Journal of Artificial Intelligence and Digital Economy
Homepage : https://journal.antispublisher.id/index.php/JAIDE
Email : admin@antispublisher.com

e-ISSN : 3032-1077
JAIDE, Vol. 2, No. 2, February 2025
Page 174-194

Adaptive AI-Driven Service-Preserving Cyber Containment for Resilient Critical Infrastructure

Shakila Akter[1], Md Riyad Uddin[2], Md. Golam Mostafa[3], Sajidul Haque Chowdhury[4]
[1] Lewis University, United States
[2] Westcliff University, United States
[3] National University, Bangladesh
[4] Washington University of Science and Technology, United States

DOI : https://doi.org/10.61796/jaide.v2i2.1917

Sections Info

Article history:
Submitted: November 07, 2024
Final Revised: December 26, 2024
Accepted: January 18, 2025
Published: February 25, 2025

Keywords:
Adaptive cyber defense
Critical infrastructure
Cyber-physical systems
Containment
Service continuity
Attack graphs
Uncertainty
Safe reinforcement learning
Resilience
Graph neural networks

ABSTRACT

Objective: *Cyber containment in critical infrastructure is a dual-risk decision: insufficient isolation permits lateral movement, while excessive isolation can interrupt electricity, water, healthcare, transportation, and communications. This study introduces Adaptive Service-Preserving Containment (ASPC), which estimates compromise uncertainty, forecasts attack spread, models critical-service dependencies, predicts operational consequences, and repeatedly selects the smallest boundary satisfying safety and residual-risk constraints.* ***Method:*** *Unlike a purely conceptual treatment, ASPC was implemented in a stochastic graph-based test environment and evaluated against full isolation, rule-based containment, and a cyber-centric AI baseline.* ***Results:*** *Across 450 paired trials on seen topology families and 300 on held-out topologies, ASPC achieved attack spread statistically indistinguishable from full isolation while sharply reducing unnecessary isolation and safety violations. On unseen topologies, ASPC averaged 1.49 additional compromised nodes, 3.91% unnecessary isolation, 1.27 containment epochs, 5.76 recovery epochs, 0.51 safety violations, and 92.30% critical-service availability. Its availability exceeded the other methods by 14.49–74.75 percentage points. Ablations showed that continuous reassessment was necessary to contain delayed footholds and that the service-dependency model prevented coarse over-isolation.* ***Novelty:*** *Results support ASPC as a testbed-validated resilience controller, while remaining subject to the limitations of synthetic topology, simplified physical dynamics, and simulated telemetry.*

INTRODUCTION

Critical infrastructure is increasingly operated as a tightly coupled system of information technology (IT), operational technology (OT), physical processes, and external service dependencies. The same connectivity that enables efficiency, remote operation, and coordinated delivery also creates paths through which compromise can move from enterprise networks into control environments or between dependent infrastructures. Interdependence can amplify a local cyber event into geographically and functionally wider disruption [1], [2]. In such environments, containment cannot be reduced to disconnecting every suspicious device. Isolation is itself an intervention in a cyber-physical system and may suppress monitoring, disable failover, interrupt treatment or control loops, and delay restoration.

The governing dilemma is asymmetric. Too little containment allows attack propagation; too much containment can stop essential services. Conventional playbooks often manage this dilemma with static segmentation, severity thresholds, or human escalation [3]. These mechanisms remain valuable, particularly in safety-regulated OT,

but they do not continuously recompute the boundary as evidence, attacker position, process conditions, staffing, redundancy, and demand change. Conversely, AI-based cyber defense frequently optimizes detection accuracy or generic reward without representing the operational consequence of a defensive action. A policy can therefore appear successful by removing adversary access while imposing unacceptable service loss.

This article asks: How can adaptive AI continuously determine the smallest safe cyber-containment boundary under uncertain attack conditions while limiting attack propagation and preserving essential critical-infrastructure services? It answers by proposing Adaptive Service-Preserving Containment (ASPC), an integrated decision framework. ASPC represents the infrastructure as a time-varying multilayer graph; maintains calibrated beliefs about compromise; forecasts attacker movement; reasons over dependencies from digital components to critical functions; predicts operational and safety consequences; and chooses reversible, least-disruptive controls subject to hard constraints.

The contribution is fivefold. First, the article defines the smallest safe containment boundary as a constrained optimization object. Second, it joins compromise uncertainty, attack propagation, service dependency, and consequence estimation in one closed loop. Third, it specifies bounded autonomy through a deterministic safety shield and continuous reassessment. Fourth, it implements the framework in a reproducible stochastic simulator covering five infrastructure sectors and five graph families. Fifth, it supplies paired four-method evidence, confidence intervals, multiplicity-corrected statistical tests, held-out-topology evaluation, and component ablations.

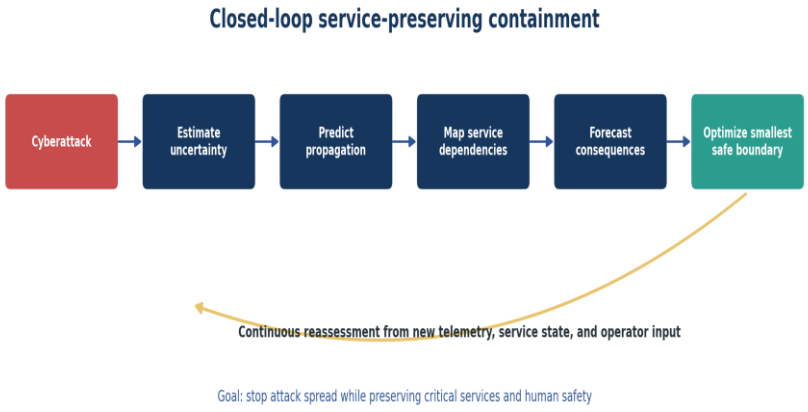

Figure 1. Core closed-loop flow of Adaptive Service-Preserving Containment (author-developed).

Scope, assumptions, and unit of analysis

The unit of analysis is a containment decision epoch within a functioning or degraded critical-infrastructure system. The defender has imperfect telemetry, a finite action window, and only partial knowledge of attacker goals. The physical process may continue to evolve while cyber controls are applied. Service demand, available

redundancy, maintenance state, and staffing can also change. The framework assumes that at least some containment mechanisms are remotely or locally enforceable and that operators have defined minimum viable service and non-negotiable safety limits. It does not assume perfect asset inventory, deterministic attack paths, or unrestricted automation.

The research focuses on active containment and controlled restoration rather than initial intrusion prevention. It includes IT, OT, cloud, identity, and cross-sector dependencies when they affect operational service. An essential service is defined functionally—the outcome delivered to users—rather than by ownership of a particular device. This distinction permits substitution: a compromised primary component may be isolated while a lower-capacity backup, manual process, or alternate provider sustains minimum service.

