TWIN-IDS: Twin-driven IDS Placement for Cost-aware Detection

Vincenzo Sammartino, Fabrizio Baiardi, Salvatore Ruggieri

Dipartimento di Informatica, Università di Pisa, Pisa, Italy
Email: vincenzo.sammartino@phd.unipi.it, {f.baiardi, salvatore.ruggieri}@unipi.it

***Abstract*—Deciding where to deploy intrusion detection sensors in an enterprise network is a budget-constrained design problem: sensors differ widely in cost, the network changes continuously, and the value of a location depends on the attack paths that cross it. We present TWIN-IDS, a framework that solves this problem *inside a live security digital twin* rather than on a static snapshot. TWIN-IDS introduces no new detection or optimisation primitive; its contribution is the integration of four established ingredients into one closed loop: a twin continuously synchronised with the physical network by the NotLine platform; edge-compromise probabilities derived from real vulnerability data (6,847 catalogue entries over 12 service classes, blended with exploit-prediction priors); an adaptive Monte Carlo intrusion simulator that estimates node risk *and* per-sensor detection probability in a single pass against twin-captured traffic baselines; and a monetary-budget knapsack solver whose placement is then stress-tested against four adversary strategies in a worst-case (Stackelberg) analysis. Because intrusions are replayed on the twin, arbitrarily many samples can be drawn without disturbing the production network, and any topology or vulnerability change triggers re-optimisation within a single monitoring cycle. Evaluated on two synthetic topology families (up to 10,000 nodes) and on twins of a university campus (126 nodes) and an industrial automation plant (202 nodes), TWIN-IDS improves detection by 18–41 percentage points over a random baseline at equal budget, reaches 80% detection with 47–55% fewer sensors, and raises the worst-case detection floor by 17.1%. A controlled decomposition separates the contribution of vulnerability-grounded edge weights ($+7$ pp) from that of path-level Monte Carlo scoring ($+12$ pp), and simulated detection probabilities agree within 2.3 pp with rule-based and machine-learning detectors.**

***Index Terms*—Security Digital Twin, IDS Placement, CVE/CVSS Vulnerability Model, Monte Carlo Simulation, Budget-constrained Optimization, Attack Path Analysis**

I. Introduction

The protection of enterprise networks with thousands of heterogeneous nodes against multi-stage intrusions requires a strategic deployment of intrusion detection system (IDS) sensors under a hard *monetary budget*. Given a network graph, candidate monitoring locations, a budget ceiling, and an attacker model, IDS placement is NP-hard [1] as it simultaneously depends on topology, edge-compromise probabilities, cost heterogeneity, and adversary behaviour [2]. Static, centrality-based heuristics cannot capture these interdependencies, and the few existing optimisation-based approaches rely on synthetic vulnerability assignments that diverge from the real CVE landscape.

Open gaps. A systematic literature review of IDS placement in Section II reveals four persistent deficiencies. (G1) Graph-centrality methods ignore the CVE/CVSS landscape that governs actual attack-path probabilities; existing vulnerability-aware methods rely on synthetic or ad hoc weight assignments. (G2) Most studies constrain the *number* of sensors rather than their monetary cost, implicitly assuming a uniform deployment cost. This assumption is violated in every real enterprise where, e.g., the cost of monitoring a SCADA gateway is an order of magnitude larger than the cost of monitoring a workstation. (G3) Placement algorithms typically assume a static network topology, but real-world environments continuously evolve because of patching, provisioning, and asset migration. (G4) Evaluating defenses against a single, fixed attack strategy overlooks the reality of a rational adversary who continuously adapts its tactics in response to the defender's evolving posture.

Contributions. TWIN-IDS is deliberately a *systems* contribution. Its building blocks—noisy-OR vulnerability aggregation, Monte Carlo risk estimation, 0/1 knapsack optimisation, Stackelberg worst-case analysis—are established techniques and we claim no novelty for any in isolation. What is new is their composition into one twin-driven loop, under a single probabilistic semantics, that closes the four gaps simultaneously rather than one at a time:

- An *end-to-end cost-aware IDS placement pipeline* coupling a four-step noisy-OR CVE/CVSS enrichment stage (6,847 entries; CVSS v3.1 temporal scoring [3]; EPSS priors [4]; role/action multipliers) with a monetary-budget 0/1 knapsack DP [5] (sample-optimality guarantee) and five risk-aware greedy strategies, plus a segment-consolidation post-processor for hybrid HIDS/NIDS deployment (§IV-A–§IV-D).
- A *simulation-derived detection model* embedded in an adaptive Monte Carlo, MC, risk engine. The detection probability $\hat{P}_{\mathrm{det}}(v, a)$ at a node v is estimated within the same simulation pass against twin-captured traffic baselines, with role-adaptive threshold calibration validated against SNORT [6] and Isolation Forest [7] within 2.3 pp (§IV-A, §IV-B).
- A *worst-case evaluation protocol* that applies the standard Stackelberg formulation across four ATT&CK-weighted adversary modes, with LP-based SSE validation via column generation [8] (§IV-E).

- A *comprehensive evaluation* on two synthetic topology families and two twins of real networks, with a rigorous algorithmic decomposition that isolates the CVE weight contribution (+7 pp) from the path-level MC algorithm (+12 pp) over the Liu et al. baseline (§V).

II. Related Work

Security digital twins. Eckhart and Ekelhart [9], [10] pioneered virtual security environments for ICS; surveys and taxonomies appear in [11]–[14], and Khan et al. [15] use AI-enabled twins for static-snapshot risk assessment. The NotLine platform [16] instead maintains a *live* twin by passive multi-protocol monitoring, inferring topology, service inventory, and per-node traffic baselines without active probing; [17] formalises twin-based Monte Carlo for ICT risk assessment and directly informs our risk engine. Follow-up work on the same platform extends it along complementary axes: [18] turns the twin into a real-time risk-assessment engine, [19] evaluates adversary strategies [20] directly against the twin rather than against a static graph, [21] proposes a quantitative validation methodology for twin-based cyber defense, and [22] uses AI-generated synthetic data to train twin-based defenses without exposing production traffic; TWIN-IDS builds on this line of work by adding cost-aware sensor placement on top of the live twin. Further work in this research line covers non-intrusive proactive-resilience modeling [23], twin-based intrusion defeat in cyber-physical systems [24], [25], graph-based cyber defence over a security twin [26], the transition from digital twins to AI-agent defenders with synthetic data [27]–[30], the effect of CVSS score ordering on attack paths [31], resilience quantification against zero-day threats via digital- and security-twin what-if analysis [32]–[34], financial risk quantification via security twins [35], formal characterization of twin fidelity [36], passive-monitoring-based DoS and virtual-machine detection [37]–[40].

IDS placement. Graph-centric methods rank nodes by betweenness, degree, or PageRank [41] but ignore cost heterogeneity; Liu et al. [42] assume a node-count budget with synthetic edge weights, [43] models a Stackelberg game without CVE-grounded weights, and Dewri et al. [44] give a multi-objective variant that also omits cost heterogeneity. Anomaly-detection surveys and systems appear in [45]–[47]. No prior work combines real CVE data, monetary cost heterogeneity, dynamic topology, and adversarial worst-case evaluation in one framework.

Ghosh et al. [48] address the complementary, downstream problem of deterministically allocating detection *rules* to existing sensors under capacity constraints; TWIN-IDS decides instead where the sensors should be, probabilistically and on a live twin.

Probabilistic attack graphs and security games. CVSS-based metrics, scalable generation, and aggregation are studied in [2], [49]–[51], while Poolsappasit et al. [52] use dynamic Bayesian graphs with single-CVE probabilities; Allodi and Massacci [53] show that CVSS base scores overestimate exploit likelihood, which motivates our EPSS-blended weighting. Stackelberg security games are surveyed in [54], with deployed systems and equilibrium methods in [55]–[58]. We adopt these formulations unchanged; to our knowledge TWIN-IDS is the first IDS placement framework to couple an exact monetary-budget knapsack with Stackelberg evaluation over four attacker models on real operational twins.

III. System Definition and Threat Model

This section defines the various entities we use in the following. We assume the NotLine platform builds the DTs we refer to in the following [16].

