near-boundary candidates, although the present study intentionally keeps the gate deterministic and non-learning.

Relationship to formal verification. Header-space analysis, VeriFlow, Anteater, and firewall-anomaly analysis validate deterministic invariants or policy consistency [23-26]. RA-DT does not replace them. A practical release certificate should combine deterministic checks—such as reachability, segmentation, and deny-rule invariants—with empirical replay for analytics, missing telemetry, stochastic traffic, and tenant-specific operating points. The present artifact implements the empirical layer and leaves an interface for invariant-check results as additional conjunctive certificate fields.

Table 1. Risk-aware release certification

Step	RA-DT release certification procedure
1	Verify checksums and compile candidate r.
2	Replay identical sampled flow IDs per twin scenario.
3	Compute tenant loss, FNR, and FPR.
4	Aggregate the five certificate quantities.
5	Normalize each quantity by its registered budget.
6	Set margin M; emit READY only when M <= 1.
7	Persist cause, tenant/scenario evidence, seeds, and hashes.

5. Experimental Methodology

Dataset and splits. UNSW-NB15 is a public intrusion-detection dataset containing modern normal and attack traffic and nine attack families [14], [15]. We use the author-provided training CSV (175,341 flows) and testing CSV (82,332 flows), verified by SHA-256 checksums in the artifact. To avoid the distribution mismatch between the official files influencing the release-gate ground truth, the official training file is stratified by attack category into model training (60%, 105,204 rows), twin validation (20%, 35,068), and independent production holdout (20%, 35,069). The official test file is retained only as an external detector-generalization check. This separation was selected after an initial design showed that some faulty transformations could accidentally improve metrics under the official train/test shift; retaining that split as gate ground truth would confound release safety with dataset shift.

Detector. We train a LightGBM binary classifier [16] on all dataset features except the record identifier, attack-category label, and binary target. Protocol, service, and state are categorical features. The model uses 260 trees, learning rate 0.065, 31 leaves, early stopping, and fixed seed 20260731. The operating threshold is selected on the twin-validation split to minimize tenant-criticality-weighted asymmetric loss. The resulting threshold is 0.18. On the untouched official test file, the detector achieves ROC AUC 0.9854, F1 0.8542, precision 0.7459, and recall 0.9993. These figures establish a functioning detector but are not the main contribution.

Safety labels and baselines. The clean, zero-offset candidate defines reference behavior. Twin SLO budgets are fixed relative to that reference before gate comparison: 1.20 times weighted and aggregate expected loss, 1.25 times CVaR, and additive allowances of 0.07 for worst-tenant FNR and maximum scenario FPR. Equivalent rules define independent production budgets from the clean production reference. A candidate is truly safe when its production margin is at most one. Baselines are aggregate nominal F1, aggregate expected loss, and CVaR-only. To compare gates fairly, each baseline passes the same number of candidates as RA-DT (47).

Evaluation questions and statistics. RQ1 asks whether twin risks rank production risks consistently. We report Spearman rank correlations for expected loss, CVaR, worst-tenant FNR, and the joint margin. RQ2 asks whether the gate prevents unsafe releases at matched throughput. Metrics are unsafe-release escape rate, safe-rejection rate, pass precision, balanced accuracy, and Matthews correlation coefficient. Percentile 95% confidence intervals are obtained from 3,000 candidate-level bootstrap resamples [29]. RQ3 asks which mutation families dominate risk; we report median production metrics by family. All numbers are produced by the supplied script and CSV outputs.

Replay controls and leakage prevention. Within a scenario, every candidate receives the same sampled flow identifiers so that differences are attributable to the release candidate rather than sampling noise. Twin and production scenario random streams are distinct, and production safety budgets are calculated only from the clean production reference. Candidate ranking uses twin outputs exclusively. The experiment does not tune gate thresholds against production labels; production labels are opened only for final comparison. The official test split is never used to define candidate safety.

Scenario semantics. The 12 scenarios are organized into nominal, attack-mixture, benign-load, missing-context, categorical-shift, rate/state-shift, and compound-stress families. Production scenarios use independent random seeds and stronger parameters rather than copies of the twin scenarios. This design evaluates transfer under controlled but non-identical conditions. It is not a statistical model of a specific SASE provider's future traffic, and the manuscript therefore reports it as a release-testing benchmark rather than a production forecast.