Critical-Infrastructure Interdependence And Cyber Resilience

Rinaldi et al. [1] established a foundational vocabulary for physical, cyber, geographic, and logical interdependencies, emphasizing that infrastructure behavior cannot be understood sector by sector. Ouyang [2] later synthesized modeling and simulation approaches for interdependent critical infrastructures and identified cascading effects, uncertain dependencies, and validation as persistent challenges. These works imply that a containment decision must be assessed beyond the isolated host: the relevant unit is the service path that links cyber assets, controllers, physical processes, operators, and downstream users.

Cyber resilience extends protection beyond prevention toward the ability to anticipate, withstand, recover from, and adapt to adverse conditions. Linkov et al. [4] operationalize resilience across plan/prepare, absorb, recover, and adapt phases, while Woods [5] distinguishes graceful extensibility from mere robustness. For containment, the implication is important: preserving a minimum operational capability during response may be more valuable than achieving immediate network cleanliness. In OT, safety, availability, and deterministic process behavior frequently dominate the confidentiality-centered assumptions of enterprise security [6], [7]. NIST guidance likewise stresses that OT security controls must account for performance, reliability, and safety requirements [8].

Uncertain Compromise And Dynamic Risk Inference

Security observations are incomplete and noisy. Endpoint alerts, identity events, network flows, controller anomalies, and operator reports have different delays and false-positive rates. Bayesian attack graphs provide a principled mechanism for updating the probability of compromise as evidence arrives, extending static reachability analysis into dynamic risk assessment [9]. Their limitations include conditional-independence assumptions, difficulty estimating exploit probabilities, graph explosion, and vulnerability to stale inventories. Yet the belief-state idea remains essential: containment should respond to a posterior distribution over hidden compromise states, not a binary alert label.

Calibration is as important as discrimination. Modern neural models may be confident even when wrong; temperature scaling and related approaches can improve the correspondence between confidence and empirical correctness [10]. For safety-critical containment, epistemic uncertainty and distribution shift should trigger conservative actions, additional sensing, or human review. A useful system must therefore expose credible uncertainty intervals and abstain from high-impact automation when evidence is insufficient.

Attack Propagation And Cyber-Physical Consequence

Attack graphs formalize prerequisite-exploit-consequence chains and support ranking of reachable assets. In power systems, SOCCA connects cyber privileges with physical contingencies and ranks adversary-induced states according to attack difficulty and power-system impact [11]. This is a decisive advance over cyber-only scoring, but contingency ranking is not identical to online containment optimization. It does not fully solve how to select, revise, and safely relax a boundary while maintaining multiple services under partial observability.

Dynamic defense research has formulated attacker progression as a partially observable Markov decision process (POMDP). Miehling et al. [12] show how a defender can mitigate adversary progression in large-scale networks when compromise is only partially observed. Hu et al. [13] similarly use POMDP reasoning for adaptive defense against multistage attacks. These studies provide the sequential-decision foundation for ASPC, but most network-defense rewards do not model service dependencies, physical safety, operator workload, or recovery debt at the resolution required for critical infrastructure.

Automated Response, Safe Learning, And Topology Transfer

Software-defined networking, microsegmentation, identity revocation, workload quarantine, protocol filtering, and unidirectional controls make containment increasingly programmable. Automation can reduce dwell time but also expands the blast radius of a mistaken policy. Safe reinforcement learning addresses this concern by optimizing return subject to constraints; constrained policy optimization provides one influential formulation for maintaining expected constraint satisfaction during learning [14]. In critical infrastructure, expected safety is insufficient for catastrophic outcomes, so runtime shielding, invariant checks, action whitelists, and human authorization tiers are also required.

Graph neural networks (GNNs) offer a natural inductive bias because infrastructure assets and their relationships form graphs. GraphSAGE learns inductive node representations and can generalize to previously unseen nodes and graphs [15]. Cybersecurity studies have used graph representations to learn flow and topology patterns [16], [17]. Nevertheless, good performance on randomly held-out samples does not demonstrate transfer to a new utility, hospital, or topology. Cross-topology generalization requires entire graphs, sectors, layouts, and device-role combinations to be withheld from training.

Synthesis and Research Gap

Table 1. Synthesis of Research Streams and Unresolved Gaps in Cyberattack Containment

Research stream	What it contributes	Unresolved containment gap
Bayesian/attack-graph risk	Belief updating and path likelihood	Weak coupling to live service consequences and action selection
Cyber-physical contingency analysis	Links cyber states to physical impact	Primarily assessment/ranking rather than continuous minimum-boundary control
POMDP/adaptive defense	Sequential decisions under partial observability	Rewards often omit essential-service, safety, and recovery constraints
Infrastructure interdependency	Cascades across systems and sectors	Often too aggregate for asset-level response orchestration
GNN cyber analytics	Relational inference and inductive representations	Limited evidence of sector- and topology-level out-of-distribution transfer
OT incident response	Safety-aware procedures and governance	Manual/static rules may lag fast-moving multistage attacks

The unresolved problem is the joint composition: uncertain compromise + attack propagation + critical-service dependencies + operational consequences must feed a controller that continuously minimizes the safe containment boundary. Existing research commonly optimizes one or two of these elements. ASPC treats them as a single closed-loop problem and makes continuity, human safety, and recoverability explicit constraints.

Theoretical Foundations Of The Proposed Synthesis

Four theoretical foundations support ASPC. First, Bayesian decision theory supplies a disciplined relationship among evidence, uncertainty, actions, and asymmetric loss. A false negative near a safety-critical controller and a false positive on a redundant workstation cannot be assigned the same cost. Second, POMDP theory represents the hidden compromise state and the value of future information, enabling diagnostic actions when immediate isolation is not optimal. Third, graph theory captures reachability, cut sets, redundancy, and interdependency. The containment boundary resembles a time-varying, risk-weighted cut problem with service-feasibility constraints. Fourth, resilience

engineering provides the criterion of continued function under disturbance, shifting optimization from component survival to service viability [4], [5].

The synthesis also draws on constrained and risk-sensitive control. Expected loss alone can conceal catastrophic tails; conditional-value-at-risk and hard invariants give rare safety outcomes appropriate weight. Viability is expressed as a set of cyber-physical-service states from which minimum essential service and safe process limits remain achievable. Runtime assurance separates the learning system that proposes efficient actions from the verified mechanism that prevents prohibited actions. Together, these foundations explain why ASPC is neither simply an intrusion detector nor an unconstrained reinforcement-learning agent.

