



SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Security-Aware Orchestration for Service Placement and Migration in 6G-Enabled Cloud–Fog–Edge Systems

Ravi Shankar¹, M.P. Mishra^{2*}

¹Research Scholar, School of Computer and Information Sciences, Indira Gandhi National Open University, Maidan Garhi, New Delhi-110068, India, E-mail: rvgr740@gmail.com, ORCID: 0009-0003-5403-7976

^{2*}Professor, School of Computer and Information Sciences, Indira Gandhi National Open University, Maidan Garhi, New Delhi-110068, India, E-mail: mpmishra@ignou.ac.in, ORCID: 0000-0002-5036-2881

ABSTRACT

In 6G-enabled Cloud–Fog–Edge systems, services may need to run on different Edge, Fog, and Cloud nodes as system conditions change. A key challenge is to make these placement and migration decisions while also considering the security of the available nodes. To address this challenge, this paper proposes a service-aware admission framework that links node-security evidence with orchestration decisions. A frozen Random-Forest detector maps 65 packet-capture-derived features to a security-event evidence score, which is used as ranking evidence rather than interpreted as a calibrated attack probability. Service-specific frozen thresholds map this score to ADMIT, ABSTAIN, or MISSING states according to different selective-risk requirements. The resulting admission rules are evaluated prospectively on a fresh 540-episode physical campaign using simultaneous one-sided Buehler confidence limits. The corresponding selective-risk upper bounds are 0.0440, 0.1384, and 0.0941 for service risk targets of 0.05, 0.20, and 0.10, respectively. The frozen admission rules are then integrated with resource and latency constraints for service placement and migration in a paired 50-seed Cloud–Fog–Edge orchestration study. Relative to a resource-only baseline, the proposed policy reduces attack-active execution from 7.47% to 1.70%, while incurring higher latency and a small no-placement rate. Compared with a generic score-threshold baseline, attack exposure and latency are statistically indistinguishable, while no-placement is significantly lower. These results demonstrate that security evidence can be incorporated into service placement and migration while making the resulting security–availability–latency trade-off explicit.

Keywords: 6G, Cloud–Fog–Edge computing, security-aware orchestration, service admission, Buehler bounds.

Introduction

The progression from centralized cloud platforms toward edge and fog computing has been driven largely by latency, bandwidth, locality, and resilience requirements [1], [2]. The same shift also changes the security problem. A service may now move among heterogeneous execution sites whose instantaneous security state is uncertain, partially observed, and time varying. This matters directly in emerging 6G-enabled distributed-computing settings, where security and resilience are explicit system capabilities rather than optional add-ons [3]. A placement controller that ignores cyber-security evidence may keep a latency-critical workload on a compromised or actively attacked node; a controller that reacts too aggressively to uncertain evidence can instead create migration churn, latency inflation, or service unavailability.

The literature on secure edge/fog scheduling has recognized that performance-only placement is inadequate. Trust-aware application placement, security-aware task offloading, and security-constrained scheduling have been proposed using trust scores, cryptographic overheads, security levels, heuristic costs, and learning-based optimizers [4]–[7]. Recent reviews likewise identify security-aware scheduling as a growing cross-layer problem in heterogeneous cloud, fog, and edge systems [8]. However, a practical gap remains between two communities. Security detectors typically output a score or label, while orchestration algorithms typically assume that a node's security state or trust score is already usable. The statistical question—when is imperfect security evidence sufficiently reliable for a particular service to remain or move? —is often left implicit.

Figure 1 places the study in the broader 6G-enabled Cloud–Fog–Edge environment. Security monitoring and trust management span the Edge, Fog, and Cloud tiers, while service orchestration and resource management coordinate placement across the distributed infrastructure. This paper addresses that gap by inserting an explicit, service-aware selective-admission layer between security sensing and orchestration. The central idea is simple but consequential. The detector does not decide placement, and the orchestrator does not receive ground-truth attack labels. Instead, a frozen detector produces a scalar security-event evidence score. For service s , a threshold τ_s maps the latest applicable causal score to ADMIT or ABSTAIN; if no applicable causal admission state is available, the state is MISSING. The threshold is paired with a service-specific selective-risk requirement ρ_s . Thus, the same node evidence can be acceptable for one service and unacceptable for another. The downstream controller sees only the admission state together with ordinary resource and latency feasibility.

This separation is important for three reasons. First, it prevents an uncalibrated machine-learning score from being silently interpreted as an attack probability. Second, it makes service heterogeneity explicit: a strict service need not share the same admission rule as a tolerant service. Third, it creates a clean statistical object that can be certified prospectively before the orchestration experiment is run. This follows the general philosophy of selective prediction, in which a predictor is permitted to abstain in exchange for controlled risk on the accepted subset [12]-[14], but the present setting differs in that the selective decision is a service admission gate feeding a distributed systems controller rather than a final classifier output.

The study combines physical and controlled-simulation evidence, and the distinction is deliberate. Detector development and certification use real packet-capture data collected from Docker-capable VMware nodes representing Edge, Fog, and Cloud. The final orchestration comparison is a controlled simulation using the frozen, physically certified admission rule. Fresh certification rows are not replayed as orchestration samples, and the orchestrator never receives certification internals. This separation avoids adapting system-level outcomes to the certification set while still allowing the operational consequences of the frozen rule to be studied under repeatable paired scenarios.

The main contributions are as follows.

- We formulate a service-aware security admission layer in which a common physical security-evidence score is converted into distinct ADMIT/ABSTAIN/MISSING states under service-specific risk targets, without treating the score as a calibrated attack probability.
- We build a 65-feature multi-protocol physical evidence pipeline and freeze a Random-Forest detector and three service-specific thresholds before a separate prospective certification campaign.
- We provide a simultaneous finite-sample certificate for all three service admission rules using one-sided Buehler limits and a familywise error allocation across the normal- and security-active admission components used by the frozen certificate.
- We integrate the frozen admission states with resource- and latency-feasible Cloud-Fog-Edge orchestration and quantify the resulting security-latency-availability trade-off against a resource-only baseline, a generic score-threshold baseline, and a perfect-information oracle. A secondary six-condition study isolates the operational impact of false alarms, misses, delay, missing evidence, and temporal correlation.

The remainder of the paper is organized as follows. Section II reviews the relevant literature. Section III presents the system and risk model. Section IV develops the service-aware admission and orchestration method. Section V describes the physical testbed, dataset, certification campaign, and controlled orchestration experiments. Section VI reports numerical results and robustness findings. Section VII concludes the paper and summarizes the main claim boundaries and future research directions.