Table 2. Experiment configuration

Item	Setting
Real data	175,341 train; 82,332 external test
Internal split	105,204 model; 35,068 twin; 35,069 production
Synthetic context	8 tenants; weights 1.0–4.0
Candidates	7 families x 5 offsets x 3 shifts = 105
Stress replay	12 twin + 12 independent production scenarios
Replay volume	6,000 flows/scenario; seed 20260731
Risk	CVaR90; FN cost 10; FP cost 1

Table 3. Stress-scenario portfolio and independence design

Family	Twin	Production
Nominal	Reference mix; moderate burst	Independent draws; stronger burst
Benign load	Volume burst	Higher volume burst
Missingness	Controlled feature masks	Different masks; higher rates
Category shift	Rare/unseen substitutions	Independent unseen substitutions
Rate/state	Scaling; stale counters	Stronger independent perturbations
Compound	Two-factor combinations	Stronger; new seeds

6. Results

A. RQ1: Twin-to-production risk transfer. The twin and production rankings are strongly aligned. Spearman correlation is 0.9896 for criticality-weighted mean loss, 0.9172 for CVaR over tenant-scenario cells, 0.9347 for worst-tenant FNR, and 0.9778 for the joint readiness margin. The lower CVaR correlation is expected because tail estimates depend on which rare tenant-scenario cells are sampled. Nevertheless, the joint margin preserves almost all rank order, supporting its use for pre-release prioritization.

B. RQ2: Gate effectiveness. Production replay labels 56 candidates safe and 49 unsafe. At the common pass count of 47, the aggregate-F1 gate allows 10 unsafe candidates to pass, an escape rate of 20.4%. RA-DT allows one, or 2.04%, a 90% relative reduction. Pass precision rises from 78.7% to 97.9%, balanced accuracy from 0.728 to 0.901, and MCC from 0.458 to 0.804. Bootstrap intervals reflect the small candidate population: the RA-DT escape-rate interval is [0, 0.065], whereas aggregate F1 is [0.100, 0.317].

The expected-risk baseline has the same overall confusion counts as RA-DT, while CVaR-only escapes five unsafe candidates (10.2%). This non-dominating result is important. Expected risk is highly effective in the present mutation portfolio, so RA-DT cannot claim a lower aggregate escape count. RA-DT's added value is the form of assurance: it binds expected risk to tail, tenant-FNR, and scenario-FPR constraints and returns the responsible SLO. The CVaR-only result further shows that a tail statistic alone is insufficient; joint constraints are needed.

The one RA-DT escape is the swapped-directional-fields candidate with no global threshold offset and a +0.18 critical-tenant shift. Its twin margin is 0.986, immediately below the boundary, while its independent production margin is 1.017. This near-boundary miss suggests a practical policy: require a safety buffer or additional replay for candidates whose margin lies close to one, rather than treating the certificate as an uncertainty-free oracle.

C. RQ3: Mutation diagnostics. Dropping security context and the combined context/rate fault are clearly dominant, with median production CVaR losses of 1.778 and 1.802, respectively, compared with 0.462 for clean candidates. Stale counters and swapped directional fields produce smaller but

meaningful increases. The rate-unit fault alone has limited median effect in this trained model, illustrating why fault severity should be measured rather than assumed. Categorical fallback increases both expected and tail risk and frequently becomes unsafe when coupled with conservative threshold changes.

Table 4. Release gate comparison at matched pass count (47/105)

Gate	N	Esc.%	Rej.%	Prec.%	BAcc
Aggregate F1	10	20.4%	33.9%	78.7%	0.728
Expected risk	1	2.0%	17.9%	97.9%	0.901
CVaR only	5	10.2%	25.0%	89.4%	0.824
RA-DT	1	2.0%	17.9%	97.9%	0.901

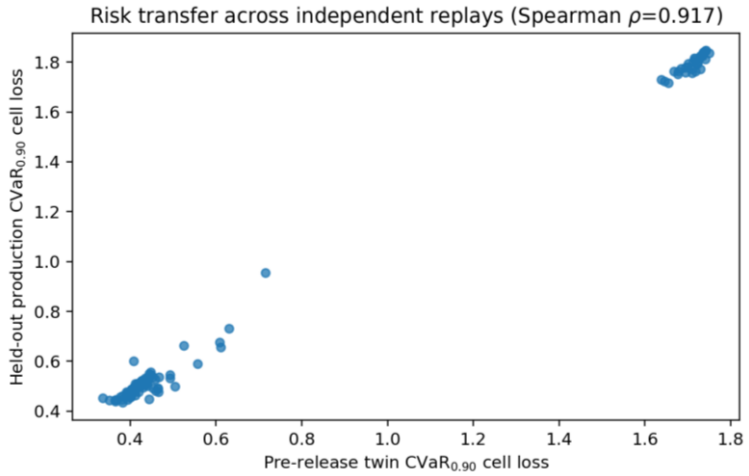

Figure 2. Twin-to-production transfer for CVaR risk across 105 release candidates.