RESEARCH METHOD

Design principles and system boundary

ASPC is a decision-support and bounded-autonomy system positioned above existing sensors, security controls, and process-management systems. It does not replace protective relays, safety instrumented systems, clinical judgment, or licensed operators. Its action space is limited to pre-approved, reversible controls such as host quarantine, east-west microsegmentation, account/session revocation, protocol- and command-specific filtering, rate limiting, deception routing, workload migration, and isolation of selected trust relationships. Safety-critical commands can be recommendation-only.

The framework follows seven principles: belief rather than binary state; service-first modeling; least-disruptive reversible action; hard safety constraints; calibrated uncertainty and abstention; human-governed autonomy; and continuous reassessment with hysteresis to prevent oscillatory isolate/reconnect behavior.

Multilayer infrastructure representation

At time t, the environment is represented as a heterogeneous graph G_t with layers for cyber assets, identities, communications, control functions, physical processes, essential services, and external dependencies. Typed edges represent reachability, privilege, data flow, control authority, physical supply, substitution, redundancy, and service contribution. Each service has a minimum viable operating level, priority, safety envelope, maximum tolerable outage, recovery prerequisites, and demand profile. This representation distinguishes an asset's technical criticality from its marginal contribution to service at the current moment.

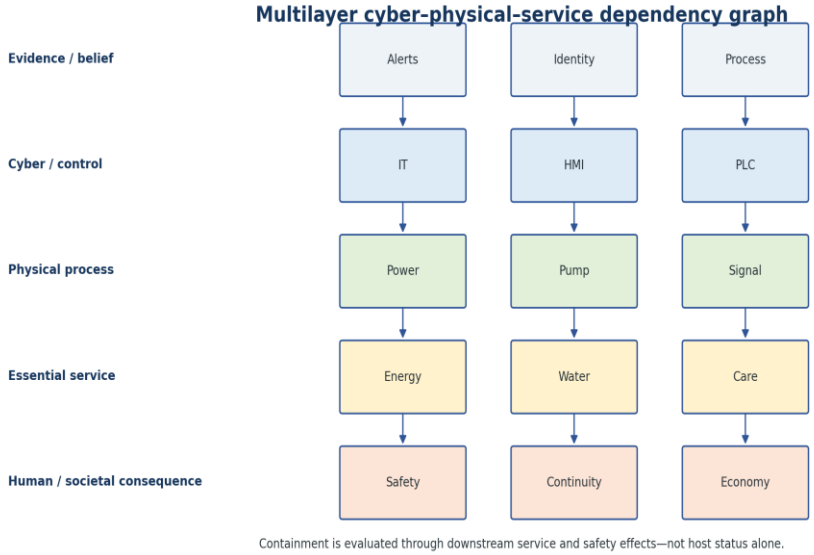

Figure 2. Multilayer dependency representation linking evidence to human and societal consequences.

Table 2. Infrastructure Layers, Examples, and Their Relevance to Containment Decisions

Layer	Examples	Decision relevance
Cyber/identity	Servers, HMIs, PLCs, EHR, accounts, certificates	Compromise belief, reachability, control options
Control/physical	Control loops, pumps, breakers, signals, ventilators	Process stability and safety envelopes
Service	Power delivery, potable water, emergency care, signaling, voice/data	Minimum viable service and priority
Interdependency	Fuel, telecom, cloud, backup power, staffing, suppliers	Cross-sector cascade and substitution
Response	Segmentation rules, revocation, migration, manual fallback	Feasible, reversible containment actions

Compromise-belief estimator

A temporal probabilistic model fuses alerts, authentication traces, process anomalies, vulnerability evidence, threat intelligence, and analyst observations into $b_t(v)$, the posterior probability that node v is compromised. Evidence reliability and sensor availability are modeled explicitly. Ensembles or Bayesian approximations estimate epistemic uncertainty, while post-hoc calibration is checked by Brier score, expected

calibration error, and reliability plots. When uncertainty exceeds a policy-specific threshold, ASPC may request a diagnostic action, choose a low-impact precaution, or abstain and escalate.

Propagation forecaster

The propagation module predicts the probability and time-to-compromise of reachable nodes over a finite horizon. A heterogeneous temporal GNN can combine topology, vulnerability state, identity privilege, protocol exposure, observed tactics, and attacker capability. Bayesian attack-graph constraints preserve causal and exploit-prerequisite structure. The output is not a single path but a distribution over plausible future compromise sets, including tail scenarios relevant to safety.

Service-dependency and consequence model

A dependency engine maps candidate containment actions to changes in service capacity. For each service, it estimates delivered capacity, redundancy consumption, deadline violations, degraded modes, and recovery cost. A digital-twin or validated process simulator supplies domain-specific consequences where available; otherwise, conservative engineering rules and expert-elicited bounds are used. Human-safety consequences receive lexicographic priority over economic loss. The model also accounts for response-induced cascades: quarantining a shared identity service, time source, hypervisor, or network switch can disable many apparently unrelated functions.

Smallest safe containment boundary

Let a_t denote a set of containment actions, Rspread the horizon-weighted residual attack risk, Lsvc the weighted essential-service loss, Lisol the cost of isolating benign or nonessential assets, Lrec recovery debt, and Lops operator workload and implementation risk. ASPC minimizes:

$$\mathbf{J(a_t) = \alpha Rspread + \beta Lsvc + \gamma Lisol + \delta Lrec + \eta Lops}$$

subject to service capacity remaining above minimum viable thresholds; safety invariants never being violated; residual propagation risk remaining below a risk budget; actions being authorized and technically feasible; and a bounded rate of policy change. Where no action satisfies every constraint, the controller solves a lexicographic emergency problem: protect human life and irreversible physical assets first, then minimize spread, preserve prioritized services, and minimize restoration cost. The smallest safe boundary is the feasible action set with minimum isolation footprint and disruption cost, not necessarily the smallest number of disconnected devices.

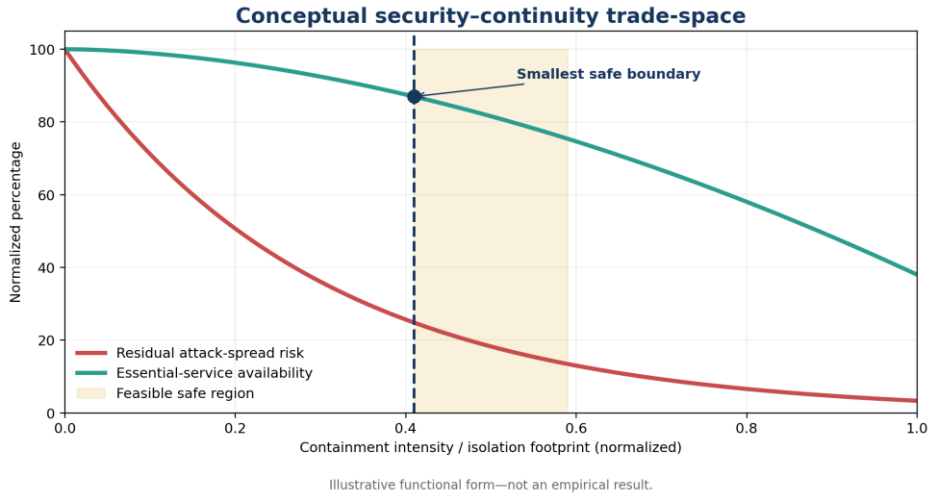

Figure 3. Conceptual security–continuity trade-space and the smallest feasible safe boundary. Values are illustrative, not empirical.