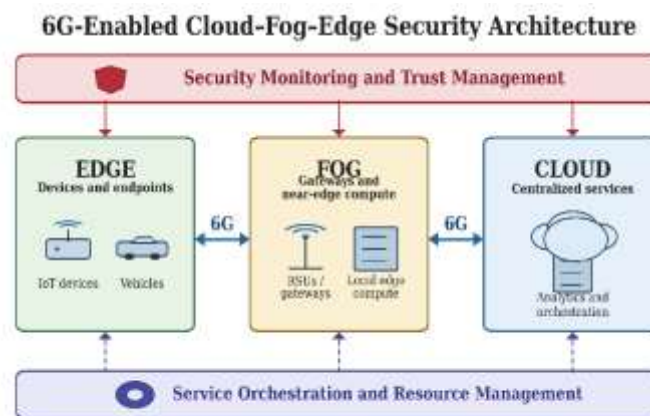


Fig. 1. 6G-enabled Cloud-Fog-Edge security and orchestration architecture.

II. Related Work

A. Edge/Fog Computing and Security-Aware Placement

Edge and fog computing move computation and storage toward users and devices to reduce response time and network load [1], [2]. The same decentralization introduces a broader attack surface, heterogeneous trust domains, and security-management challenges that differ from a centrally administered cloud [9]. In this setting, placement decisions cannot be based only on CPU and delay when an execution node may become unsafe during service operation.

Several lines of work therefore incorporate security into offloading or placement. Yadav and Baranwal [4] proposed trust-aware application placement in fog systems. Li et al. [5] jointly considered security demand and deadline constraints in edge-cloud task offloading. Almuselem [6] combined security, load balancing, and energy considerations in a multi-tier edge-cloud architecture. Long et al. [7] designed a security-aware DAG scheduler based on a trust/risk model and risk-dependent security mechanisms. A recent systematic review confirms a broad movement toward security-aware task scheduling across cloud, fog, and edge environments, while also highlighting fragmentation in risk assessment, enforcement, optimization targets, and evaluation practice [8]. Recent work on the broader compute continuum also reviews service-placement algorithms and practical edge-cloud deployment, while 6G-oriented studies consider cross-domain orchestration across heterogeneous infrastructures [2], [10], [11].

These studies establish the importance of security-aware scheduling, but their security variables are usually directly consumable by the scheduler: a trust value, a security requirement, a security level, or a risk estimate. Our focus is one step earlier. We ask how noisy physical cyber-security observations become an operationally usable service-placement constraint without exposing hidden truth or treating an arbitrary detector score as a probability. The proposed method therefore complements rather than replaces scheduling optimization: the admission layer filters candidate nodes before the existing resource/latency placement logic is applied.

B. Selective Prediction and Abstention

Selective classification formalizes the idea that a predictor may reject difficult inputs to achieve lower error on the covered subset. El-Yaniv and Wiener [12] characterized the risk-coverage trade-off, while later work constructed high-probability selective classifiers and integrated reject options into learned predictors [13], [14]. The same conceptual structure is useful for cyber-security evidence: an admission mechanism need not declare every node “safe” or “unsafe.” It can withhold positive admission when evidence is too risky or unavailable.

Our formulation differs in three ways. First, the positive action is not a class label; it is permission for a particular service to use a node. Second, each service has a distinct risk requirement ρ_s , so one evidence score can lead to different decisions for different services. Third, the selected rules are subjected to a fresh prospective physical campaign after threshold freezing. This last step is intended to prevent post-hoc tuning on certification outcomes.

C. Finite-Sample Certification

For a service-admission claim, raw empirical error is insufficient. A small observed risk can still be compatible with a substantially larger population risk when the sample is finite. We therefore use one-sided Buehler confidence limits, which were originally developed for exact confidence statements involving binomial parameters [15] and have subsequently been studied as finite-sample confidence constructions under monotonicity constraints [16]. In the frozen prospective design, separate one-sided bounds are constructed for service-specific admission probabilities in the normal and security-active strata and then combined through ratio monotonicity. The goal is not to claim that the detector is universally safe; it is to certify a sharply defined service-specific admission rule under the balanced prospective protocol and a stated familywise confidence allocation.

III. System Model and Problem Formulation

A. Cloud-Fog-Edge Model

Let $\mathcal{N} = \mathcal{N}_E \cup \mathcal{N}_F \cup \mathcal{N}_C$ denote the available compute nodes. The logical evaluation topology contains two Edge nodes, two Fog nodes, and one Cloud node. Let C_n denote the total normalized CPU capacity of node n ; the Edge, Fog, and Cloud capacities are 4, 8, and 16 units per node, respectively. Let $c_{s,n}(t)$ denote the CPU capacity available to service s on node n immediately before the orchestration decision at epoch t , as exposed by the resource ledger. When $n = p_s(t-1)$, the service’s own active CPU allocation is credited back when evaluating continued placement. Let $\mathcal{S} = \{A, B, C\}$ denote the three services. Service s has CPU demand d_s , maximum serving latency $\bar{\ell}_s$, and selective security-risk requirement ρ_s . The frozen service contracts are shown in Table I.

Table I: Service Contracts

Service	CPU d_s	ρ_s	$\bar{\ell}_s$ (ms)
A	1	0.05	15
B	2	0.20	30
C	3	0.10	20

The non-monotone ordering of latency, CPU, and security requirements is intentional. In particular, Service A is the strictest security contract, Service C is intermediate, and Service B is the most tolerant. Security criticality is therefore represented explicitly through ρ_s rather than inferred from CPU demand or latency class.

Let $p_s(t-1) \in \mathcal{N} \cup \{\emptyset\}$ denote the authoritative serving node immediately before decision epoch t , and let

$p_s(t)$ denote the resulting placement after the decision and any completed migration transition. A resulting placement is resource-feasible when

$$\sum_{s:p_s(t)=n} d_s \leq C_n, \quad \forall n \in \mathcal{N}, \quad (1)$$

and latency-feasible for service s on node n when

$$\ell_{s,n}(t) \leq \bar{\ell}_s. \quad (2)$$

Figure 2 summarizes the service-placement model. The admission state defined below is combined with residual CPU capacity and serving-latency feasibility to construct the risk-admissible candidate set used by the orchestrator.

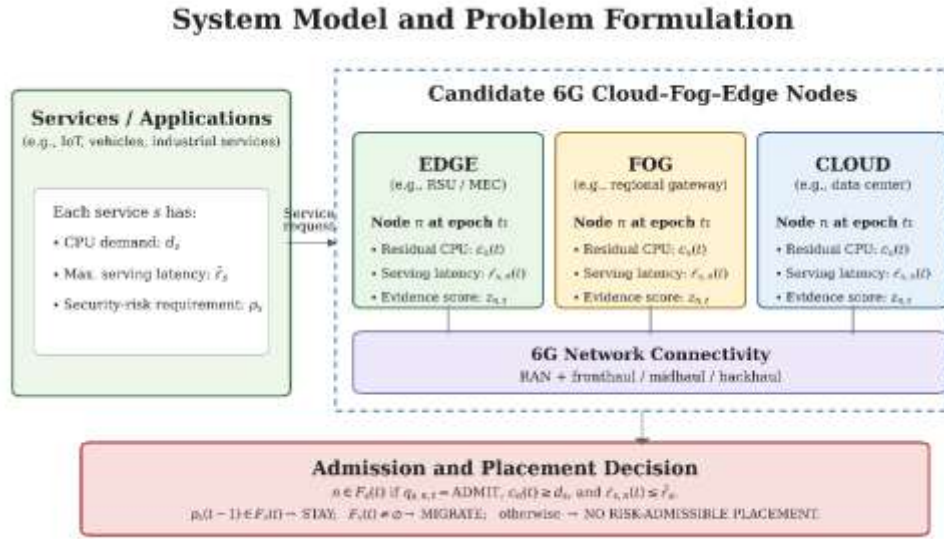


Fig. 2. Service-aware security admission and placement system model.