A Security DT at time t is a state snapshot $\mathcal{T}(t)$ that provides the basis for the enterprise network graph. Formally, it is a tuple $\mathcal{T}(t) = (V, E, \mathcal{L}, \ell, \mathcal{R}, \lambda, \Sigma, \sigma, \mathcal{B}, \beta)$, where:

- V is the set of network nodes (assets).
- $E \subseteq V \times V \times \mathcal{A}$ is the set of typed communication edges, where $\mathcal{A}$ represents the set of possible adversary actions (e.g., pivoting, lateral movement).
- $\mathcal{L} = \langle L_{\text{dmz}}, L_{\text{app}}, L_{\text{int}}, L_{\text{db}}, L_{\text{scada}} \rangle$ is a tuple with five network layers, representing the logical segmentation of the enterprise architecture.
- $\ell : V \to \mathcal{L}$ maps each node to its architectural layer.
- $\mathcal{R}$ is the set of node roles, and $\lambda : V \to \mathcal{R}$ maps each node to its specific functional role (e.g., `web_server`, `scada_gateway`).
- Σ is the set of known services from which vulnerabilities are derived.
- $\sigma : V \to 2^{\Sigma}$ maps each node to the services it runs.
- $\mathcal{B}$ is the space of baseline traffic profiles, and $\beta : V \to \mathcal{B}$ assigns each node v its profile, $\beta(v) = (\boldsymbol{\mu}(v), \boldsymbol{\sigma}^2(v))$, where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ are the empirical mean and variance of the traffic observables.

The tuple $\mathcal{L}$ defines a five-tier decomposition: the Demilitarized Zone (L_{dmz}) handles external-facing assets, followed by the Application (L_{app}), Internal (L_{int}), Database (L_{db}), and SCADA (L_{scada}) segments. In some networks, one or more tiers may be empty. The attributes $\ell(v)$ and $\lambda(v)$ are correlated but distinct, and NotLine infers them from different evidence: ℓ from the addressing plan (VLAN and subnet), λ from the observed service fingerprint $\sigma(v)$. Each role is admissible in only a subset of layers—a `scada_gateway` never sits in L_{dmz}—so λ constrains ℓ without determining it, a `web_server` being legitimate in either L_{dmz} or L_{app}. Both are retained because costs and detection thresholds are indexed by role, whereas entry points and targets are indexed by layer. $\mathcal{T}(t)$ provides the foundational data to derive the simplified weighted graph G for risk analysis.

A. Network Graph Model

Definition 1 (Enterprise Network Graph)**.** *An enterprise network is modelled as a directed weighted multigraph $G = (V, E, w, \lambda, \kappa)$, where V is the finite set of nodes; $E \subseteq V \times V \times \mathcal{A}$ with action set $\mathcal{A} = \{\texttt{pivot}, \texttt{lateral}, \texttt{cred\_theft}, \texttt{priv\_esc}\}$; $w : E \to [0.05, 0.98]$ is the edge-compromise probability (§IV-A); λ :*

TABLE I: Base monitoring cost $\bar{\kappa}(r)$ by node role in MU [59].

Role tier	Example roles	$\bar{\kappa}(r)$ (MU)
Critical OT/DB	`scada_gateway`, `db_primary`	3,500
High-value App	`app_server`, `auth_service`	2,000
DMZ / Perimeter	`web_server`, `mail_relay`	1,200
Infrastructure	`firewall`, `dns`, `ldap`	800
Internal	`workstation`, `monitoring`	350

$V \to \mathcal{R}$ assigns one of twelve node roles; and $\kappa : V \to \mathbb{R}^+$ is the cost of deploying a sensor.

The topology maps to the five-tier architecture $\mathcal{L}$ defined in Section III. Monitoring cost is non-uniform, reflecting the real variance in total cost of ownership across asset types [59]: $\kappa(v) = \bar{\kappa}(\lambda(v)) \cdot \eta_v$ with $\eta_v \sim \mathrm{Uniform}(0.85, 1.15)$. Base costs range from 350 MU for a workstation to 3,500 MU for a critical OT gateway, which carries specialised hardware, licensing, and operational overhead (Table I).

A *segment* $\mathcal{S}_k \subseteq V$ is a maximal set of nodes sharing a /24 prefix or VLAN inferred by NotLine; its *gateway* g_k is the node whose ARP table covers the broadest address range in the segment.

B. Threat Model

We consider an adversary who gains an initial foothold at $v_{\text{entry}} \in L_{\text{dmz}}$ and targets a high-value asset $v_{\text{target}} \in L_{\text{db}} \cup L_{\text{scada}}$. Entry-point probabilities are derived from the initial access vector frequencies reported in DBIR 2024 [60]:

$$P(v_{\text{entry}} = v) = \frac{\phi(\lambda(v))}{\sum_{u \in L_{\text{dmz}}} \phi(\lambda(u))}, \quad v \in L_{\text{dmz}} \qquad (1)$$

where $\phi(r)$ is the share of breaches whose initial access vector maps to role r. These values are not tuned: they are read off the DBIR 2024 distribution [60]—web application exploitation (0.39, `web_server`), phishing (0.35, `mail_relay`), abuse of remote access with stolen credentials (0.15, `vpn`/`rdp_gateway`)—with the residual 0.11 (physical access, supply chain, unattributed) spread uniformly over the other DMZ roles. Any other incident corpus can be substituted; replacing this distribution by a uniform one costs 7.3 pp of detection (§V-B).

The adversary actions in an intrusion are driven by its *strategy* [16] $m \in \mathcal{M} = \{\texttt{Rnd}, \texttt{Mpp}, \texttt{Stlth}, \texttt{Nsy}\}$. Four strategies are considered for adversaries sharing the same goal: a random walk (`Rnd`); maximum-probability path selection (`Mpp`), greedily taking the highest-w edge at each hop; stealth (`Stlth`), minimising detection exposure; and aggressive probing (`Nsy`). Other strategies are possible, but these span the spectrum of rational adversary behaviour [54], [58] and are combined via ATT&CK-derived frequency weights (§IV-E).

IV. The TWIN-IDS Framework

Figure 1 shows the end-to-end pipeline. (i) The twin, built and continuously synchronised by NotLine through passive monitoring, supplies live topology, service inventory, and traffic baselines; any change (asset addition, patch event, VLAN migration) raises a $\Delta\mathcal{T}$ update and a re-optimisation cycle—the bidirectional synchronisation that distinguishes a live twin from a static snapshot. (ii) CVE/CVSS enrichment turns service vulnerabilities into probabilistic edge weights. (iii) A Monte Carlo risk engine simulates intrusions on the twin, scoring nodes while estimating per-sensor detection probabilities against the captured baselines. (iv) A cost-aware optimiser selects the placement, which (v) is evaluated under multiple adversary models. End-to-end latency is below 3 min on the enterprise twin ($|V| \approx 235$).

The twin in the loop. Two properties distinguish this pipeline from an offline optimiser reading a network inventory. First, intrusions are *executed on the twin*: sample size is bounded by computation, not by what production can tolerate, and behaviours unacceptable in an operating plant (aggressive scanning, credential abuse on a SCADA gateway) are sampled freely. Second, the loop is automatic on the input side and gated on the output side. Physical-to-virtual, NotLine's passive monitoring detects asset additions, patch events, and VLAN migrations and emits a delta $\Delta\mathcal{T}$ that invalidates the affected edge weights and triggers re-optimisation. Virtual-to-physical, TWIN-IDS is *advisory*: it emits a placement diff—sensors to add or withdraw, with cost and detection deltas—enacted by operators or by the configuration-management system. Deployment is not actuated automatically, because touching an operational asset requires change-management approval and because an auto-actuated posture would be a predictable function of topology changes that an adversary able to provoke them could exploit.