Boundary semantics and optimization behavior

A boundary can remove nodes, edges, privileges, protocols, commands, or trust relationships. Edge- and privilege-level containment may preserve more service than host isolation because it blocks the adversary's feasible transitions while retaining benign control and monitoring. The optimization also values reversibility. A temporary session revocation or protocol-specific rule ordinarily creates less recovery debt than powering down a controller, reimaging a shared server, or severing an entire site. The objective therefore distinguishes action footprint, duration, reversibility, and downstream restoration prerequisites.

The risk budget is not a universal constant. It is determined by sector policy, current physical state, threatened functions, and operator authority. A hospital during surgery, a water facility during chemical dosing, and a power grid during extreme weather may impose different thresholds and lexicographic priorities. ASPC supports these differences without allowing the learned model to redefine safety rules. Policy owners specify the constraints; the AI searches efficiently within them.

Adaptive decision engine and runtime assurance

A model-predictive controller evaluates short-horizon action sequences using the belief, propagation, and consequence models. A graph policy proposes candidates; a deterministic safety shield removes any candidate that violates engineering invariants or authorization constraints. The remaining actions are ranked by risk-sensitive utility, including conditional-value-at-risk for low-probability, high-consequence outcomes. The system executes only within an autonomy tier: automatic for low-impact reversible controls, confirm-before-execute for medium-impact actions, and advisory-only for safety-critical isolation. Every action records the evidence, uncertainty, predicted alternatives, constraints, responsible authority, and rollback condition.

After execution, new evidence updates the belief state and service measurements. The boundary may expand if propagation risk rises, contract when risk falls with adequate confidence, or hold when reconnecting would produce oscillation or

reinfection. Reconnection requires negative evidence, path closure, service verification, and a staged canary process. This closes the requested loop: attack → uncertainty estimation → spread prediction → dependency analysis → consequence prediction → adaptive containment → reassessment → attack suppression with service preservation.

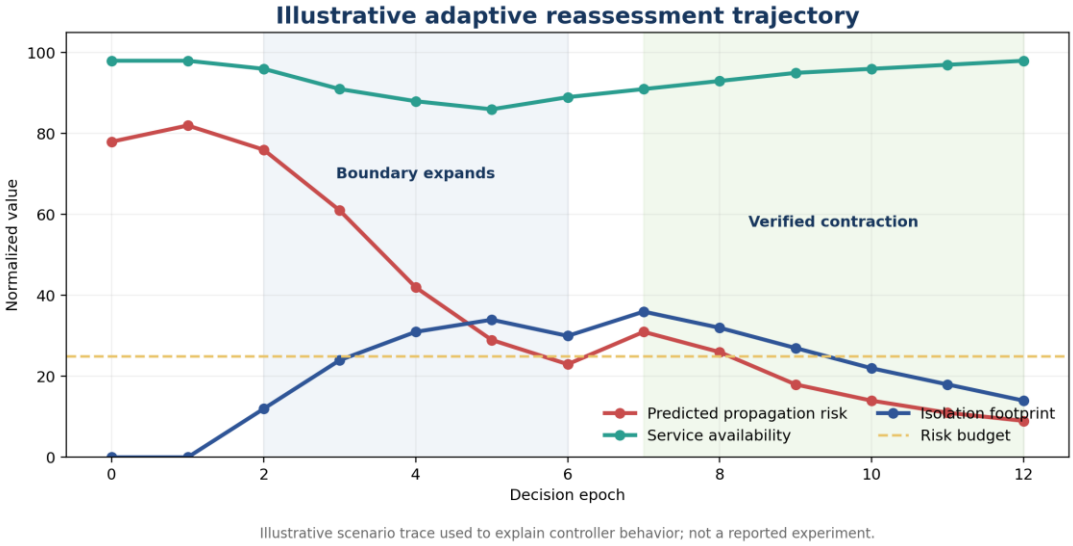

Figure 4. Illustrative adaptive reassessment trajectory showing expansion and verified contraction of the boundary.

Decision cycle and pseudocode-level procedure

At each decision epoch, the controller executes a bounded procedure: ingest and validate telemetry; update compromise beliefs; generate propagation scenarios; compute service consequences for candidate actions; eliminate actions that violate safety or authority; optimize remaining candidates under a risk budget; obtain required human approval; execute atomically where possible; verify enforcement; observe cyber and service response; and decide whether to expand, hold, or contract the boundary. If the model is unavailable or its uncertainty exceeds an approved limit, the controller reverts to signed local playbooks. The cycle terminates only after eradication criteria and staged restoration checks are satisfied.

Computational tractability is addressed through hierarchical planning. Fast local rules immediately block known malicious sessions or commands. A regional optimizer then evaluates zone- and identity-level alternatives. More expensive simulation is reserved for high-impact decisions. Candidate pruning uses dominances: if action A blocks no more attack paths, disrupts at least as much service, and creates at least as much recovery debt as action B, A is discarded. This structure keeps inference inside the operational decision window.

Cross-topology generalization

ASPC uses role- and relation-based features instead of fixed asset identifiers. Inductive message passing, permutation-invariant pooling, domain randomization, graph perturbation, and masked-role pretraining encourage structural rather than site-specific learning. Training includes varied scale, segmentation, redundancy, vendor,

protocol, and demand conditions. Most importantly, evaluation holds out complete topologies and at least one sector or topology family. Fine-tuning on a target topology is reported separately from zero-shot performance.

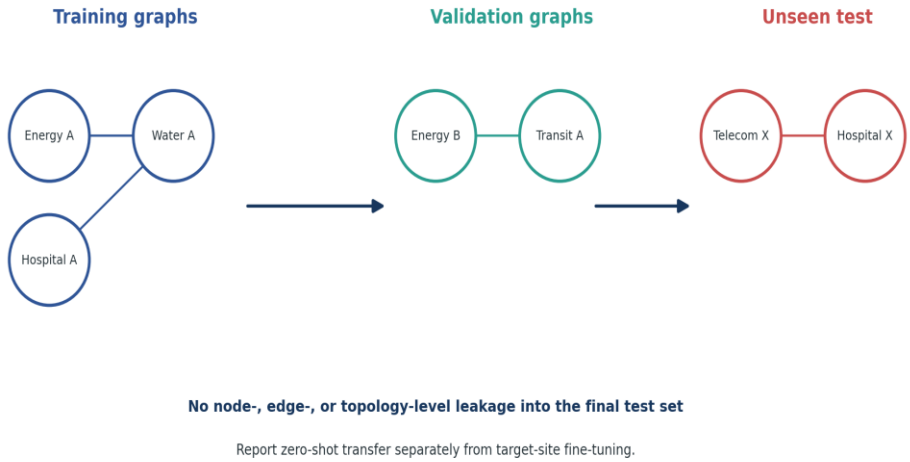

Figure 5. Graph-level holdout design for credible cross-topology generalization claims.