B. Security Evidence and Truth

For a node observation available at decision epoch t , let $\mathbf{x}_{n,t} \in \mathbb{R}^{65}$ denote the ordered packet-capture-derived feature vector. The frozen detector f_θ returns

$$z_{n,t} = f_\theta(\mathbf{x}_{n,t}) \in [0,1], \quad (3)$$

where larger $z_{n,t}$ means stronger evidence for the controlled security-event-active class. We deliberately call $z_{n,t}$ a security-event evidence score. Although the Random-Forest implementation exposes the score through a class-1 predict_proba interface, the score is not assumed to be calibrated as $\Pr(Y = 1 | \mathbf{x}_{n,t})$.

For offline development and certification, let the physical truth associated with an observation be

$$Y \in \{0,1\}, \quad (4)$$

where $Y = 0$ denotes PHYS_NORMAL and $Y = 1$ denotes CONTROLLED_SECURITY_EVENT_ACTIVE. Ground truth is evaluator-owned. It is used to develop and certify the admission rule, but it is not an input to the operational orchestrator.

C. Service-Specific Admission

For service s , let τ_s be its frozen admission threshold. Given the latest applicable causal evidence score $z_{n,t}$ for node n , define the effective admission state

$$q_{s,n,t} = \begin{cases} \text{ADMIT}, & z_{n,t} \leq \tau_s, \\ \text{ABSTAIN}, & z_{n,t} > \tau_s, \\ \text{MISSING}, & \text{no applicable causal evidence is available.} \end{cases} \quad (5)$$

Once an ADMIT or ABSTAIN state has been issued, the latest causal state remains effective until a newer causal observation updates it. Thus, absence of a fresh observation does not by itself overwrite an existing admission

state with MISSING.

Only ADMIT is considered security-feasible for new or continued service placement. The key service-aware property is immediate: if $\tau_A < z_{n,t} \leq \tau_B$, a node can be rejected for A but accepted for B. The service identity therefore changes the admissibility decision without changing the underlying detector or physical observation.

D. Selective Risk

For a generic scored observation in the certification population, define the service- s admission indicator

$$G_s = \mathbb{I}\{z \leq \tau_s\}. \quad (6)$$

The selective false-safe risk is

$$R_s = \Pr(Y = 1 \mid G_s = 1). \quad (7)$$

The prospective CERT540 protocol is balanced between the two physical truth strata, so $\Pr(Y = 0) = \Pr(Y = 1) = 1/2$ within the stated certification population. Define the class-conditional admission probabilities

$$a_{y,s} = \Pr(G_s = 1 \mid Y = y), \quad y \in \{0,1\}. \quad (8)$$

Bayes' rule therefore gives, for this balanced prospective population,

$$R_s = \frac{a_{1,s}}{a_{0,s} + a_{1,s}}, \quad a_{0,s} + a_{1,s} > 0. \quad (9)$$

The service requirement is

$$R_s \leq \rho_s. \quad (10)$$

This is an admission-level risk statement tied to the frozen prospective population/protocol. It is not the uncontrolled attack prevalence of an arbitrary deployment, and ρ_s is a service-specific selective-risk target, not a detector threshold.

E. Orchestration Objective

The proposed architecture does not collapse security, latency, and availability into a single weighted score. Doing so would impose an application-specific value judgment that is not part of the frozen experiment. Instead, security acts as a feasibility gate and the downstream controller preserves the existing resource/latency semantics.

For each service, define the admissible candidate set

$$\mathcal{F}_s(t) = \{n \in \mathcal{N} : q_{s,n,t} = \text{Admit}, c_{s,n}(t) \geq d_s, \ell_{s,n}(t) \leq \bar{\ell}_s\}. \quad (11)$$

If the pre-decision serving node $p_s(t-1)$ remains in $\mathcal{F}_s(t)$, the least disruptive action is STAY. Otherwise, if $\mathcal{F}_s(t) \neq \emptyset$, the existing deterministic placement rule selects a feasible destination and triggers MIGRATE. If no risk-admissible feasible destination exists, the controller reports NO RISK-ADMISSIBLE PLACEMENT. This constraint-first formulation is useful because the security guarantee remains interpretable independently of the resource optimizer.

IV. Proposed Service-Aware Admission Framework

Figure 3 summarizes the complete proposed framework. Panel (A) uses the three-node physical testbed for security-evidence development, service-specific threshold selection, and fresh prospective certification. The resulting frozen admission interface exposes only ADMIT, ABSTAIN, or MISSING.

Panel (B) transfers this frozen interface to the controlled five-node Cloud-Fog-Edge orchestration topology. The proposed policy combines the per-service admission state with resource and latency feasibility to select STAY, MIGRATE, or NO RISK-ADMISSIBLE PLACEMENT. Thus, physical evidence development and certification remain separated from the downstream controlled orchestration evaluation.