The same engine serves two regimes. *At design time* it is the evaluation instrument that compares strategies, adversary modes, and budgets on identical twin states, which is how every result in §V is obtained. *At run time* the identical code path is invoked by the $\Delta\mathcal{T}$ trigger to re-score the twin and re-solve the placement; the results of §V-D are produced in this online configuration, replayed on captured states for reproducibility. Simulation is thus not a substitute for the twin but the mechanism through which it is queried. Accordingly, the sub-3-minute figure at $|V| \approx 235$ measures the wall-clock interval from a $\Delta\mathcal{T}$ event to an available recommendation—delta ingestion and graph rebuild, incremental CVE re-enrichment of the affected nodes, the Monte Carlo re-run that dominates it at 72 s, and the knapsack solve with consolidation—excluding the operators' approval time. Re-optimisation thus completes within one monitoring cycle.

A. Vulnerability and Detection Modelling

Per-CVE probability, EPSS priors, and Noisy-OR aggregation. Rather than relying on arbitrary manual weights, TWIN-IDS derives probabilistic edge weights from real vulnerability data. Given a CVE c affecting a service s, its base exploitation probability is first derived from the CVSS v3.1 FIRST temporal metrics (exploit maturity, remediation level, report confidence). To align this theoretical severity with observed threat activity, the base probability is blended with the Exploit Prediction Scoring System (EPSS) [4] prior through

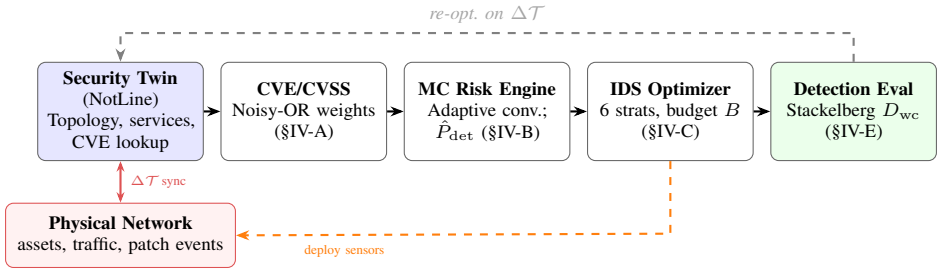

Fig. 1: TWIN-IDS pipeline. The red bidirectional arrow denotes continuous passive synchronisation between NotLine and the physical network (topology discovery, traffic baseline capture, patch events). The dashed orange arrow represents IDS sensor deployment feedback to the physical infrastructure. The dashed gray arrow indicates event-driven re-optimisation on topology changes $\Delta\mathcal{T}$.

the convex combination $p_c^{\text{fin}} = \alpha_E\, p_c^{\text{EPSS}} + (1{-}\alpha_E)\, p_c^{\text{CVSS}}$, with $\alpha_E = 0.70$.

On the semantics of the blend. The two terms are not homogeneous and we do not claim they are: EPSS is a calibrated probability of exploitation in the wild over a 30-day horizon, whereas a CVSS temporal score is an ordinal severity index with no probabilistic reading. We therefore treat p_c^{CVSS} as a monotone severity-to-propensity transform and give it the minority weight, precisely because CVSS scores overestimate exploit likelihood [53]; the residual 0.30 retains sensitivity to impact and attack complexity, which EPSS does not model, and covers CVEs with no EPSS score, where the blend degenerates to the CVSS term alone. The 30-day horizon matches the twin re-synchronisation cycle, so weights and the exploitation window they represent share a time scale. What the blend does *not* yield is an absolutely calibrated per-edge probability; establishing that requires ground-truth exploitation telemetry (§VI). Placement quality is in any case insensitive to the split (≤ 1.2 pp for $\alpha_E \in [0.50, 0.90]$), since risk scoring depends on the *relative ordering* of edge weights rather than their absolute scale; any aggregation preserving that ordering can be substituted.

When multiple vulnerabilities affect the same service, their compound exploitation probability is aggregated using the noisy-OR model [61]. This effectively computes the probability of *at least one* successful exploit, assuming vulnerabilities are exploited independently. Finally, this raw service-level probability is modulated by role- and action-specific multipliers and clamped to $[0.05, 0.98]$ to yield the final edge weight $w(u, v, a)$. The bounds encode two modelling decisions. The lower one denies that an edge can be *impossible*: absence of catalogued vulnerabilities is ignorance, not invulnerability, and zeroing such edges would delete them from the reachability graph, making whole regions invisible to the sampler and biasing risk towards the well-documented part of the network. The upper one denies certainty: a weight-1 edge is taken in every sampled path, collapsing estimator variance, degenerating the maximum-probability adversary into one deterministic route, and letting a single CVE dominate the placement. The values follow the usual practice of bounding attack-graph exploitation probabilities away from $\{0, 1\}$ [49], [52]; being a truncation of the weight distribution, the clamp affects only the extremes of the ordering and, by the argument used for α_E, leaves the induced node ranking essentially unchanged. Any alternative weighting is admissible provided no edge receives probability 0 or 1.

Simulation-derived detection model. For each node v, NotLine provides a d-dimensional baseline traffic profile $\boldsymbol{\mu}(v)$ and variance $\boldsymbol{\sigma}^2(v)$. The feature vector comprises five observables: packet rate, byte rate, connection entropy, protocol-mix KL-divergence, and unique-destination count. Baseline windows are 14 days (university) and 21 days (industrial); windows shorter than 7 days materially raise SCADA false positives (12.4% at 3 days vs. 5.0% at 14 or more).

During simulation run i, if attack a targets v producing feature vector $\mathbf{x}_i$, we define the Anomaly Score AS_i based on the deviation $\boldsymbol{\delta}_i(v, a) = \mathbf{x}_i - \boldsymbol{\mu}(v)$:

$$\text{AS}_i(v, a) = \left\| \frac{\boldsymbol{\delta}_i(v, a)}{\boldsymbol{\sigma}(v)} \right\|_2 \tag{2}$$

AS_i is thus the length of the observed deviation measured in standard deviations of the node's own benign behaviour, which makes scores comparable across nodes with very different traffic volumes. A host-based IDS (HIDS) fires when $\text{AS}_i(v, a) \geq \tau_r$, where τ_r is calibrated *per role class* using the twin's historical benign traffic to achieve a target 5% FPR. This role-adaptive calibration achieves $\hat{\text{FPR}}_r \in [4.7\%, 5.3\%]$ across all role classes, with system-wide F_1 improvement of $+0.6$ pp over a single global threshold. The empirical detection probability is:

$$\hat{P}_{\text{det}}(v, a) = \frac{|\{i \in \mathcal{P}(v, a) : \text{AS}_i(v, a) \geq \tau_r\}|}{|\mathcal{P}(v, a)|} \tag{3}$$

where $\mathcal{P}(v, a)$ is the set of simulations where action a targets v.

Definition 2 (Host Sensor Coverage)**.** *A HIDS at $v \in \mathcal{I}$ monitors all edges incident to v: $C_{\text{h}}(v) = \{(u, v, a) \in E\} \cup \{(v, w, a) \in E\}$, with per-traversal detection probability $\hat{P}_{\text{det}}(v, a)$.*

Given an attack path $\pi = (v_1, a_1), \ldots, (v_L, a_L)$ and sensor

set $\mathcal{I}$, path detection follows the independent-sensor model:

$$\Pr[\text{det.} \mid \pi, \mathcal{I}] = 1 - \prod_{\substack{i=1 \\ v_i \in \mathcal{I}}}^{L} \big(1 - \hat{P}_{\text{det}}(v_i, a_i)\big) \tag{4}$$

Intuitively, an intrusion escapes detection only if it evades every monitored node it traverses; each additional sensor on the path multiplies the evasion probability by a factor smaller than one, so detection grows with path length but with diminishing returns.

B. Monte Carlo Risk Engine

A node's risk is the frequency with which sampled attack paths cross it.