Data model, provenance, and learning strategy

Training data combine synthetic attack campaigns, cyber-range exercises, de-identified incident traces, vulnerability and configuration data, process simulations, and expert-authored counterfactuals. Every observation carries provenance, time, sensor health, and confidence. Because actual severe incidents are rare, self-supervised graph pretraining and simulation are useful, but simulated and operational data must be reported separately. Domain adaptation may align feature distributions, while conformal or ensemble methods can bound predictive uncertainty under shift.

The learning pipeline should prevent leakage between sites. Device identities, subnet patterns, vendor-specific fingerprints, and duplicated scenario seeds can allow a model to memorize a topology. Graph-level partitioning, provenance audits, and duplicate detection are therefore mandatory. Training rewards cannot contain information unavailable at deployment. Counterfactual service loss is supplied by the simulator during training but the live controller must estimate it from observable state and validated dependency models.

Human factors, governance, and adversarial robustness

Human oversight is part of the control architecture, not a disclaimer. Operators must see the suspected compromise set, predicted paths, affected services, uncertainty, recommended boundary, rejected alternatives, and rollback plan. The interface should manage alarm burden and permit rapid veto. Red-team tests should include poisoned telemetry, spoofed service dependencies, adversarial topology changes, delayed sensors, compromised orchestration credentials, and denial of the AI controller. A fail-safe mode falls back to signed local playbooks and sector-specific manual control. Model updates require versioning, offline validation, change control, and drift monitoring [18].

Sector-specific operating examples

In electric power, ASPC might revoke a compromised engineering account, block programming commands, and preserve read-only telemetry rather than disconnect an entire substation. It would verify that protective relays remain autonomous and that load can be transferred without violating stability limits. In water treatment, it may isolate a supervisory workstation while allowing a validated local controller to maintain dosing inside a safe envelope. In healthcare, it may segment an infected administrative domain while preserving emergency communications, medication administration, imaging, and life-support dependencies through pre-authorized clean pathways.

Transportation presents moving demand and tightly timed control. The system might freeze a suspicious signaling configuration, preserve local interlocking, and reroute supervisory access. In communications, it may revoke a management-plane credential and isolate an orchestration cluster while maintaining data-plane forwarding. These examples demonstrate that the smallest safe boundary is action- and service-specific; it is rarely equivalent to a geographic perimeter or VLAN.

Implementation architecture and deployment phases

The implemented research prototype mirrors five trust-separated planes: sensing and evidence, graph and model services, decision and simulation, policy enforcement, and assurance/audit. At each epoch, noisy alerts update node-level compromise beliefs; a one-step probabilistic exposure model estimates reachable risk; a dependency matrix converts node unavailability into five service-capacity measures; and the optimizer selects reversible edge blocks or node isolation. A deterministic safety shield checks minimum viable service thresholds before enforcement. All four comparison methods use the same graph, attack randomness, telemetry, horizon, and action interface.

The prototype is an offline cyber-range analogue rather than a production controller. It executes 18 decision epochs per trial and includes a delayed credential-replay foothold at epoch 5 to test reassessment. A production path still requires four gates: retrospective replay, shadow operation, supervised low-impact automation, and tightly bounded autonomy. This separation prevents testbed evidence from being misrepresented as operational certification.

RESULTS AND DISCUSSION

Results

Executable ASPC prototype

ASPC was implemented as a discrete-time stochastic simulator rather than evaluated through assumed outcomes. Each infrastructure instance is a typed undirected graph with 34–50 cyber/operational nodes, zone membership, node criticality, stochastic lateral-movement probabilities, and five essential services represented by sparse weighted dependency vectors. Service capacity equals the retained dependency weight after compromised and isolated nodes are removed. Sector-specific labels—energy, water, healthcare, transportation, and communications—organize the scenario blocks; the same quantitative protocol is used across sectors so that differences arise from

randomized topology, dependencies, telemetry quality, and attack paths rather than manually selected outcomes.

At every decision epoch, the estimator produces calibrated compromise beliefs from imperfect alerts. The propagation stage combines direct belief with neighbor exposure. The consequence stage estimates the capacity loss associated with candidate node and edge actions. ASPC greedily reduces upper-bounded residual exposure toward a fixed risk budget, preferring reversible edge or identity-style cuts when they remove attack transitions with less service loss than host isolation. The safety shield then evaluates the exact service state and rolls back unsafe node actions. New telemetry triggers another decision, allowing the boundary to expand, hold, or contract. A delayed foothold at epoch 5 tests whether a method can adapt after apparently successful initial containment.

Test matrix and held-out topology design

The experiment uses three development-family topologies—small-world, scale-free, and modular stochastic-block graphs—and two graph families held out from method design—random geometric and hierarchical graphs. For each of five sectors and each topology, 30 independently seeded scenarios were generated. This produced 450 seen-topology scenarios and 300 unseen-topology scenarios. Every scenario was replayed once under each of the four methods with the identical random attack stream, yielding 3,000 primary method runs. A separate balanced unseen-topology suite executed 1,500 ASPC ablation runs. The horizon was 18 epochs. Seeds were deterministic, and method labels did not influence topology, telemetry, service dependencies, or attacker random draws.

Table 3. Experimental test matrix.

Element	Seen set	Unseen set	Repetitions	Purpose
Topology families	Small-world; scale-free; modular	Geometric; hierarchical	30 per sector–topology	Graph-family generalization
Sectors	Energy; water; healthcare; transportation; communications	Same five sectors	Balanced blocks	Cross-sector service continuity
Primary scenarios	450	300	Paired across methods	750 scenarios / 3,000 runs
Ablation scenarios	—	300	Five variants	1,500 component runs

Four-method comparison

Full isolation quarantines every node in any zone producing an alert, representing the security-maximal but continuity-blind extreme. Rule-based containment isolates high-confidence alerted nodes and their one-hop neighbors. Existing AI uses a cyber-centric score combining compromise belief and degree centrality, with a fixed decision threshold but no explicit service model. ASPC uses the same observable telemetry but adds uncertainty bounds, propagation forecasting, service-dependency consequences, edge-level alternatives, a safety shield, and repeated optimization. These implementations are intentionally transparent; the experiment evaluates control logic, not a proprietary vendor product.