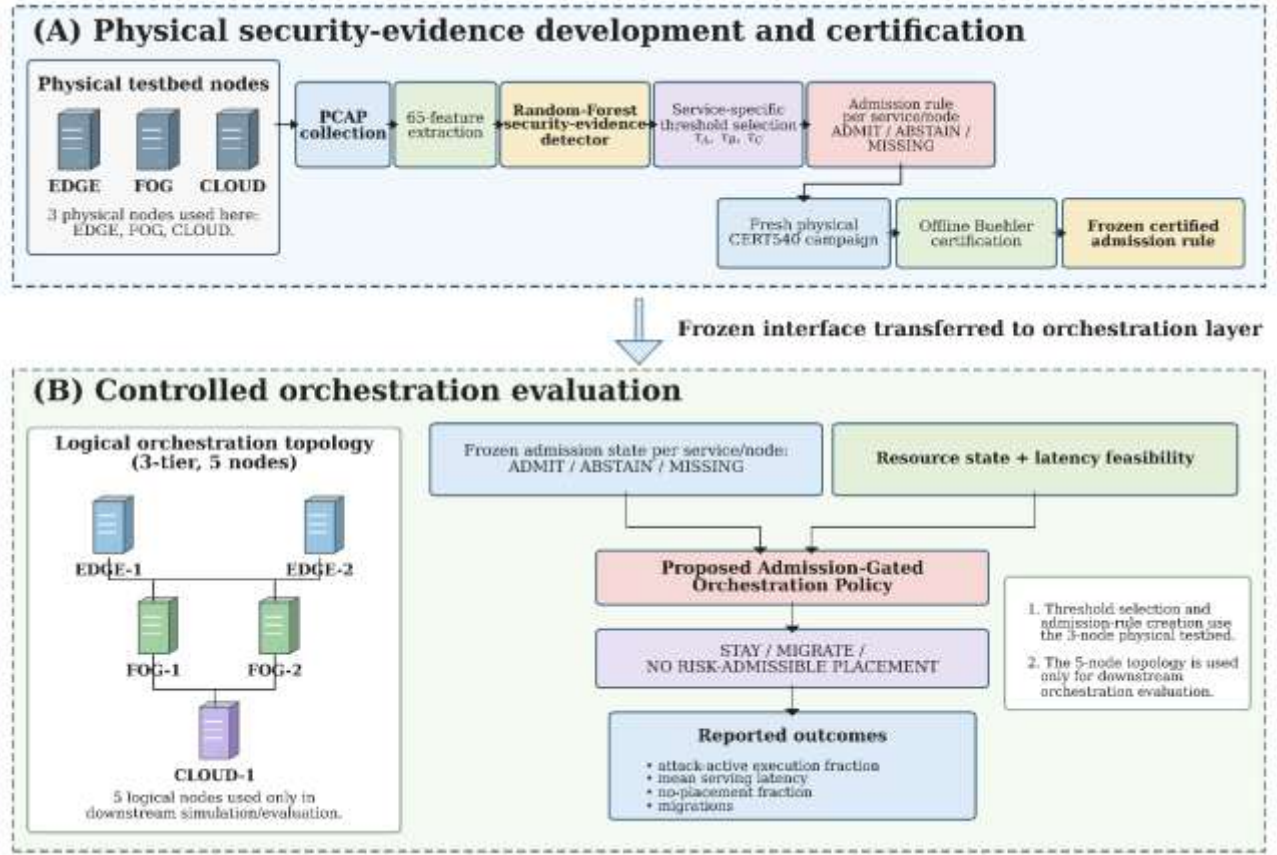


Fig. 3. Proposed service-aware admission and orchestration framework.

A. Frozen Detector and Threshold Development

The detector-development dataset contains 540 valid physical episodes. The frozen split is 324 TRAIN rows, 108 VALIDATION rows, and 108 HELD_OUT rows. The selected detector is a Random Forest [17] implemented as a frozen numericization adapter followed by a RandomForestClassifier. We emphasize that the contribution is not the use of Random Forest itself; the detector is the evidence source for the admission framework.

Service thresholds were developed only after the detector was fixed. The threshold-development pool used the 216 non-TRAIN rows (108 DV2 and 108 DV4 rows). The resulting thresholds were frozen before prospective certification:

$$\begin{aligned} \tau_A &= 0.5667228343, & \tau_B &= 0.6001413684, \\ \tau_C &= 0.5780239380 \end{aligned} \quad (12)$$

No fresh certification row was allowed to refit the detector or retune any τ_s .

A useful way to view the threshold-development objective is as a constrained coverage problem. Let $\mathcal{T}$ denote the finite candidate-threshold set used by the frozen development protocol, let $\hat{\mathcal{C}}_s(\tau)$ denote the fraction of development evidence admitted by threshold $\tau \in \mathcal{T}$, and let $\hat{U}_s^{\text{dev}}(\tau)$ denote the development-stage conservative risk quantity used by that protocol. The design seeks a practically useful threshold while respecting the service target,

$$\tau_s \in \operatorname{argmax}_{\tau \in \mathcal{T}} \hat{\mathcal{C}}_s(\tau) \quad \text{s.t.} \quad \hat{U}_s^{\text{dev}}(\tau) \leq \rho_s, \quad (13)$$

with the important methodological rule that Eq. (13) is a development step, not the final certificate. The final claim comes only from the fresh prospective campaign described next.

B. Prospective Buehler Certificate

After the detector and thresholds were frozen, a new 540-episode physical certification schedule was executed. Let $n_0 = n_1 = 270$ denote the numbers of normal and security-active certification episodes. For certification episode i , let $G_{s,i} = \mathbb{I}\{z_i \leq \tau_s\}$ denote the admission indicator for service s . The admitted counts are

$$A_{y,s} = \sum_{i: Y_i=y} G_{s,i}, \quad y \in \{0,1\}, \quad (14)$$

and the empirical class-conditional admission probabilities

$$\hat{a}_{y,s} = \frac{A_{y,s}}{n_y}. \quad (15)$$

Because the prospective protocol is balanced, the empirical selective risk can equivalently be written as

$$\hat{R}_s = \frac{A_{1,s}}{A_{0,s} + A_{1,s}}, \quad A_{0,s} + A_{1,s} > 0. \quad (16)$$

The frozen certificate does not place a confidence limit directly on $\hat{R}_s$. Instead, it constructs a one-sided Buehler lower bound $L_{0,s}$ for the normal-stratum admission probability $a_{0,s}$ and a one-sided Buehler upper bound $U_{1,s}$ for the security-active-stratum admission probability $a_{1,s}$. Since $a_{1,s}/(a_{0,s} + a_{1,s})$ is increasing in $a_{1,s}$ and decreasing in $a_{0,s}$, ratio monotonicity gives the conservative selective-risk upper bound

$$U_{R,s} = \frac{U_{1,s}}{L_{0,s} + U_{1,s}}, \quad L_{0,s} + U_{1,s} > 0. \quad (17)$$

The service is certified when

$$U_{R,s} \leq \rho_s. \quad (18)$$

For reference, the locked directional components $(U_{1,s}, L_{0,s})$ are approximately (0.04332, 0.94065), (0.15195, 0.94586), and (0.09821, 0.94586) for Services A, B, and C, respectively; Eq. (17) reproduces the final reported upper bounds.

The familywise failure probability is fixed at $\Delta_{\text{tot}} = 0.05$. With three service claims and two directional bounds per service, six simultaneous one-sided bounds are required. We therefore use

$$\alpha_{\text{dir}} = \frac{\Delta_{\text{tot}}}{3 \times 2} = \frac{0.05}{6} = 0.008333 \dots \quad (19)$$

This allocation is frozen before the certificate is interpreted. The Buehler quantities are offline certificate objects; they are not controller inputs, and no score-plus/minus-uncertainty rule is introduced.

C. Proposed Admission-Gated Orchestration Policy

Algorithm 1 summarizes the operational logic of the proposed Admission-Gated Orchestration Policy (implemented in the codebase as AdmissionGatedPolicy). The orchestrator receives the admission state, not the raw detector score, ground truth, ρ_s , or confidence-bound internals. Resource accounting remains controlled by the resource ledger and latency retains its normal serving-placement meaning.