Definition 3 (Node Risk Score)**.** *For adversary mode m and N simulated attack paths:*

$$\hat{\rho}_m^{(N)}(v) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[v \in \pi_i^*] \tag{5}$$

Dual adaptive stopping criterion. Simulation runs in batches of b intrusions (the symbol b is used throughout for the batch size, and is not to be confused with the monetary budget B). After every batch we compute two convergence indicators: the ℓ_∞ drift $\delta^{(k)} = \|\hat{\boldsymbol{\rho}}^{(kb)} - \hat{\boldsymbol{\rho}}^{((k-1)b)}\|_\infty$ and the rank stability $\mathrm{RS}^{(k)}$, the Jaccard similarity of the top-20% riskiest nodes between consecutive batches. Convergence is declared when, for W consecutive batches:

$$k \geq K_{\min} \;\wedge\; \max_{j=k-W+1}^{k} \delta^{(j)} < \varepsilon \;\wedge\; \min_{j=k-W+1}^{k} \mathrm{RS}^{(j)} \geq \theta_{\text{rank}} \tag{6}$$

In words: sampling stops only when *no* node's risk estimate has moved appreciably and the identity of the riskiest nodes has stopped changing, over several consecutive batches—the first condition guards the values used by the knapsack, the second guards the ranking used by the greedy strategies. We use $W{=}3$: it yields a 0.9% false-convergence rate with median $N_{\text{adapt}}{=}4{,}200$ (72 s at $|V|{=}235$), against 8.3% for $W{=}2$ and a 31% runtime penalty for $W{=}5$. A verification batch after tentative convergence guards against plateau artefacts.

The engine estimates $\hat{P}_{\text{det}}(v, a)$ *within* the same pass: whenever a path targets a node, it scores the deviation against the twin-captured baseline and updates the detection counter. This removes a separate calibration phase and keeps detection estimates consistent with the edge-weight model.

C. Cost-Aware IDS Placement

Given a total monetary budget B (in monetary units, MU), the placement problem is a budget-constrained binary optimisation:

$$\max_{\mathbf{x} \in \{0,1\}^{|V|}} \hat{D}(\mathcal{I}) \quad \text{s.t.} \quad \sum_{v \in V} \kappa(v)\, x_v \leq B \tag{7}$$

where $x_v \in \{0, 1\}$ indicates if a sensor is placed at v: the defender buys the subset of monitoring locations whose expected detection is highest and whose total price fits the budget. The detection rate is the average, over the simulated intrusions, of the per-path detection probability:

$$\hat{D}(\mathcal{I}) = \frac{1}{N_{\text{adapt}}} \sum_{i=1}^{N_{\text{adapt}}} \Pr[\text{det.} \mid \pi_i, \mathcal{I}] \tag{8}$$

We implement five greedy placement strategies (RA, DEG, BET, PR, RND) that share a budget-exhaustion loop and one exact solver.

Exact 0/1 knapsack (KNP). The path-level objective of Eq. 4 is non-additive, so the dynamic program uses per-node risk scores $\hat{\rho}_m(v)$ as additive item values—surrogates for path-visitation frequency under mode m—and the selected placement is then re-evaluated against the true $\hat{D}(\mathcal{I})$ (Eq. 8). Costs are discretised [5], [62] as $c_v = \lceil \kappa(v)/\delta \rceil$, $W = \lfloor B/\delta \rfloor$.

The next proposition bounds the loss from optimising against Monte Carlo estimates rather than the exact objective. It is a *sample-complexity* statement: it compares the placement chosen from N simulated paths with the one that the exact path distribution *of the same model* would select, and says nothing about the fidelity of that model to the physical network, which is an empirical question addressed in §V. Empirical optimality within the twin does not imply true optimality in the field.

Proposition 1 (Sample Optimality)**.** *Let $\hat{\mathcal{I}}^*$ and $\mathcal{I}^*$ denote, respectively, the placement that maximises the empirical detection estimate and the placement that maximises its expectation under the model. For any $\delta_c \in (0, 1)$, with probability $\geq 1{-}\delta_c$: $D(\hat{\mathcal{I}}^*) \geq D(\mathcal{I}^*) - 2\varepsilon_N$, where $\varepsilon_N = \sqrt{(\ln(2|\Pi_B|) + \ln(1/\delta_c))/(2N)}$.*

Proof. By Hoeffding's inequality and a union bound over the set of all budget-feasible placements Π_B [63]. □

The empirical cross-validation gap of 0.6 pp confirms a negligible over-fitting.

D. Network IDS Extension and Hybrid Deployment

A network IDS (NIDS) deployed at the gateway g_k of segment $\mathcal{S}_k$ captures all Layer-3 traffic across g_k: $C_{\text{n}}(\mathcal{S}_k) = \{(u, v, a) \in E : u \in \mathcal{S}_k \vee v \in \mathcal{S}_k\}$.

Observing aggregate flows, a NIDS sees only three of the five features (packet rate, byte rate, connection entropy), which costs a median 4.1 pp of $\hat{P}_{\text{det}}^{\text{net}}$ (7.8 pp on SCADA gateways); per-flow scoring suits high-volume IT segments and aggregate scoring low-volume OT segments, where per-flow signal dilutes [46]. The hybrid cost model is $\kappa_{\text{net}}(\mathcal{S}_k) = \kappa_{\text{tap}} + \beta \max_{v \in \mathcal{S}_k} \kappa(v)$ ($\kappa_{\text{tap}}{=}200$ MU, $\beta{=}1.5$).

Definition 4 (Segment Consolidation)**.** *Segment $\mathcal{S}_k$ is* consolidable *w.r.t. placement $\mathcal{I}$ iff (C1) $\kappa_{\text{net}}(\mathcal{S}_k) < \sum_{v \in V_k'} \kappa(v)$, where $V_k' = \mathcal{I} \cap \mathcal{S}_k$; and (C2) $\hat{D}(\mathcal{I}', m) \geq \hat{D}(\mathcal{I}, m) - \epsilon_D$ for all $m \in \mathcal{M}$, where $\mathcal{I}' = (\mathcal{I} \setminus V_k') \cup \{g_k\}$ and $\epsilon_D = 0.01$.*

Consolidation is sound because a segment NIDS dominates any union of host sensors inside the segment: for $V' \subseteq \mathcal{S}_k$, $C_{\text{n}}(\mathcal{S}_k) \supseteq \bigcup_{v \in V'} C_{\text{h}}(v)$, since every edge incident to some $v' \in V' \subseteq \mathcal{S}_k$ already satisfies the disjunctive condition defining C_{n}. Coverage can therefore only be lost through

the weaker per-flow visibility of the gateway, which is exactly what condition (C2) tests. The post-processor runs in $O(|\mathcal{K}| \cdot |\mathcal{M}| \cdot N_{\text{adapt}})$ time, reusing existing simulation paths. Freed budget is reinvested via a single-item knapsack update.

The hybrid detection integrates HIDS and NIDS contributions:

$$P_i^{\text{eff}} = \begin{cases} \hat{P}_{\text{det}}(v_i, a_i) & v_i \in V_{\text{HIDS}}, \\ \hat{P}_{\text{det}}^{\text{net}}(g_k, a_i) & v_i \in \mathcal{S}_k,\ g_k \in V_{\text{NIDS}}, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

This effective probability replaces $\hat{P}_{\text{det}}$ in Eq. 4 when evaluating the consolidated hybrid deployment.

E. Stackelberg Worst-Case Formulation

Game formulation. The defender (leader) commits to a placement $\mathcal{I}$; the attacker (follower) best responds by choosing the traversal mode that minimises detection:

$$\mathcal{I}^* = \arg\max_{\mathcal{I} \in \Pi_{\mathcal{D}}} \min_{m \in \mathcal{M}} D(\mathcal{I}, m) \triangleq D_{\text{wc}}(\mathcal{I}) \tag{10}$$

Multi-mode risk scoring. Rather than optimising against a single adversary, we score nodes using a weighted mixture of mode-specific risk: $\rho_{\text{mm}}(v) = \sum_{m \in \mathcal{M}} \omega_m \cdot \hat{\rho}_m^{(N)}(v)$, $\sum_m \omega_m = 1$. Mode weights are derived from MITRE ATT&CK v14 [64] (576 techniques, 196,042 procedure references): ω_{Rnd}=0.30, ω_{Mpp}=0.25, ω_{Stlth}=0.35, ω_{Nsy}=0.10. De-biased weights via inverse-detectability yield $(0.26, 0.22, 0.43, 0.09)$. These weights are frequency estimates over a cross-sector corpus, not universal constants: a defender facing a specific threat profile should re-estimate them from the relevant ATT&CK groups, which changes nothing in the framework since ω enters only as a mixture weight. Two properties nonetheless make them a safe default. Mixing is itself conservative—it prevents over-fitting to whichever single adversary the analyst assumes—and the worst-case objective is evaluated against *every* mode irrespective of ω, so a misspecified ω degrades the mixture but cannot silently drop an adversary. Empirically, D_{wc} varies by at most 1.8 pp across five weighting schemes including the uniform one, leaving the ordering of strategies unchanged (§V-C).