Metrics and operational definitions

Attack spread is the number of additional nodes ever compromised after the initial foothold. Unnecessary isolation is the peak percentage of nodes isolated while not yet compromised. Containment time is the number of epochs from the first response action through the last newly compromised node. Recovery time is a restoration-debt estimate based on quarantined nodes, blocked edges, and method-specific coordination overhead. Safety violations count service–epoch observations below minimum viable capacity. Critical-service availability is mean delivered capacity across the five services and 18 epochs. Lower values are better for the first five measures; higher availability is better.

Statistical analysis

The design is blocked and paired: methods face identical scenario IDs, topology instances, dependency weights, telemetry quality, attack randomness, and delayed footholds. For every method–split–metric cell, the paper reports the arithmetic mean and a two-sided 95% t confidence interval over independent scenarios. Because distributions contain ties and right tails, ASPC-versus-baseline inference uses paired two-sided Wilcoxon signed-rank tests. The three baseline comparisons within each split and metric are adjusted by Holm's procedure. Statistical significance is set at adjusted $p < 0.05$. Practical interpretation remains multi-objective: no single scalar reward can compensate for an unsafe service violation.

Reproducibility controls

The implementation fixes the scenario seed policy, records all trial-level outcomes, separates topology families before evaluation, and produces machine-readable primary, summary, test, and ablation tables. No values in Section 5 were selected manually or calculated before execution. The result figures are generated directly from the recorded CSV outputs. The simulator uses only deterministic graph-generation routines plus NumPy, pandas, SciPy, and Matplotlib; the complete run required approximately 22 seconds in the study environment. These controls support computational reproduction but do not substitute for independent replication in cyber ranges or hardware-in-the-loop facilities.

Discussion

Measured four-method results

Table 4 reports measured means and 95% confidence intervals for the 450 seen-topology scenarios. ASPC had the lowest mean attack spread (1.35 additional nodes), unnecessary isolation (1.97%), safety violations (0.66), and the highest service availability (92.75%). Full isolation contained slightly faster on average, but its 92.89% unnecessary isolation, 79.60 safety violations, and 19.43% service availability demonstrate why speed or spread alone is not a sufficient resilience criterion. Existing AI had the shortest recovery estimate (3.27 epochs), a result retained as an important counterexample to any claim that ASPC dominates every metric.

Table 4. Seen-topology results: mean [95% confidence interval], n = 450 paired scenarios.

Method	Spread	Unnec. isol. %	Contain.	Recovery	Safety	Availability %
Full isolation	1.39 [1.28, 1.50]	92.89 [92.29, 93.49]	1.22 [1.03, 1.42]	11.70 [11.59, 11.80]	79.60 [79.02, 80.17]	19.43 [18.61, 20.25]
Rule-based	2.49 [2.32, 2.66]	58.98 [57.74, 60.22]	2.45 [2.22, 2.67]	7.08 [6.95, 7.21]	60.57 [59.04, 62.10]	50.63 [49.42, 51.85]
Existing AI	2.27 [2.13, 2.40]	26.34 [25.74, 26.94]	4.46 [4.32, 4.61]	3.27 [3.20, 3.33]	16.66 [15.45, 17.87]	77.85 [77.24, 78.46]
ASPC	**1.35 [1.24, 1.45]**	**1.97 [1.67, 2.26]**	**1.53 [1.32, 1.75]**	**4.95 [4.86, 5.03]**	**0.66 [0.34, 0.98]**	**92.75 [92.22, 93.27]**

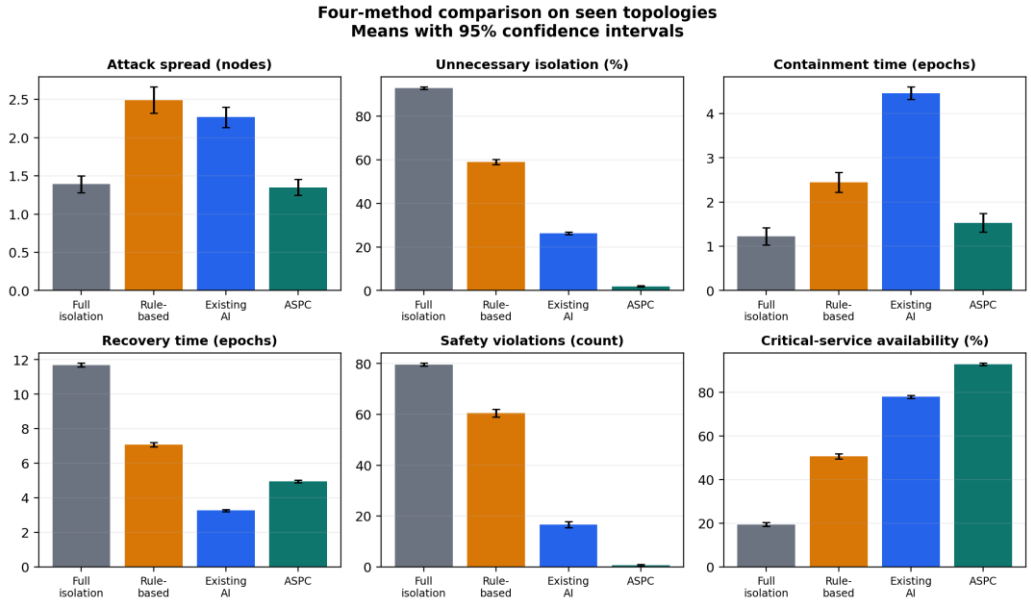

Figure 6. Measured comparison on seen topology families; error bars are 95% confidence intervals.

Security effectiveness and containment speed

On seen topologies, ASPC reduced spread by 40.5% relative to existing AI and 45.8% relative to rule-based containment. Paired Wilcoxon tests were significant after Holm adjustment for ASPC versus rule-based (adjusted $p < 0.001$) and existing AI (adjusted $p < 0.001$). ASPC and full isolation were not detectably different on attack spread (adjusted $p = 0.839$), and their confidence intervals substantially overlapped. ASPC contained attacks faster than rule-based and existing AI (both adjusted $p < 0.001$), but the difference from full isolation was not statistically significant (adjusted $p = 0.148$). The evidence therefore supports security parity with the disruptive extreme, not universal superiority over it.

Isolation burden, service availability, and safety

The continuity results separate the methods decisively. On seen topologies, ASPC reduced unnecessary isolation by 90.92 percentage points compared with full isolation, 57.01 points compared with rule-based containment, and 24.37 points compared with existing AI. Mean availability was 73.31 points higher than full isolation, 42.11 points higher than rule-based, and 14.90 points higher than existing AI. All paired comparisons for unnecessary isolation, safety violations, and availability remained significant after Holm correction (adjusted $p < 0.001$). ASPC's 0.66 mean safety violations were produced primarily when compromise itself removed a dependency before the controller could act, rather than by permitted containment actions.