Algorithm 1 Proposed Admission-Gated Orchestration Policy

Require: Services $\mathcal{S}$, nodes $\mathcal{N}$, current placements $p_s(t-1)$, effective causal admission states $q_{s,n,t}$, and resource/latency state

- 1: for each service $s \in \mathcal{S}$ do
- 2: Build $\mathcal{F}_s(t)$ using Eq. (11)
- 3: if $p_s(t-1) \in \mathcal{F}_s(t)$ then
- 4: action $\leftarrow$ STAY
- 5: else if $\mathcal{F}_s(t) \neq \emptyset$ then
- 6: destination $\leftarrow$ lowest-latency node in $\mathcal{F}_s(t)$, breaking ties by node identifier
- 7: action $\leftarrow$ MIGRATE
- 8: else
- 9: action $\leftarrow$ NO RISK-ADMISSIBLE PLACEMENT
- 10: end if
- 11: end for

D. Evaluation Metrics

We report three primary system outcomes. Let T_s be total episode time, T_s^{atk} the time service s executes while its serving node is attack-active, and T_s^{nop} the time with no serving placement. Then

$$E_s = \frac{T_s^{\text{atk}}}{T_s} \quad (20)$$

is the attack-active execution fraction and

$$N_s = \frac{T_s^{\text{nop}}}{T_s} \quad (21)$$

is the no-placement fraction. Here, no-placement is a serving-state metric: it includes any interval in which the service has no active serving node, including migration cutover/downtime. It is therefore distinct from the controller outcome NO RISK-ADMISSIBLE PLACEMENT, which indicates that no risk-admissible feasible placement is available. Mean serving latency L_s is calculated only over intervals in which the service is actually placed; no-placement time is never recoded as zero-latency service. Migration count is reported as a secondary operational metric rather than folded into a weighted score.

V. Physical Testbed, Dataset, and Experimental Method

A. Physical Testbed

The physical evidence pipeline uses three Docker-capable VMware virtual machines representing Edge, Fog, and Cloud, together with a separate traffic generator. The generator creates approved benign and controlled security-event activity; it is not an orchestration node. Three containerized application services provide the workload context, while packet captures are collected and transformed into the frozen feature representation. The physical truth is binary: PHYS_NORMAL or CONTROLLED_SECURITY_EVENT_ACTIVE. Service identity is not a detector class. This separation is essential because the same detector output is later interpreted under three different service-risk contracts.

B. 65-Feature Multi-Protocol Dataset

One physical episode produces one packet capture, one ordered 65-feature vector, and one dataset row. The feature set contains 23 common transport/L4 features and 42 protocol-specific L7 features: 10 HTTP, 9 SSH, 8 MQTT, 7 CoAP, and 8 DNS features. The final detector-development branch spans nine profiles: P1, P2, P3, N1, H1, H2, MQTT, CoAP, and DNS.

The 540-row development dataset is balanced by the frozen design and split as follows:

$$|\mathcal{D}_{\text{train}}| = 324, \quad |\mathcal{D}_{\text{val}}| = 108, \quad |\mathcal{D}_{\text{held}}| = 108. \quad (22)$$

The split semantics are TRAIN = DV0+DV1+DV3, VALIDATION = DV2, and HELD_OUT = DV4. The selected CANDIDATE-3 Random Forest is frozen as a whole pipeline, including numericization and classifier stages.

C. Independent Zeek Cross-Validation

As an independent check of the frozen physical feature representation, the same immutable packet captures from the 540-episode detector-development campaign, and the fresh 540-episode certification campaign were reprocessed through a separate Zeek-based reconstruction pipeline. Across 1,080 packet captures and 65 features per episode, this produced 70,195 exact matches out of 70,200 feature comparisons (99.9929%). The five remaining differences were confined to mqtt_min_inter_event_gap_ns and arose from the different timestamp-association semantics of the frozen Tshark frame.time_epoch path and the independent Zeek event reconstruction. No tolerance matching, packet-capture modification, detector refitting, threshold retuning, or certificate recomputation was used. This analysis is therefore treated as an independent feature-reproducibility check; Zeek does not replace the frozen feature extractor, detector, or certification procedure.

D. Fresh Prospective CERT540 Campaign

The certification campaign is independent of threshold adaptation. Its factorization is

$$\begin{aligned} & 9 \text{ profiles} \times 5 \text{ DV conditions} \times 3 \text{ nodes} \\ & \times 2 \text{ truths} \times 2 \text{ repetitions} = 540. \end{aligned} \quad (23)$$

All 540 scheduled physical episodes completed validly on the first physical attempt. The campaign contains 270 normal and 270 security-active episodes, 180 episodes per Edge/Fog/Cloud tier, 108 per DV condition, and 60 per profile. Freshness checks found zero exact overlap in packet-capture hashes and zero exact 65-feature-vector matches with the detector-development data.

E. Controlled Orchestration Evaluation

After certification, the detector, thresholds, ρ_s , and certificate are held fixed. The orchestration experiment is controlled simulation, not a second physical campaign. This distinction matters: the physical testbed establishes and certifies the security-admission rule, while controlled simulation evaluates how that frozen

rule changes placement behavior under paired exogenous scenarios. Table II summarizes the key frozen parameters of the controlled orchestration study. In this runtime, $c_{s,n}(t)$ is the causal CPU capacity available to service s on candidate node n from the authoritative resource ledger, including credit for the service's own active allocation when its current serving node is evaluated. Serving latency is a controlled placement-latency abstraction: each node has a fixed base value b_n , and the feasibility test instantiates $\ell_{s,n}(t) = b_n$ for candidate node n . These base values are not claimed to be measured VMware, radio, queueing, or end-to-end request latencies.

The main final comparison uses the missing-evidence operating condition and 50 paired random seeds (400000–400049). For each seed, one exogenous realization is generated once and replayed across four policies:

1. ResourceOnlyPolicy: ignores security evidence and uses resource/latency feasibility only;
2. ProbabilityThresholdPolicy: generic score-threshold baseline with fixed threshold 0.5;
3. Proposed Admission-Gated Orchestration Policy (AdmissionGatedPolicy): service-specific frozen certified admission gate;
4. SimulationPerfectInformationOracle: evaluator-only ideal reference with simulated truth.

All policies start from the same neutral placement and share the same exogenous realization within a seed, but retain independent mutable runtime/resource state. The 50 seeds therefore yield 200 policy episodes and 600 service-level rows. The oracle is descriptive and excluded from confirmatory inference.