LP-based SSE validation. Following [8], we solve the Strong Stackelberg Equilibrium LP:

$$\begin{aligned} \max_{\mathbf{q}, z} \quad & z \\ \text{s.t.} \quad & \sum_{\mathcal{I}} q_{\mathcal{I}} D(\mathcal{I}, m) \geq z \quad \forall m \in \mathcal{M} \\ & \mathbf{q} \geq \mathbf{0},\ \ \mathbf{1}^\top \mathbf{q} = 1 \end{aligned} \tag{11}$$

Because enumerating all $|\Pi_B|$ placements is intractable beyond small instances, we validate via column generation on $|V| \leq 50$ instances [8].

V. Experimental Evaluation

Four research questions structure the evaluation. **RQ1** (*cost-efficiency*): how does twin-based, cost-aware, CVE-grounded placement compare with graph-centrality and non-CVE baselines, and how sensitive is it to pipeline parameters? **RQ2** (*adversarial robustness*): how does it hold up against adaptive adversaries under worst-case analysis? **RQ3** (*dynamic adaptability*): can it reconfigure effectively after zero-day injections and node failures? **RQ4** (*generalisation*): does it transfer across structurally divergent real OT/IT topologies?

A. Setup and Baselines

Implementation and instances. Python 3.11 with NetworkX 3.2 on an Intel Xeon w7-3465X (1 TB RAM, Ubuntu 24.04), all seeds fixed; b=500, $K_{\min}$=4, W=3, ε=0.005, r_v=0.20, θ_{rank}=0.95, δ=100 MU. Three twins are used: a synthetic enterprise twin ($|V|\approx$235, $C_{\text{tot}}\approx$\$248k), a university twin ($|V|$=126, 14-day passive coverage, \$142.4k), and an industrial automation twin ($|V|$=202, 21-day coverage at 92% of nodes, \$389.5k, 53% OT).

Baselines. Six placement strategies: our risk-aware greedy (RA), degree (DEG), betweenness (BET), PageRank (PR), random (RND), and exact knapsack (KNP). Two external baselines: LIU (Liu et al. [42], synthetic weights, node-count budget) and LIU-CVE (fair variant with CVE-grounded weights under monetary budget), plus MXCOV (maximum-coverage heuristic).

Detection model validation. We first check that the simulation-derived $\hat{P}_{\text{det}}$ is a faithful proxy for realistic IDS behaviour. The ATTACKWALK sampler generated 2,000 successful intrusion traces under the `Mpp` adversary—sequences of feature-deviation vectors $\boldsymbol{\delta}_i(v, a)$ (Eq. 2) against twin-captured baselines—and 5,000 benign traces from the same baseline windows. SNORT uses the Snort3-community set (1,247 ATT&CK-mapped rules); IFOR uses 100 trees, contamination=0.05, and the same five features. Table II reports per-role recall at 5% FPR. **$\hat{P}_{\text{det}}$ falls within 2.3 pp of IFOR recall on all role classes** (max 3.1 pp, SCADA gateways). SNORT is stronger on well-documented patterns (SMB lateral movement, +8.1 pp) and markedly weaker on encrypted pivoting (VPN/RDP, −9.3 pp), so the simulated scores favour neither the signature- nor the anomaly-based paradigm.

False negatives. Since a missed intrusion costs far more than a false alarm, Table II must also be read as a false-negative rate, $\text{FNR}_r = 1 - \text{recall}_r$: at the calibrated 5% FPR it ranges from 0.18 on firewall/DNS nodes to 0.57 on VPN/RDP and 0.62 on SCADA gateways—the assets that matter most, and whose sparse traffic gives an anomaly detector least to work with. Hence no single sensor should be expected to stop an intrusion, which is precisely the argument for placing sensors along *paths*: by Eq. 4 an intrusion crossing three monitored nodes with FNR=0.5 each escapes with probability 0.125. The 5% target is moreover a defender-chosen operating point exposed by τ_r (Eq. 3), not a property of the framework. At the placement level the corresponding quantity is $1 - D_{\text{wc}}$, the share of worst-case intrusions reaching the target undetected: 0.59 for RA at 5% budget, 0.52 under ATT&CK-weighted scoring (§V-C).

B. RQ1: Cost-Efficiency and Baseline Comparison

Sensor count and budget efficiency. RA reaches 80% detection with 5 sensors against 11 for RND on the small twin

TABLE II: Detection model validation: per-role recall at 5% FPR; $\text{FNR}_r = 1 - \text{recall}_r$.

Role class	TWIN-IDS	SNORT	IFOR	Δ_{IF}
Firewall / DNS	0.82	0.79	0.80	+0.02
DB primary	0.74	0.76	0.73	+0.01
Workstation	0.71	0.75	0.70	+0.01
Web server	0.68	0.71	0.67	+0.01
App / auth server	0.65	0.68	0.64	+0.01
Mail relay	0.61	0.63	0.60	+0.01
VPN / RDP gateway	0.43	0.33	0.41	+0.02
SCADA gateway	0.38	0.39	0.35	+0.03
System F_1 (weighted)	0.816	0.798	0.804	+0.012

TABLE III: Comparing all strategies on the enterprise twin ($|V| \approx 235$).

Strategy/Budget	**1%**	**2%**	**5%**	**10%**	**15%**	**20%**
RA	0.18	0.30	0.58	0.73	0.84	0.91
LIU-CVE	0.12	0.22	0.46	0.62	0.77	0.87
BET	0.11	0.20	0.40	0.59	0.75	0.85
MXCOV	0.10	0.19	0.39	0.58	0.74	0.86
LIU	0.09	0.17	0.39	0.56	0.73	0.86
DEG	0.09	0.16	0.34	0.53	0.70	0.80
RND	0.05	0.10	0.26	0.43	0.59	0.72
RA vs. LIU-CVE	+6pp	+8pp	+12pp	+11pp	+7pp	+4pp
LIU-CVE vs. LIU	+3pp	+5pp	+7pp	+6pp	+4pp	+1pp

($|V|$=31, -55%) and 22 against 41 on the large one ($|V|$=218, -46%). Against monetary budget (Fig. 2), **RA dominates at every level**: at 5% it attains $\hat{D}$=0.58, +18 pp over BET and +32 pp over RND. The margin peaks in the 2–10% range—the regime most enterprises actually operate in [59], [60]—and closes by 20%, where budget saturation makes most placements viable.

Knapsack vs. greedy. KNP reaches $\hat{D}$=0.600 at 5% budget (+3.4% over RA), with the gap largest at low budgets (3.3% at 2%) and vanishing at high ones (0.8% at 15%). RA is thus a competitive fallback where the $O(|V| \cdot W)$ table exceeds memory: at $|V|$=10,000 this occurs above 0.75% budget, at a median cost of 1.9 pp.

Algorithmic decomposition against Liu et al. The LIU-CVE fair baseline—Liu et al.'s degree ranking with our CVE edge weights under a monetary budget—separates the two sources of advantage (Table III). **At 5% budget LIU-CVE reaches 0.46, i.e. +7 pp over LIU from CVE-grounded weights alone, and the residual +12 pp up to RA's 0.58 is attributable purely to path-level MC scoring**, showing the two ingredients contribute independently.

Parameter sensitivity. The DBIR-calibrated entry-point distribution is worth +7.3 pp over a uniform one, whereas the EPSS blend is nearly immaterial ($\leq$ 1.2 pp for $\alpha_E \in [0.50, 0.90]$): *where* intrusions start matters more than *how* severity and exploitability are mixed. The F_1-optimal operating point θ^*=0.72 yields F_1^*=0.816 (precision 0.847, recall 0.788).