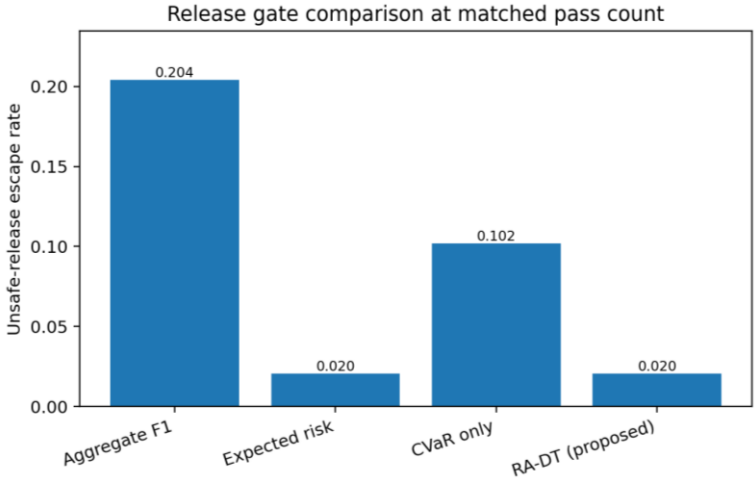

Figure 3. Unsafe-release escape rate at a matched pass count.

Table 5. Median production risk by mutation family

Mutation	Mean	CVaR90	Max FPR
Clean	0.080	0.462	0.333
Rate x1000	0.081	0.475	0.343
Category fallback	0.090	0.496	0.337
Stale counters	0.090	0.526	0.354
Direction swap	0.093	0.533	0.362
Dropped context	0.334	1.778	0.974
Combined bug	0.337	1.802	0.983

7. Ablation And Safety-Buffer Analysis

Certificate-component ablation. To isolate the role of each risk view, we rank candidates by one twin metric at a time and pass the best 47, exactly matching RA-DT throughput. Weighted expected loss produces the same binary decisions as the joint margin in this candidate portfolio: one unsafe escape, 17.9% safe rejection, and 97.9% pass precision. CVaR-only permits five unsafe candidates, while nominal F1 permits ten. Worst-tenant FNR-only and maximum-scenario FPR-only each permit fifteen because these metrics ignore complementary harm. The result does not imply that mean risk is always sufficient; it shows that the generated faults make weighted expected loss the active discriminator for many candidates. The joint certificate is therefore best understood as a guardrail architecture whose value appears when different constraints become active, rather than as a guaranteed numerical improvement in every dataset.

Active constraints and failure attribution. A maximum-normalized-risk certificate retains the reason for rejection. For example, context-dropping candidates are typically dominated by expected or tail loss, whereas threshold increases can be dominated by tenant FNR and threshold decreases by scenario FPR. This distinction matters operationally because corrective actions differ: a schema/context failure requires restoring telemetry integrity; an FNR failure calls for threshold or model review; an FPR failure may require load-aware tuning or exception handling. A single F1 value provides no comparable remediation path.

Safety-buffer sensitivity. The only RA-DT escape has a twin margin of 0.986 and a production margin of 1.017. We therefore evaluate a conservative margin cap without rerunning or relabeling candidates. Caps of 0.900, 0.925, 0.950, and 0.975 pass 21, 27, 34, and 36 candidates, respectively, with zero unsafe escapes in the independent holdout. The default cap of 1.000 passes 47 with one escape. At 0.975, pass precision is 100%, but 35.7% of safe candidates are rejected. This is an explicit security-throughput frontier: tighter caps reduce near-boundary transfer risk at the cost of slower delivery. Organizations can pre-register a buffer based on release criticality instead of silently changing the criterion after observing outcomes.

Interpretation limits. The buffer result is post-hoc sensitivity analysis on the same 105 candidates and should not be read as proof that 0.975 is universally optimal. Its purpose is to demonstrate a transparent operational control. A future implementation should estimate margin uncertainty through repeated scenario sampling, sequential confidence bounds, or conformal risk control, and allocate additional replay effort to candidates close to the decision boundary.