Table II: Key Controlled-Orchestration Simulation Parameters

Parameter	Frozen value / semantics
Logical topology	EDGE-1, EDGE-2, FOG-1, FOG-2, CLOUD-1
CPU capacities C_n	4, 4, 8, 8, 16 normalized units
Base latencies b_n	5, 6, 10, 12, 25 ms, respectively
Service contracts	Table I; CPU demands 1/2/3 and latency limits 15/30/20 ms
Simulation horizon	300 s
Evaluation schedule	Start 0 s; 5-s interval; 60 decision epochs
Attack-active truth process	Starts at 40, 90, 140, 190, 240 s; 20-s duration; target node sampled uniformly from the five logical nodes
Migration lifecycle	Eight nonzero inter-stage durations, each 0.5 s
Main evidence condition	CASE-MISSING; each scheduled evidence opportunity omitted independently with probability 1/5; otherwise, nominal evidence model
Main seed family	50 paired seeds, 400000–400049
Policies	ResourceOnly; ProbabilityThreshold (score cutoff 0.5); Proposed Policy; perfect-information Oracle
Confirmatory inference	Seed is replication unit; 20,000 paired bootstrap resamples; Bonferroni-adjusted 99.1667% CI per contrast

The main statistical analysis uses the seed as the replication unit because services within one episode share workload, attack timing, and evidence opportunities. Three primary metrics are tested against two non-oracle comparators, producing six paired confirmatory contrasts. We report paired bootstrap confidence intervals using 20,000 resamples. To control the familywise error rate at 0.05 across the six predeclared contrasts, we apply a Bonferroni adjustment, so that each contrast uses an error level of $0.05/6 = 0.008333 \dots$ and therefore a confidence level of $1 - 0.05/6 = 99.1667\%$. No weighted composite score and no p-value-based ranking are used.

F. Uncertainty Robustness Cases

A secondary descriptive study evaluates the proposed policy under one nominal and five uncertainty conditions using the same 50-seed family. Table III summarizes their meanings. The main CASE-MISSING experiment remains the only confirmatory performance comparison using the frozen admission rule. CASE-N, CASE-FA, CASE-MISS, CASE-DELAY, and CASE-CORR are reported as nominal/robustness evidence and are not promoted to additional certificate claims.

Table III: Controlled Evidence Conditions

Case	Operational meaning
CASE-N	Nominal reference; no additional injected impairment.
CASE-FA	Additional false-alarm pressure during normal periods.
CASE-MISS	Additional false-safe/miss behavior during attack-active periods.
CASE-DELAY	Evidence delivery delayed by one evaluation epoch.
CASE-MISSING	Scheduled evidence/issuance is absent; missing is not converted into a numeric

Case	Operational meaning
	score.
CASE-CORR	Temporally correlated burst errors with matched marginal behavior.

VI. Numerical Results and Discussion

A. Prospective Three-Service Certification

Table IV reports the fresh CERT540 results. Service A admits 263 of 270 normal episodes and 4 of 270 security-active episodes, giving 267 total admissions and an empirical selective risk of 1.50%. The Buehler upper bound is 4.40%, below $\rho_A = 5\%$. Service B admits more security-active rows, as expected from its more tolerant contract, but its 13.84% upper bound remains below $\rho_B = 20\%$. Service C lies between the two, with a 9.41% upper bound below $\rho_C = 10\%$. All three service rules therefore satisfy Eq. (18) under the frozen prospective protocol; Fig. 4 visualizes the same result against the three risk targets.

Table IV: Fresh Prospective Cert540 Selective-Risk Certificate

Service	τ_s	ρ_s	Normal admitted	Security admitted	Total admitted	Empirical $\hat{R}_s$	Buehler $U_{R,s}$
A	0.566723	0.05	263/270	4/270	267/540	0.01498	0.04402
B	0.600141	0.20	264/270	27/270	291/540	0.09278	0.13841
C	0.578024	0.10	264/270	15/270	279/540	0.05376	0.09406

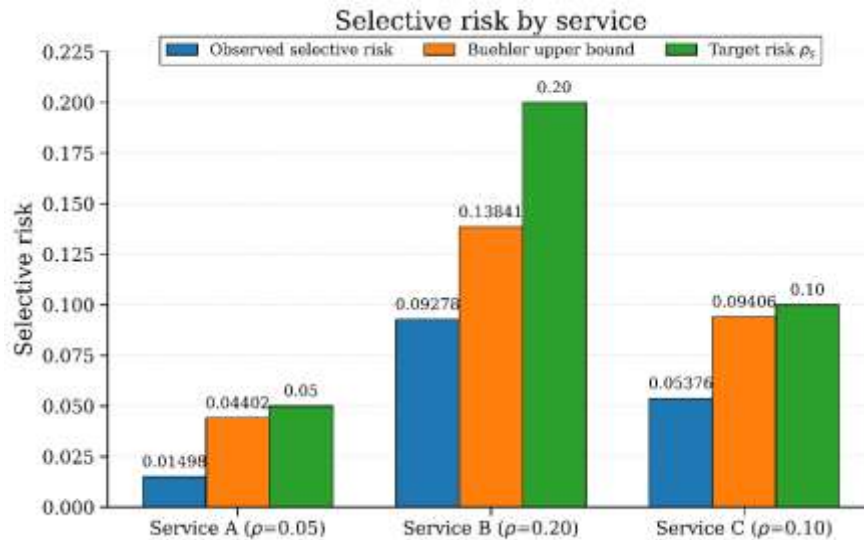


Fig. 4. Selective-risk certification for Services A, B, and C.

Two points are worth emphasizing. First, the service ordering is not determined solely by the numeric threshold. Thresholds are coupled to different ρ_s and to the frozen evidence distribution. Second, the certificate is a statement about the complete admission rule under the prospective population/protocol; it is not a universal property of Random Forest or a claim that every future traffic distribution will satisfy the same bounds.

B. Main Orchestration Comparison

Table V and Fig. 5 compare the four policies under CASE-MISSING using equal-service macro means for attack-active execution, mean serving latency, and no-placement fraction. Lower values are preferable for all three metrics. The perfect-information Oracle is included only as an evaluator-side descriptive reference. The resource-only policy has the best latency and zero no-placement because it never withholds a node for security reasons, but this comes with a 7.47% attack-active execution fraction. The proposed Admission-Gated Orchestration Policy reduces this exposure to 1.70%. The cost is a modest latency increase to 6.57 ms and a no-placement fraction of 0.61%.

Table V: MAIN 50-SEED CASE-MISSING ORCHESTRATION RESULTS

Policy	Attack-active execution	Mean serving latency (ms)	No-placement fraction	Migrations/seed
Resource Only Policy	0.07467	5.333	0.00000	0.00
Probability Threshold Policy	0.01462	6.624	0.00804	7.24

Policy	Attack-active execution	Mean serving latency (ms)	No-placement fraction	Migrations/seed
Proposed policy	0.01696	6.566	0.00607	5.46
Perfect Information Oracle	0.00729	6.231	0.00364	3.28

Relative to the ResourceOnlyPolicy baseline, the proposed method reduces mean attack-active execution by 5.77 percentage points, approximately a 77% relative reduction. This is the main security effect of admission gating. It should not be presented without the accompanying operational cost: the security gate changes placement decisions and migration activity, producing higher serving latency and a small amount of no-placement time associated with migration cutover/downtime. No policy produced a No Risk-Admissible Placement decision in the main CASE-MISSING experiment.

The comparison with the ProbabilityThresholdPolicy baseline is more informative about the value of service-specific admission. Its security exposure is slightly lower in the raw mean (1.46% versus 1.70%), while the proposed method has lower latency, lower no-placement, and fewer migrations. However, raw means alone do not establish statistical separation.