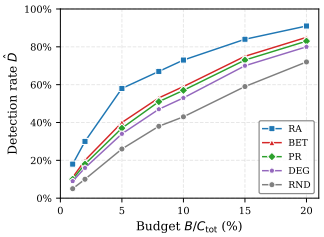

Fig. 2: Detection rate vs. monetary budget (enterprise twin). RA dominates at every budget; the largest gap occurs at 5%.

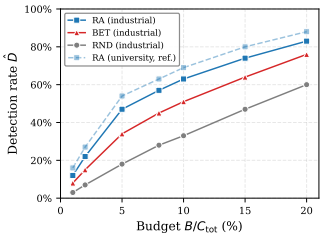

Fig. 3: Industrial vs. university twin. RA advantage narrows in OT-heavy topologies (f_{OT}=0.53).

TABLE IV: Per-mode detection and Stackelberg D_{wc} (enterprise twin, 5% budget).

Strategy	**Rnd**	**Mpp**	**Stlth**	**Nsy**	D_{wc}
RA	0.58	0.55	0.41	0.62	0.41
BET	0.40	0.38	0.34	0.43	0.34
PR	0.37	0.35	0.31	0.41	0.31
DEG	0.34	0.32	0.28	0.37	0.28
RND	0.26	0.24	0.20	0.27	0.20
KNP	**0.61**	**0.58**	**0.44**	**0.65**	**0.44**

C. RQ2: Adversarial Robustness

Table IV reports per-mode detection and the Stackelberg worst-case floor D_{wc} at 5% budget on the enterprise twin.

RA achieves D_{wc}=0.41, which is 20.6% above BET and 105% above RND. KNP further lifts D_{wc} to 0.44 (+7.3%). The stealth mode is consistently the binding constraint: it achieves the lowest detection across all strategies. This confirms that the hardest adversary models to counter are those that minimise detection exposure.

Multi-mode scoring. ATT&CK-weighted mixture achieves D_{wc}=0.48, a **17.1% improvement** over single-mode RA (D_{wc}=0.41), because it concentrates sensors on the paths followed by the stealth attacker; inverse-detectability de-biasing lifts it further to 0.50.

LP-SSE validation. On instances with $|V| \leq 50$, KNP closes 87% of the SSE gap. The residual gap grows only mildly with instance size, from a median Δ_{SSE} of 0.021 ($|V|$=20) to 0.031 ($|V|$=50) for KNP, and from 0.014 to 0.019 for ATT&CK-weighted RA, whose p95 never exceeds 0.041. A deterministic commitment to a single knapsack placement is therefore a close approximation of the mixed-strategy equilibrium.

D. RQ3: Dynamic Adaptability

Starting from $\hat{D}$=0.58 (RA, 5% budget, enterprise twin), injecting five zero-day edges drops detection to 0.46 and re-optimisation restores it to 0.54, recovering **72.7%** of the loss; failing the top-5 nodes drops it to 0.50 and recovery reaches 0.56 (**75.0%**). Under combined disruption detection falls to 0.41 and recovers to 0.51 (60.9%), the lower figure reflecting the compounding of simultaneous topology and vulnerability change.

TABLE V: Topological properties of real-network instances.

Twin	$\|V\|$	f_{OT}	ρ_{int}	$\bar{d}$	CVSS mean	CVE/node
Enterprise (synth.)	235	0.11	0.018	3.1	7.1	3.2
University (real)	126	0.09	0.031	3.0	6.9	2.9
Industrial (real)	202	0.53	0.009	2.8	8.2	5.7

Scalability and computational overhead. The full pipeline runtime follows $T = 0.0031|V|^{1.12}$ (R^2=0.998), dominated by the MC risk engine. At $|V|$=10,000, the entire pipeline completes in 418 ± 12 s. The overhead relative to the baselines is structural and should be stated plainly: DEG, BET, PR and MXCOV each require a single pass over the graph—$O(|V|+|E|)$ for degree, $O(|V||E|)$ for betweenness—and terminate in seconds even on the largest instance, whereas TWIN-IDS additionally pays for N_{adapt} sampled paths and, for KNP, an $O(|V| \cdot W)$ dynamic program. The relevant question is therefore not whether the framework is more expensive than a centrality ranking, but whether its absolute cost is compatible with the deployment cycle it serves; the sub-3-minute figure at $|V|$=235 and the sub-7-minute figure at $|V|$=10,000 indicate that it is, given that the placement it produces is enacted over days and re-optimised on topology events rather than continuously. The end-to-end re-optimization of the enterprise security twin ($|V|$=235), takes less than 3 min, enabling reactive posture adjustment within a single monitoring cycle.

NIDS consolidation. Hybrid placement reaches $\hat{D}$=0.627 against 0.580 for HIDS-only at 5% budget (+4.9 **pp at identical cost**) and lifts D_{wc} from 0.410 to 0.442; 8 of 17 candidate segments satisfy both conditions, freeing 3,300 MU at a detection cost of 0.6 pp.

E. RQ4: Cross-Domain Generalisation

We evaluate TWIN-IDS on two real operational security twins with markedly different topological characteristics (Table V).

On the **university twin** ($|V|$=126), RA needs 44% fewer sensors than RND for 70% detection and KNP attains D_{wc}=0.40 (+8.1% over RA); column generation converges at z^*_{SSE}=0.431, so Δ_{SSE}=0.011—below the p95 of the $|V|$=50 distribution, which confirms the approximation quality on a real topology. On the **industrial twin** ($|V|$=202, Fig. 3), RA remains superior but **its margin over BET narrows from** +18 **to** +13 **pp at 5% budget**: with f_{OT}=0.53 the risk distribution is more uniform and risk-based ranking discriminates less. Here KNP gives D_{wc}=0.34 and ATT&CK RA 0.37, with z^*_{SSE}=0.389 and Δ_{SSE}=0.019 after 4,312 columns.

Cross-twin baseline comparison. Table VI reports detection rates for all strategies on both twins. The algorithmic decomposition holds across domains: **RA vs. LIU-CVE ranges from** +9 **pp (industrial) to** +11 **pp (university)**, while LIU-CVE vs. LIU contributes +3 to +5 pp from CVE weight quality. The compression of the algorithmic gap in the industrial setting may be due to the higher internal-tier density (ρ_{int}) and the larger OT fraction, which flattens the risk landscape and bounds the marginal value of path-level MC scoring.

TABLE VI: External baselines on both security twins (5% and 10% budget).

Strategy	University ($\|V\|$=126)		Industrial ($\|V\|$=202)	
	5%	10%	5%	10%
RA	0.54	0.69	0.47	0.63
LIU-CVE	0.43	0.58	0.38	0.54
LIU	0.40	0.57	0.33	0.49
MXCOV	0.38	0.55	0.36	0.52
RND	0.22	0.40	0.18	0.33
RA vs. LIU-CVE	+11pp	+11pp	+9pp	+9pp
LIU-CVE vs. LIU	+3pp	+1pp	+5pp	+5pp

Noisy-OR sensitivity. Across 412 dependent CVE pairs in 37 service instances (NVD CWE chains [65]), relaxing independence moves detection by only 0.8 pp on the enterprise twin (placement Jaccard 0.921) and 1.4 pp on the industrial one (0.871); above f_{OT}=0.40 we therefore recommend the Bayesian chain correction as the default.

VI. DISCUSSION

What the results say. Risk-aware, CVE-grounded placement dominates every baseline across the budget spectrum, +7 pp of the margin coming from weight quality and +12 pp from path-level scoring (RQ1); ATT&CK weighting lifts the worst-case floor by 17.1% with an SSE gap under 0.019 (RQ2); re-optimisation recovers 61–75% of disruption-induced loss within one monitoring cycle (RQ3); and the advantage compresses to +9 pp in OT-heavy topologies, whose flatter risk landscape leaves path-level scoring less to exploit (RQ4).

Ecological validity and practice. Evaluating on a university campus (f_{OT}=0.09) and a manufacturing plant (f_{OT}=0.53) gives better ecological validity than studies confined to synthetic topologies [41]–[43], and yields a usable rule: above f_{OT}=0.40 expect a smaller return from path-level scoring and prefer the Bayesian chain correction [49], [52] to plain noisy-OR. Operationally, the sub-3-minute latency fits continuous security operations, consolidation frees 3,300 MU for 0.6 pp of detection, and the KNP boundary at $|V| \approx 7{,}500$ covers most enterprises. Finally, since concentrating sensors where high-probability paths intersect is structurally the criterion of optimal honeypot placement [66], [67], TWIN-IDS risk scores can seed a joint detection-and-deception placement.