Table 6. Component Ablation At 47 Passed Candidates

Gate	N	Esc.%	Rej.%	Prec.%
F1	10	20.4%	33.9%	78.7%
Expected loss	1	2.0%	17.9%	97.9%
CVaR90	5	10.2%	25.0%	89.4%
Worst FNR	15	30.6%	42.9%	68.1%
Max FPR	15	30.6%	42.9%	68.1%
RA-DT	1	2.0%	17.9%	97.9%

Table 7. Release throughput under a stricter margin cap

Cap	Passed	Nesc	Escape	Safe-rej.	Precision
0.900	21	0	0.0%	62.5%	100.0%
0.925	27	0	0.0%	51.8%	100.0%
0.950	34	0	0.0%	39.3%	100.0%
0.975	36	0	0.0%	35.7%	100.0%
1.000	47	1	2.0%	17.9%	97.9%

8. Discussion

Release engineering implications. First, a security release gate should operate on the complete executable path, not only the model file. Several damaging candidates in this study are preprocessing or context faults. Second, equal pass-count evaluation matters: a gate can trivially reduce escapes by blocking everything. RA-DT is evaluated at the same delivery throughput as each baseline. Third, the certificate provides a compact review surface for change-approval boards. A blocked candidate can be traced to a specific normalized SLO, tenant, scenario, and mutation manifest.

Why the joint certificate matters even when expected risk ties. A scalar expected-loss gate can work well when the candidate population and criticality weights happen to align with deployment harm. It does not, however, state that worst-tenant FNR or maximum scenario FPR is within an explicit bound. These are materially different governance claims. RA-DT converts them into conjunctive release obligations, which is useful when contractual or regulatory objectives are expressed per tenant or per failure mode. The framework also permits organizations to change budgets without retraining the detector.

Operational integration. A production RA-DT service would ingest signed release artifacts, sanitize representative telemetry, generate deterministic and stochastic scenarios, and publish a machine-readable certificate. Canary deployment can then update the twin with observed shift. Candidates near

the margin should receive more samples, additional scenarios, or a stricter buffer. This adaptive replay policy is a direct extension of the observed near-boundary escape.

Threats to validity. Construct validity is limited because SASE is represented through network-flow detection, transformations, and policy thresholds; the experiment does not emulate every SASE function such as CASB, DLP, DNS security, or WAN path optimization. Tenant identities and criticalities are synthetic, although their generation is deterministic and disclosed. Internal validity is strengthened by isolated splits, fixed checksums, seeds, and code, but the relative SLO budgets are design choices rather than universal standards. External validity is limited to one public dataset and one tabular detector; real provider traces could have different drift and tenant structure. Statistical validity is constrained by 105 candidates, hence the bootstrap intervals and cautious interpretation. Finally, attack labels are treated as ground truth even though any benchmark can contain labeling limitations [15].

9. Reproducibility And Artifact Structure

The artifact is organized to let an independent reader regenerate every reported number. download_data.py retrieves the two official UNSW-NB15 CSV files and verifies SHA-256 digests before use. experiment.py performs preprocessing, deterministic tenant assignment, model fitting, mutation generation, scenario replay, gate evaluation, candidate-level bootstrapping, and figure generation. requirements.txt records the Python dependencies, while README.md gives command-line instructions and identifies which fields are real versus synthetic.

The supplied results directory contains candidate-level twin and production metrics, binary gate decisions, gate summaries, percentile bootstrap intervals, model metrics, safety limits, correlations, mutation summaries, figures, and the serialized detector. The experiment writes a machine-readable summary.json and a human-readable RESULTS.md. The training and test checksum prefixes are bec7dd5ec88d and 734fe6642edf, respectively; the artifact records the complete SHA-256 digests. Raw data are intentionally excluded from the compact delivery archive because the download script retrieves the official files and checks integrity.

Reproducibility does not eliminate benchmark limitations. The tenant mapping is a controlled experimental construct, not an assertion about the dataset's original provenance. Candidate mutations are executable transformations chosen to represent realistic release faults, but they are not observed incidents from a named provider. These boundaries are encoded in the artifact metadata and repeated in the manuscript so that subsequent work can replace the synthetic context with authorized enterprise traces without changing the certificate logic.

10. Conclusion

RA-DT frames SASE release readiness as a multi-tenant risk certification problem. It compiles complete release candidates, injects plausible implementation and policy mutations, replays tenant-aware stress scenarios, and issues a joint certificate over average loss, aggregate loss, CVaR, worst-tenant FNR, and maximum scenario FPR. In a reproducible UNSW-NB15 experiment, twin and independent production margins correlate at 0.978. At matched release throughput, RA-DT reduces unsafe escapes from 20.4% under an aggregate-F1 gate to 2.04%, while matching the expected-risk baseline and outperforming CVaR-only. The honest interpretation is not that the joint gate always beats every scalar risk measure, but that it turns release approval into explicit, auditable, tenant-aware obligations. Future work should validate the framework on privacy-preserving multi-provider traces, add deterministic policy invariants, and allocate replay effort adaptively near the readiness boundary.