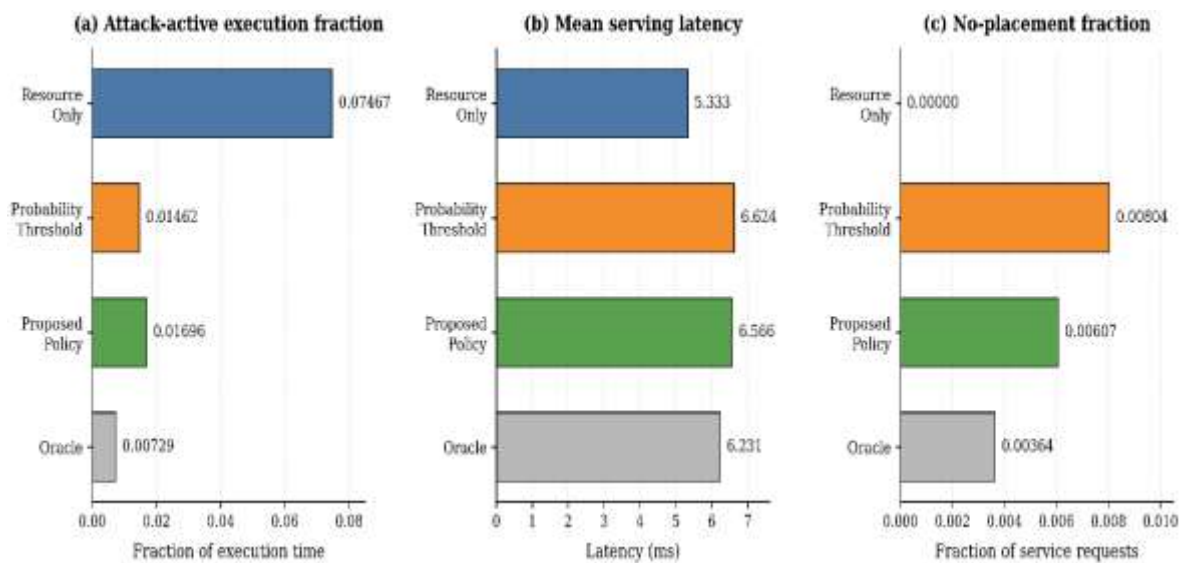


Fig. 5. Main CASE-MISSING orchestration performance comparison.

C. Confirmatory Paired Inference

Table VI reports the six predeclared paired effects, defined as the proposed policy minus comparator. Negative values favor the proposed method for all three metrics. Each contrast is reported with a Bonferroni-adjusted 99.1667% confidence interval, corresponding to the familywise error-rate target of 0.05 across the six-contrast family.

Against ResourceOnlyPolicy, the attack-active execution difference is -0.05771 with Bonferroni-adjusted interval $[-0.07273, -0.04338]$, confirming lower attack exposure. The latency difference is $+1.2325$ ms with interval $[1.0282, 1.4720]$, confirming a latency cost. No-placement increases by $+0.00607$ with interval $[0.00504, 0.00713]$. The result is therefore a clear trade-off, not a free improvement.

Against ProbabilityThresholdPolicy, the attack-exposure difference is $+0.00233$ and the latency difference is -0.0584 ms; both Bonferroni-adjusted intervals cross zero. These two outcomes are statistically inconclusive. By contrast, the no-placement difference is -0.00198 with Bonferroni-adjusted interval $[-0.00320, -0.00084]$, supporting lower no-placement for the service-specific proposed policy.

Table VI: Confirmatory Paired Effects: Proposed Policy Minus Comparator

Comparator	Metric	Mean effect	99.1667% adjusted CI	Interpretation
Resource Only	Attack-active execution	-0.05771	$[-0.07273, -0.04338]$	Proposed lower
Resource Only	Latency (ms)	+1.23247	$[1.02822, 1.47202]$	Proposed higher
Resource Only	No-placement	+0.00607	$[0.00504, 0.00713]$	Proposed higher

Comparator	Metric	Mean effect	99.1667% adjusted CI	Interpretation
Probability Threshold	Attack-active execution	+0.00233	[-0.00184, 0.00653]	Inconclusive
Probability Threshold	Latency (ms)	-0.05841	[-0.28447, 0.12516]	Inconclusive
Probability Threshold	No-placement	-0.00198	[-0.00320, -0.00084]	Proposed lower

This result explains why an “overall best” claim would be misleading. Compared with the security-agnostic baseline, admission gating strongly reduces security exposure but intentionally gives up some availability and latency. Compared with a generic threshold, the service-specific rule mainly improves placement availability while leaving the other two primary outcomes statistically unresolved at the familywise level. The intended contribution is therefore controlled, service-aware risk management rather than universal domination of every metric.

D. Service Movement in the Main Case

Table VII summarizes the corresponding service-level placement shares and movement decisions in the main CASE-MISSING experiment. Across the 50 CASE-MISSING seeds, Service A migrated 93 times, Service B 81 times, and Service C 99 times, for 273 migrations in total, or 5.46 per seed. At least one migration occurred in 44, 43, and 40 seeds for A, B, and C, respectively. These counts are operationally useful because they show that the migration burden is not dominated by only one service. They should not, however, be confused with the number of Stay decisions; a seed with zero migration can still contain many control epochs at which the controller explicitly chose to stay.

Table VII: Service Placement And Movement Under Case-Missing For The Proposed Policy

Service	Edge (%)	Fog (%)	Cloud (%)	No placement (%)	STAY	MIGRATE	To Edge	To Fog	To Cloud
A	95.08	4.30	0.00	0.62	2907	93	87	6	0
B	69.63	29.83	0.00	0.54	2919	81	44	37	0
C	65.89	33.45	0.00	0.66	2901	99	50	49	0
Overall	76.86	22.53	0.00	0.61	8727	273	181	92	0

E. Six-Condition Robustness Study

Table VIII compares the proposed policy across the nominal reference and five uncertainty mechanisms.

Table VIII: Proposed Policy Under Nominal and Five Uncertainty Conditions

Condition	Attack-active execution	Mean latency (ms)	No-placement fraction	Migrations/seed
CASE-N	0.00951	6.492	0.00582	5.24
CASE-FA	0.01120	7.540	0.03780	34.02
CASE-MISS	0.01438	6.486	0.00584	5.26
CASE-DELAY	0.02780	6.470	0.00582	5.24
CASE-MISSING	0.01696	6.566	0.00607	5.46
CASE-CORR	0.01482	7.531	0.01493	13.44

These results are descriptive. They use the same 50-seed family, but we do not retrofit the main six-contrast confirmatory inference after observing the robustness outcomes.

The nominal CASE-N condition produces 0.95% attack-active execution, 6.49 ms mean serving latency, 0.58% no-placement, and 5.24 migrations per seed. Three uncertainty cases remain close to this movement level: CASE-MISS (5.26), CASE-DELAY (5.24), and CASE-MISSING (5.46) migrations per seed. Their consequences differ, however. CASE-DELAY yields the largest attack exposure among the six conditions (2.78%), which is consistent with a controller continuing to act on older evidence before the delayed observation becomes causally available.