Modelling assumptions. Four assumptions delimit the applicability of the framework, and we state the direction of the bias each introduces. (A1) *Vulnerability independence.* Noisy-OR treats the CVEs of a service as independently exploitable, whereas real vulnerabilities correlate through shared libraries and CWE chains; aggregate exploitation is therefore over-estimated where correlated CVEs concentrate, most notably in OT-heavy topologies with the highest CVE density (5.7 vs. 3.2, Table V), and is partly corrected by the Bayesian chain variant recommended for $f_{\mathrm{OT}} > 0.40$. (A2) *Sensor independence.* Eq. 4 multiplies per-node evasion probabilities, yet sensors sharing a detection principle fail together against a technique built to defeat it; path detection is therefore an optimistic bound, mitigable by diversifying detection technology along a path, which our cost model does not yet express. (A3) *Static adversary within a cycle.* The attacker's mode is fixed during optimisation, whereas real

adversaries learn a deployed configuration over time; D_{wc} protects against the worst mode in $\mathcal{M}$, not against one invented in response to the placement. (A4) *Complete information.* The follower observes the placement exactly. This is conservative—an attacker knowing less does no better, so D_{wc} is a lower bound—at the price of a posture possibly over-defensive against less-informed adversaries.

Limitations. The principal limitation is one of evaluation scope. Although both real twins are built from operational networks, the detection probabilities they yield are derived from simulated intrusions scored against twin-captured baselines, not from alerts observed in a production sensor deployment; the agreement within 2.3 pp with SNORT and Isolation Forest establishes that the simulated scores are a plausible proxy, not that they predict field performance. Validation against operational IDS logs, and ultimately a deployment study in which recommended sensors are installed and their alerts compared against the predicted $\hat{P}_{\mathrm{det}}$, is the natural next step and the one that would most strengthen the practical claim. Two further limitations are technical: benign baselines may not capture seasonal or production-schedule variability, so the 5% false-positive target may need re-tuning in large SOCs; and the exact SSE and KNP solvers remain confined to $|V| \leq 50$ and $|V| \lesssim 7{,}500$, beyond which the greedy fallback RA costs a median 1.9 pp.

Future work. We plan to scale exact Stackelberg computation to $|V| \approx 500$ via double-oracle methods [56], [57], to replace the static adversary set with reinforcement-learning-based co-adaptive attackers and continuous sensor re-placement [68], and to extend the Bayesian compound-chain model [69] for OT environments with $f_{\mathrm{OT}} > 0.40$.

VII. CONCLUSION

We presented TWIN-IDS, a framework that places intrusion detection sensors under a monetary budget by closing the loop around a live security digital twin. Its contribution is one of integration rather than of new primitives: an established vulnerability-aggregation pipeline, an adaptive Monte Carlo risk engine, an exact knapsack solver, and a standard worst-case game formulation are composed under a single probabilistic semantics, so that CVE fidelity, cost heterogeneity, topology dynamics, and adversary adaptivity are addressed together rather than in isolation. Empirically, weight quality contributes $+7$ pp and path-level Monte Carlo scoring $+12$ pp ($+9$ pp in OT-heavy domains), ATT&CK-weighted scoring lifts the worst-case floor by 17.1% with $\Delta_{\mathrm{SSE}} \leq 0.019$, and re-optimisation recovers 61–75% of disruption-induced loss in under three minutes—all obtained without perturbing the production network, since the twin absorbs the simulation load. A field validation against operational sensor telemetry is the natural next step.

REPRODUCIBILITY AND DATA AVAILABILITY

Everything needed to reproduce the paper is at https://github.com/ilsamaritano/TWIN-IDS: the implementation, one configuration file fixing every parameter of §V-A (seeds included), the anonymised university and industrial twin exports, the synthetic topology generators, and driver scripts regenerating every table and figure. The vulnerability snapshot ships as a resolved CVE-to-service mapping with its retrieval date, since re-querying the live catalogue and EPSS later yields different scores by design. Identifiers and service banners are pseudonymised; topology, role, layer, and cost attributes are preserved.

REFERENCES

[1] A. Clark, Q. Zhu, R. Poovendran, and T. Basar, "Deceptive routing in relay networks," *Lecture Notes in Computer Science (GameSec)*, vol. 7638, pp. 171–185, 2012.

[2] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," in *Proc. ACM CCS*, 2006, pp. 336–345.

[3] FIRST.Org, Inc., "Common vulnerability scoring system v3.1 specification document," Forum of Incident Response and Security Teams (FIRST), Tech. Rep., 2019. [Online]. Available: https://www.first.org/cvss/specification-document

[4] J. Jacobs, S. Romanosky, I. Adjerid, and W. Baker, "Improving vulnerability remediation through better exploit prediction," vol. 6, no. 1, 2020, p. tyaa015.

[5] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer, 2004.

[6] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *Proc. USENIX LISA*, 1999, pp. 229–238.

[7] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, pp. 1–39, 2012.

[8] V. Conitzer and T. Sandholm, "Computing the optimal strategy to commit to," in *Proc. 7th ACM Conf. Electronic Commerce (EC)*, 2006, pp. 82–90.

[9] M. Eckhart and A. Ekelhart, "Towards security-aware virtual environments for digital twins," in *Proc. ACM Workshop on Cyber-Physical Systems Security and Privacy (CPS-SPC)*, 2018, pp. 61–72.

[10] ——, "A specification-based state replication approach for digital twins," *Proc. ACM Workshop on Cyber-Physical Systems Security & Privacy*, pp. 36–47, 2019.

[11] C. Alcaraz and J. Lopez, "Digital twin: A comprehensive survey of security threats," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1475–1503, 2022.

[12] R. Bitton, T. Gluck, O. Stan, M. Inokuchi, Y. Ohta, Y. Yamada, T. Yagyu, Y. Elovici, and A. Shabtai, "A taxonomy of digital twins for cyber-physical systems security," in *Proc. European Symposium on Research in Computer Security (ESORICS) Workshops*, 2018, pp. 270–284.

[13] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.

[14] M. Dietz and G. Pernul, "Digital twin: Empowering enterprises towards a system-of-systems approach," *Business & Information Systems Engineering*, vol. 62, no. 2, pp. 179–184, 2020.

[15] R. Khan, M. Z. Islam, S. Y. Siddiqui, M. A. Khan, J. Ahmad *et al.*, "Artificial intelligence-enabled digital twin framework for cybersecurity risk assessment," in *Proc. IEEE International Conference on Cyber Security and Protection of Digital Services*, 2022, pp. 1–8.

[16] F. Baiardi, V. Sammartino, and S. Ruggieri, "NotLine: A non-intrusive automated platform to build a digital twin," in *2025 29th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 2025, pp. 1–8.

[17] F. Baiardi, S. Ruggieri, and V. Sammartino, "Anticipating disasters through a security twin," in *Dynamics of Disasters: Hybrid Threats*. Springer, 2026, pp. 1–14.

[18] F. Baiardi and V. Sammartino, "Simulation-powered cybersecurity: Real-time risk assessment via non-intrusive security twin," *The Journal of Supercomputing*, vol. 82, no. 5, p. 311, 2026.

[19] F. Baiardi, V. Sammartino, and S. Ruggieri, "Evaluating adversary strategies through a security twin," in *2026 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2026, pp. 1–7.

[20] V. Sammartino, "Vulnerabilities in autonomous execution: A survey of security threats and defenses in LLM-driven multi-agent systems," Available at SSRN 7218206.

[21] F. Baiardi and V. Sammartino, "A quantitative framework for the validation of twin-based cyber defense," *Procedia Computer Science*, vol. 274, pp. 721–730, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050925037913

[22] F. Baiardi, S. Ruggieri, and V. Sammartino, "AI-enabled cybersecurity using synthetic data," in *2025 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2025, pp. 140–145.