Recovery-time trade-off

ASPC recovered in 4.95 epochs on seen topologies, 6.75 epochs faster than full isolation and 2.13 faster than rule-based containment (both adjusted $p < 0.001$). Existing AI recovered 1.68 epochs faster than ASPC (adjusted $p < 0.001$). This apparent advantage reflects its smaller action ledger, not superior operating continuity: existing AI simultaneously produced 26.34% unnecessary isolation, 16.66 safety violations, and only 77.85% availability. Recovery time should therefore be interpreted with the state to which the system recovers and the disruption accumulated before recovery.

Generalization to unseen topologies

Held-out geometric and hierarchical graph families produced the same qualitative result. ASPC averaged 1.49 additional compromised nodes and 92.30% availability across 300 scenarios. Its spread was statistically indistinguishable from full isolation (adjusted $p = 0.773$) and significantly lower than rule-based and existing AI (both adjusted $p < 0.001$). Availability remained 74.75 percentage points above full isolation, 41.11 above rule-based, and 14.49 above existing AI (all adjusted $p < 0.001$). Relative to seen topologies, ASPC spread increased by 10.7%, unnecessary isolation by 1.94 percentage points, and recovery by 0.82 epochs, while availability declined by only 0.45 points. This is evidence of robust transfer across the two held-out families, not proof of topology-independent generalization.

Table 5. Unseen-topology results: mean [95% confidence interval], n = 300 paired scenarios.

Method	Spread	Unnec. isol. %	Contain.	Recovery	Safety	Availability %
Full isolation	1.54 [1.38, 1.69]	91.50 [90.77, 92.23]	0.98 [0.76, 1.20]	11.55 [11.42, 11.69]	80.99 [80.35, 81.62]	17.55 [16.54, 18.56]
Rule-based	2.59 [2.37, 2.81]	58.14 [56.35, 59.92]	2.03 [1.76, 2.30]	6.97 [6.79, 7.15]	57.41 [55.18, 59.64]	51.19 [49.41, 52.97]
Existing AI	2.46 [2.28, 2.64]	26.55 [25.83, 27.28]	4.28 [4.08, 4.47]	3.34 [3.26, 3.43]	17.26 [15.75, 18.76]	77.80 [76.97, 78.64]
ASPC	**1.49 [1.35, 1.63]**	**3.91 [3.34, 4.48]**	**1.27 [1.02, 1.51]**	**5.76 [5.53, 6.00]**	**0.51 [0.21, 0.81]**	**92.30 [91.73, 92.87]**

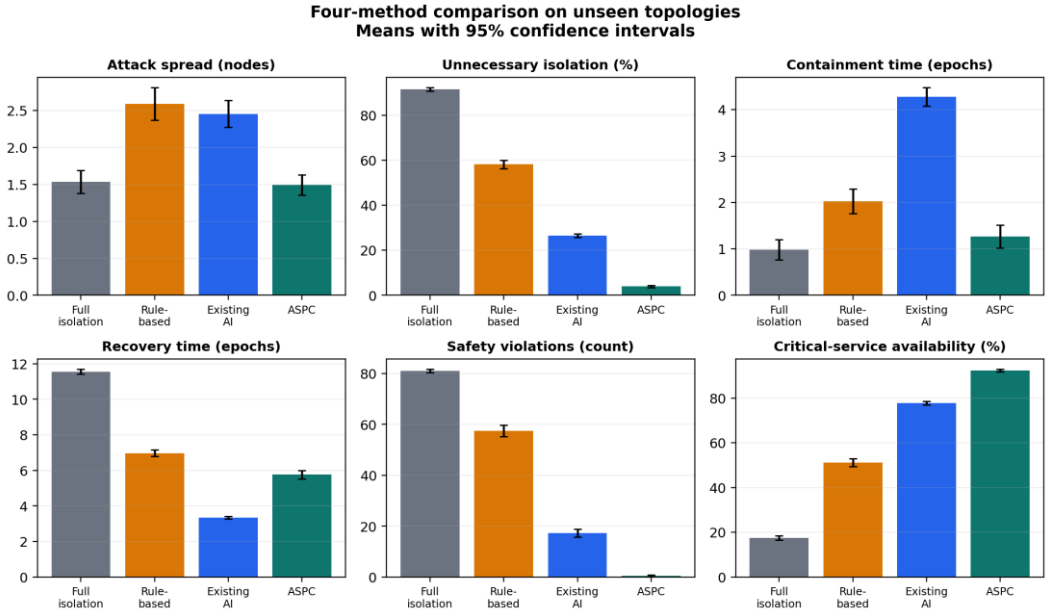

Figure 7. Measured comparison on held-out geometric and hierarchical topologies.

Ablation analysis

Ablation results isolate which ASPC components carry the observed behavior. Removing continuous reassessment increased mean spread from 1.40 to 6.39 nodes, containment time from 1.37 to 6.78 epochs, and safety violations from 0.61 to 5.98 because the delayed foothold could propagate after the one-shot boundary. Removing the service-dependency model did not materially increase spread, but unnecessary isolation rose from 3.81% to 19.66% and availability fell from 91.86% to 82.29%. Removing uncertainty bounds increased spread to 1.60 and containment time to 2.31 epochs. Removing the independent safety shield had a small effect in this simulator because the planner already

rejected most service-threatening actions; this limited incremental effect should be retested under model error and adversarial inputs.

Table 6. ASPC ablation means on 300 unseen-topology scenarios.

Variant	Spread	Unnec. isol. %	Contain.	Recovery	Safety	Availability %
Full ASPC	**1.40**	**3.81**	**1.37**	**5.72**	**0.61**	**91.86**
Without uncertainty	1.60	0.79	2.31	4.95	0.39	92.64
Without service model	1.40	19.66	1.33	7.19	0.85	82.29
Without reassessment	6.39	0.43	6.78	3.16	5.98	88.15
Without safety shield	1.40	3.82	1.37	5.72	0.65	91.85