References

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

[2] Chandramouli, R., & Butcher, Z. (2023, September). A zero trust architecture model for access control in cloud-native applications in multi-cloud environments. NIST Special Publication 800-207A. https://doi.org/10.6028/NIST.SP.800-207A

[3] Cybersecurity and Infrastructure Security Agency. (2023, April). Zero trust maturity model (Version 2.0).

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

[5] Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z., & Doss, R. (2022). Zero Trust Architecture (ZTA): A comprehensive survey. IEEE Access, 10, 57143–57179. https://doi.org/10.1109/ACCESS.2022.3174679

[6] Phiayura, P., & Teerakanok, S. (2023). A comprehensive framework for migrating to zero trust architecture. IEEE Access, 11, 19487–19511. https://doi.org/10.1109/ACCESS.2023.3248622

[7] 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

[8] 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 the IEEE International Conference on Smart Cloud (pp. 5–10). https://doi.org/10.1109/SmartCloud.2016.22

[9] Cybersecurity and Infrastructure Security Agency. (2026, June). The journey to zero trust: Using secure access service edge in a modern TIC 3.0 solution.

[10] Almasan, P., Pastor, A., Barlet-Ros, P., Yannuzzi, M., & Cabellos-Aparicio, A. (2022). Network digital twin: Context, enabling technologies, and opportunities. IEEE Communications Magazine, 60(11), 22–27. https://doi.org/10.1109/MCOM.001.2200012

[11] 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

[12] 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

[13] Grieves, M., & Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In Transdisciplinary perspectives on complex systems (pp. 85–113). Springer. https://doi.org/10.1007/978-3-319-38756-7_4

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

[15] 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: A Global Perspective, 25(1-3), 18–31. https://doi.org/10.1080/19393555.2015.1125974

[16] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems 30 (pp. 3146–3154).

[17] 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

[18] Duchi, J. C., & Namkoong, H. (2021). Learning models with uniform performance via distributionally robust optimization. The Annals of Statistics, 49(3), 1378–1406. https://doi.org/10.1214/20-AOS2004

[19] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning (Vol. 70, pp. 1321–1330). PMLR.

[20] Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., & Lakshminarayanan, B. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In Advances in Neural Information Processing Systems 32.

[21] 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 32.

[22] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3), e0118432. https://doi.org/10.1371/journal.pone.0118432

[23] Kazemian, P., Varghese, G., & McKeown, N. (2012). Header space analysis: Static checking for networks. In Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (pp. 113–126).

[24] Khurshid, A., Zou, X., Zhou, W., Caesar, M., & Godfrey, P. B. (2013). VeriFlow: Verifying network-wide invariants in real time. In Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (pp. 15–27).

[25] Mai, H., Khurshid, A., Agarwal, R., Caesar, M., Godfrey, P. B., & King, S. T. (2011). Debugging the data plane with Anteater. In Proceedings of the ACM SIGCOMM (pp. 290–301). https://doi.org/10.1145/2018436.2018470

[26] Al-Shaer, E., & Hamed, H. (2004). Discovery of policy anomalies in distributed firewalls. In Proceedings of the IEEE INFOCOM (Vol. 4, pp. 2605–2616).

[27] 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

[28] National Institute of Standards and Technology. (2012, September). Guide for conducting risk assessments. NIST Special Publication 800-30 Revision 1. https://doi.org/10.6028/NIST.SP.800-30r1

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

[30] Ding, J., Shen, Z., & Liu, W. (2026). Game-theoretic cost-sensitive adversarial training for robust cloud intrusion detection against GAN-based evasion attacks. Applied Sciences, 16(8), 3944. https://doi.org/10.3390/app16083944

[31] Zhao, W., Chen, T., Yang, J. S., & Qiu, L. (2026). AutoML-Pipeline: A RAG-enhanced code generation framework with pre-validation for cloud-native machine learning workflows. IEEE Access, 14, 41932–41945. https://doi.org/10.1109/ACCESS.2026.3673923

[32] Mo, T., Zhang, C., Zou, J., Guo, Z., & Rhee, M. (2026). Self-evolving AI agents with dual memory for automated software testing and bug localization. IEEE Access, 14, 111086–111102. https://doi.org/10.1109/ACCESS.2026.3713401

[33] Wang, B., Wang, Z., Zhao, W., Zhang, F., & Shang, W. (2026). DRL-Adapt: Deep reinforcement learning for adaptive routing convergence optimization in large-scale networks. IEEE Open Journal of the Communications Society, 7, 849–863. https://doi.org/10.1109/OJCS.2026.3687441