[23] V. Sammartino, "A framework for proactive cyber-resilience: Non-intrusive modeling for autonomous defense," in *2025 29th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. IEEE, 2025.

[24] F. Baiardi, S. Ruggieri, and V. Sammartino, "A security twin to defeat intrusions in cyber physical systems," in *Proceedings of the 32nd European Safety and Reliability Conference (ESREL)*, 2025.

[25] V. Sammartino and M. Schneider, "Model-driven security analysis of SD-Access fabrics using digital twins: A probabilistic framework in software-defined campus and cyber-physical networks," *Future Generation Computer Systems*, p. 108698, 2026.

[26] F. Baiardi, V. Sammartino, and S. Ruggieri, "Graph-based cyber defence using a security twin," in *29th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. IEEE, 2025.

[27] F. Baiardi and V. Sammartino, "From digital twins to AI agents: A synthetic data paradigm for next-generation cybersecurity," in *Artificial Intelligence in Cybersecurity: Unlocking the Power of Large Language Models*, 2026.

[28] V. Sammartino and M. Pasquini, "HAVE: Host active verification engine for closing the contextual reality gap in security digital twins," arXiv preprint arXiv:2606.06968, 2026.

[29] V. Sammartino, "Q-FE: A quantum-native 6G far-edge architecture securing industrial IoT digital twins via CSIDH-PQC and asynchronous federated learning," arXiv preprint arXiv:2606.03611, 2026.

[30] ——, "SA-DTS: Semantic-aware digital twin synchronization over 6G networks," arXiv preprint arXiv:2606.03617, 2026.

[31] F. Baiardi and V. Sammartino, "Quantifying the impact of CVSS score ordering on attack paths," in *EAI GOODTECHS 2026 – 12th EAI International Conference on Smart Objects and Technologies for Social Good*. EAI, 2026.

[32] ——, "Quantifying resilience of cyber-physical systems to zero-day threats: A digital twin-based what-if analysis," in *Proceedings of the 36th European Safety and Reliability Conference (ESREL)*, 2026.

[33] ——, "Quantifying resilience of cyber-physical systems to zero-day threats: A security twin-based what-if analysis framework," in *Proceedings of the 36th European Safety and Reliability Conference (ESREL)*, 2026.

[34] V. Sammartino, "Hybrid quantum graph neural networks for robust botnet detection in modern IoT ecosystems," *Future Generation Computer Systems*, p. 108650, 2026.

[35] F. Baiardi and V. Sammartino, "Quantifying cyber robustness and financial risk via security twins," Available at SSRN 6890389, 2026.

[36] V. Sammartino and F. Baiardi, "Discovering when a security twin cheats: An equivalence theorem," Available at SSRN 7277218.

[37] V. Sammartino, F. Baiardi, and S. Ruggieri, "DOS classification, and virtual machine detection via passive monitoring through security twins," in *IEEE COINS 2026*. IEEE, 2026.

[38] V. Sammartino, "NotLine: Dynamic network topology and risk assessment through passive discovery," FAIR General Conference, Rome, Italy, 2025.

[39] F. Baiardi, S. Ruggieri, and V. Sammartino, "Security twins e il futuro della previsione di intrusioni cyber," *ICT Security Magazine*, 2025.

[40] A. K. Pandey, V. Sammartino, A. Vangala, and A. K. Das, "Blockchain-enabled quantum-resistant hybrid isogeny-based signcryption framework for digital twin-assisted healthcare system," *IEEE Communications Standards Magazine*, 2026.

[41] B. Hyder, S. S. Gokhale, and R. Poovendran, "Network intrusion detection system placement for attack detection at a system level," *Computer Communications*, vol. 106, pp. 58–67, 2017.

[42] X. Liu, H. Zhao, M. Liu, and C. Cui, "Optimal placement of heterogeneous intrusion detection systems for mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1858–1871, 2018.

[43] R. Colbaugh and K. Glass, "Predictability-oriented defense against adaptive adversaries," in *Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2012, pp. 2721–2727.

[44] R. Dewri, N. Poolsappasit, I. Ray, and D. Whitley, "Optimal security hardening using multi-objective optimization on attack tree models of networks," in *Proc. ACM Conference on Computer and Communications Security (CCS)*, 2007, pp. 204–213.

[45] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1–2, pp. 18–28, 2009.

[46] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Proc. NDSS*, 2018.

[47] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, "An empirical evaluation of entropy-based traffic anomaly detection," *Proc. ACM IMC*, pp. 151–156, 2008.

[48] A. Ghosh, D. Ditale, M. Albanese, and P. Mukherjee, "Optimizing ids rule placement via set covering with capacity constraints," *Computers & Security*, vol. 161, p. 104748, 2026.

[49] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia, "An attack graph-based probabilistic security metric," in *Proc. IFIP Annual Conference on Data and Applications Security and Privacy (DBSec)*, 2008, pp. 283–296.

[50] J. Homer, S. Zhang, X. Ou, D. Schmidt, Y. Du, S. R. Rajagopalan, and A. Singhal, "Aggregating vulnerability metrics in enterprise networks using attack graphs," in *Journal of Computer Security*, vol. 21, no. 4, 2013, pp. 561–597.

[51] K. Ingols, R. Lippmann, and K. Piwowarski, "Practical attack graph generation for network defense," in *Proc. ACSAC*, 2006, pp. 121–130.

[52] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using Bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 61–74, 2012.

[53] L. Allodi and F. Massacci, "Comparing vulnerability severity and exploits using case-control studies," *ACM Transactions on Information and System Security*, vol. 17, no. 1, pp. 1–20, 2014.

[54] M. Tambe, "Security and game theory: Algorithms, deployed systems, lessons learned," *Cambridge University Press*, 2011.

[55] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed ARMOR protection: The application of a game theoretic model for security at the Los Angeles International Airport," in *Proc. International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2008, pp. 125–132.

[56] B. An, M. Tambe, F. Ordonez, E. Shieh, and C. Kiekintveld, "Refinement of strong Stackelberg equilibria in security games," in *Proc. AAAI Conference on Artificial Intelligence*, 2011, pp. 587–593.

[57] M. Jain, D. Korzhyk, O. Vaněk, V. Conitzer, M. Pěchouček, and M. Tambe, "A double oracle algorithm for zero-sum security games on graphs," in *Proc. AAMAS*, 2011, pp. 327–334.

[58] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe, "Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness," in *Journal of Artificial Intelligence Research*, vol. 41, 2011, pp. 297–327.

[59] W. Lee, W. Fan, M. Miller, S. J. Stolfo, and E. Zadok, "Toward cost-sensitive modeling for intrusion detection and response," *Journal of computer security*, vol. 10, no. 1-2, pp. 5–22, 2002.

[60] Verizon Business, "2024 Data Breach Investigations Report," https://www.verizon.com/business/resources/reports/dbir/, 2024.

[61] J. Pearl, "Probabilistic reasoning in intelligent systems: Networks of plausible inference," in *Morgan Kaufmann*, 1988.

[62] S. Martello, D. Pisinger, and P. Toth, "Dynamic programming and strong bounds for the 0–1 knapsack problem," *Management Science*, vol. 45, no. 3, pp. 414–424, 1999.

[63] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[64] MITRE Engenuity, "ATT&CK evaluations: Enterprise 2024 results," MITRE Engenuity, Tech. Rep., 2024. [Online]. Available: https://attackevals.mitre-engenuity.org/results/enterprise

[65] National Vulnerability Database, "CWE weakness chaining data, NVD catalogue," https://nvd.nist.gov/vuln/categories, 2024.

[66] T. E. Carroll and D. Grosu, "A game theoretic investigation of deception in cyber security," in *Proc. 19th Int. Conf. Computer Communications and Networks (ICCCN)*, 2011, pp. 1–6.

[67] L. Spitzner, *Honeypots: Tracking Hackers*. Addison-Wesley, 2003.

[68] R. Gangupantulu, R. Pamula, T. Cody, A. Rahman, and S. Park, "Using cyber digital twins for automated network security analysis," in *Proc. Annual Simulation Symposium (ANSS)*, 2021, pp. 1–12.

[69] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security," *NIST Special Publication 800-82 Rev. 2*, 2015.