False alarms produce a qualitatively different failure mode. CASE-FA keeps attack exposure low (1.12%) but drives migration count to 34.02 per seed and no-placement to 3.78%. Service-wise, the proposed policy records 601, 583, and 517 migrations for A, B, and C across the 50 seeds. Thus, a conservative security signal can protect against unsafe execution while still being operationally undesirable because it repeatedly disqualifies otherwise healthy nodes.

CASE-CORR is the second most disruptive condition by movement, with 13.44 migrations per seed and 1.49% no-placement. Temporal clustering matters even when a marginal error rate appears manageable because

several consecutive adverse decisions can exhaust feasible alternatives or trigger repeated placement changes. This observation motivates modeling temporal dependence explicitly rather than evaluating only independent per-epoch error.

F. Interpretation

The experiments expose three practical lessons. First, security-aware orchestration should not be evaluated only by attack exposure. A mechanism can achieve low exposure by rejecting many nodes, yet produce unacceptable migration churn or availability loss. Second, service-specific admission is useful even when a generic threshold is competitive on average security exposure. In the main experiment, the service-specific policy reduces no-placement and migration burden relative to the generic threshold while maintaining statistically indistinguishable attack exposure and latency. Third, uncertainty type matters. False alarms, misses, delay, missing evidence, and correlated errors do not simply scale a common performance curve; they produce different operational failure signatures.

These lessons are why the proposed architecture keeps three primary outcomes separate rather than reporting a single weighted utility. A deployment may legitimately choose different operating points depending on whether security exposure, latency, continuity, or migration overhead is dominant. The contribution of the certificate is to make the security side explicit and auditable before those deployment-specific trade-offs are considered.

VII. Conclusion

This work examined how imperfect security evidence can be used in Cloud-Fog-Edge service placement without treating a detector score as an attack probability. A frozen 65-feature Random-Forest detector and service-specific admission thresholds were evaluated in a fresh 540-episode physical certification campaign. The resulting Buehler selective-risk upper bounds were 4.40%, 13.84%, and 9.41% for Services A, B, and C, respectively, all below their predefined risk targets.

The certified admission rules were then used in a paired 50-seed orchestration study. Compared with the resource-only baseline, the proposed policy reduced attack-active execution from 7.47% to 1.70%, although this came with higher serving latency and a small amount of no-placement time. Compared with the generic 0.5 score-threshold baseline, attack exposure and latency were not statistically distinguishable, while the proposed policy achieved lower no-placement. The robustness study also showed that different evidence failures lead to different operational effects: false alarms mainly increased migration activity, delayed evidence increased attack exposure, and correlated errors increased both movement and availability cost.

These results are limited to the frozen detector, feature representation, thresholds, and prospective certification protocol used in this study. The orchestration results come from controlled simulation, so the reported latency and migration values should not be interpreted as production-network measurements. The additional robustness cases are descriptive, and the 6G scope of this work is the Cloud-Fog-Edge orchestration layer rather than the radio interface.

Overall, the results show that service-specific security admission can make the security-availability-latency trade-off explicit before placement decisions are made. Future work should test the same admission interface in a live multi-node orchestration system and study migration hysteresis, recovery delay, and joint risk-aware optimization without weakening the statistical admission guarantee.

References

1. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.
2. L. Belcastro, F. Marozzo, A. Orsino, D. Talia, and P. Trunfio, "Navigating the edge-cloud continuum: A state-of-practice survey," *IEEE Access*, vol. 14, pp. 40622–40647, 2026, doi: 10.1109/ACCESS.2026.3673012.
3. ITU-R, "Framework and overall objectives of the future development of IMT for 2030 and beyond," Recommendation ITU-R M.2160-0, Nov. 2023.
4. R. Yadav and G. Baranwal, "Trust-aware framework for application placement in fog computing," in *Proc. IEEE Int. Conf. Advanced Networks and Telecommunications Systems (ANTS)*, 2019, pp. 1–6, doi: 10.1109/ANTS47819.2019.9118122.
5. Z. Li, V. Chang, H. Hu, D. Yu, J. Ge, and B. Huang, "Profit maximization for security-aware task offloading in edge-cloud environment," *J. Parallel Distrib. Comput.*, vol. 157, pp. 43–55, Nov. 2021, doi: 10.1016/j.jpdc.2021.05.016.
6. W. Almuseelem, "Energy-efficient and security-aware task offloading for multi-tier edge-cloud computing systems," *IEEE Access*, vol. 11, pp. 66428–66439, 2023, doi: 10.1109/ACCESS.2023.3290139.
7. L. Long, Z. Liu, J. Shen, and Y. Jiang, "SecDS: A security-aware DAG task scheduling strategy for edge computing," *Future Gener. Comput. Syst.*, vol. 166, Art. no. 107627, 2025, doi: 10.1016/j.future.2025.107627.

- 10.1016/j.future.2024.107627.
8. S. Al Shamaa and W. Shi, "A systematic review of security-aware task scheduling in cloud, fog, and edge computing," *J. Cloud Comput.*, 2026, doi: 10.1186/s13677-026-00936-z.
 9. M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19293–19304, 2017, doi: 10.1109/ACCESS.2017.2749422.
 10. I. Taleb, J.-L. Guillaume, and B. Duthil, "A survey on services placement algorithms in integrated cloud-fog/edge computing," *ACM Comput. Surv.*, vol. 57, no. 11, Art. no. 271, pp. 1–36, 2025, doi: 10.1145/3729214.
 11. R. Cuervo Bello, G. Scivoletto, G. Landi, R. Mihai, C. C. Petrache, M. Camelo, J. Marquez-Barja, and N. Slamnik-Krijestorac, "Zero-touch cross-domain orchestration in the 6G compute continuum: A framework evaluation in real-life scenarios," *IEEE Open J. Commun. Soc.*, vol. 7, pp. 1376–1395, 2026, doi: 10.1109/OJCOMS.2026.3662044.
 12. R. El-Yaniv and Y. Wiener, "On the foundations of noise-free selective classification," *J. Mach. Learn. Res.*, vol. 11, pp. 1605–1641, 2010.
 13. Y. Geifman and R. El-Yaniv, "Selective classification for deep neural networks," in *Advances in Neural Information Processing Systems* 30, 2017, pp. 4878–4887.
 14. Y. Geifman and R. El-Yaniv, "SelectiveNet: A deep neural network with an integrated reject option," in *Proc. 36th Int. Conf. Machine Learning*, vol. 97, 2019, pp. 2151–2159.
 15. R. J. Buehler, "Confidence intervals for the product of two binomial parameters," *J. Amer. Statist. Assoc.*, vol. 52, no. 280, pp. 482–493, 1957, doi: 10.1080/01621459.1957.10501404.
 16. P. Kabaila, "Better Buehler confidence limits," *Statist. Probab. Lett.*, vol. 52, no. 2, pp. 145–154, Apr. 2001, doi: 10.1016/S0167-7152(00)00199-1.
 17. L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.