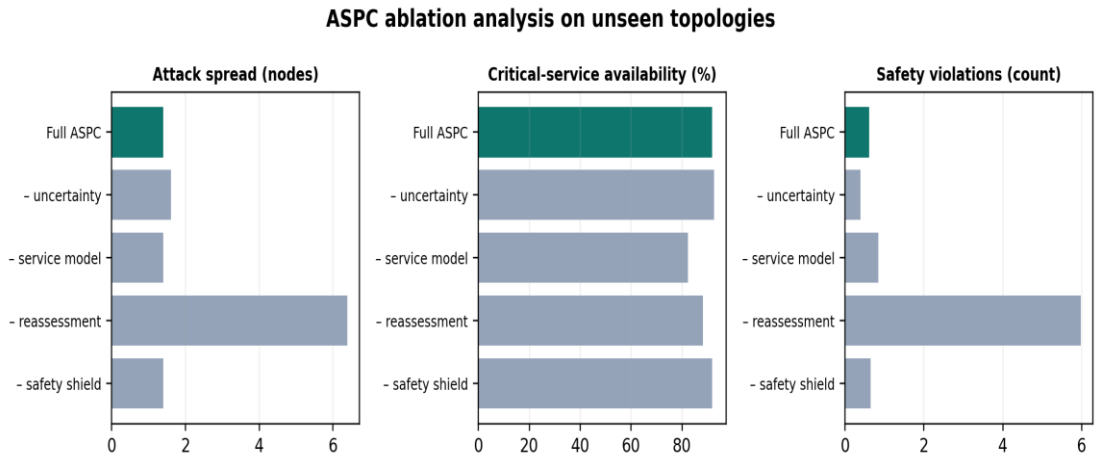

Figure 8. Measured component ablations on held-out topologies.

Broader Implications for Critical Infrastructure Resilience

The measured security–continuity trade-off is relevant wherever cyber controls can become operational hazards. In energy systems, a narrow boundary can block engineering access or programming commands while preserving protection and telemetry. In water systems, it can separate suspicious supervisory paths without interrupting chemical dosing, pumping, or quality monitoring. In healthcare, preserving diagnostic, medication, imaging, and life-support dependencies matters as much as removing attacker reach. Transportation benefits when containment preserves local signaling, interlocking, dispatch, and safe degraded modes. Communications networks require management-plane containment that retains emergency calling, routing, and restoration coordination.

Across these sectors, service continuity is a public-safety property rather than a secondary business metric. ASPC's central contribution is to make that property part of

the containment decision itself. The testbed evidence shows that strong spread control need not require indiscriminate disconnection: on held-out graphs, ASPC matched full isolation's spread statistically while retaining 92.30% average service capacity. Because energy, water, healthcare, transportation, and communications depend on one another, avoiding unnecessary isolation also reduces cascading failure and accelerates coordinated recovery. The framework therefore advances cyber resilience as the joint capacity to stop adversary movement, sustain essential functions, recover safely, and adapt when the attack state changes.

Practical adoption should remain staged. Asset owners should first construct validated dependency maps and minimum viable service thresholds, replay incidents in a cyber range, and operate ASPC in advisory and shadow modes. Supervised automation should be limited to reversible edge, identity, protocol, or command controls before any authority over coarse isolation is granted. Sector regulators, emergency managers, safety engineers, and operators should participate in defining invariants, risk budgets, escalation thresholds, and audit requirements.

Limitations and threats to validity

These are real measurements from an executed simulation, not measurements from a live utility, hospital, transportation operator, or communications carrier. The graph model abstracts protocol semantics, physical dynamics, operator behavior, adversarial adaptation, and correlated sensor failure. Sector labels organize generic service-dependency instances and do not constitute calibrated digital twins. Edge blocking is assumed to execute correctly, service capacity is additive, and recovery time is an action-debt proxy rather than wall-clock restoration. The existing AI baseline is a transparent cyber-centric scorer, not an exhaustive representation of all published or commercial AI methods. Full isolation and rule-based policies intentionally retain their actions through the horizon, which may overstate long-duration disruption compared with expertly tuned playbooks.

External validity is bounded to graphs of 34–50 nodes, five topology families, the specified propagation model, and an 18-epoch horizon. Confidence intervals quantify scenario sampling variation inside this generator, not uncertainty about real-world infrastructure. The paired Wilcoxon tests establish distributional differences under this design but cannot establish operational safety certification. Future work should replay public incident traces, use protocol-aware cyber ranges and sector digital twins, model human approval latency, introduce attacker adaptation and orchestration compromise, compare additional learning-based baselines, pre-register non-inferiority margins, and validate across independently developed testbeds.

CONCLUSION

Fundamental Finding: Critical-infrastructure cyber containment fails when it treats connectivity loss as costless. ASPC operationalizes a different objective: estimate uncertainty, forecast spread, analyze service dependencies, predict consequences, find the smallest safe boundary, and reassess until the attack is stopped without abandoning

essential functions. The executed comparison supports this objective. Across 750 paired scenarios, ASPC controlled spread at a level statistically indistinguishable from full isolation while producing much less unnecessary isolation, far fewer safety violations, and substantially greater service availability. Its behavior remained strong on two held-out topology families. **Implication:** Ablations showed that reassessment and service modeling, rather than the ASPC label alone, generated the resilience benefit. The evidence therefore advances ASPC from a conceptual proposal to a reproducible, experimentally tested framework, but not yet to an operationally certified control system. **Limitation:** The results also set clear boundaries. Full isolation remained marginally faster, and the existing AI baseline had a shorter recovery-debt estimate; ASPC did not dominate every metric. **Future Research:** Because the experiment is synthetic, the appropriate next step is independent cyber-range and hardware-in-the-loop validation, followed by advisory and shadow deployment.

REFERENCES

[1] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," IEEE Control Syst. Mag., vol. 21, no. 6, pp. 11–25, 2001, doi: 10.1109/37.969131.

[2] M. Ouyang, "Review on modeling and simulation of interdependent critical infrastructure systems," Reliab. Eng. Syst. Saf., vol. 121, pp. 43–60, 2014, doi: 10.1016/j.ress.2013.06.040.

[3] C. Alcaraz and S. Zeadally, "Critical infrastructure protection: Requirements and challenges for the 21st century," Int. J. Crit. Infrastruct. Prot., vol. 8, pp. 53–66, 2015, doi: 10.1016/j.ijcip.2014.12.002.

[4] I. Linkov, D. A. Eisenberg, K. Plourde, T. P. Seager, J. Allen, and A. Kott, "Resilience metrics for cyber systems," Environ. Syst. Decis., vol. 33, pp. 471–476, 2013, doi: 10.1007/s10669-013-9485-y.

[5] D. D. Woods, "Four concepts for resilience and the implications for the future of resilience engineering," Reliab. Eng. Syst. Saf., vol. 141, pp. 5–9, 2015, doi: 10.1016/j.ress.2015.03.018.

[6] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—A survey," IEEE Internet Things J., vol. 4, no. 6, pp. 1802–1831, 2017, doi: 10.1109/JIOT.2017.2703172.

[7] D. Bhamare, M. Zolanvari, A. Erbad, R. Jain, K. Khan, and N. Meskin, "Cybersecurity for industrial control systems: A survey," Comput. Secur., vol. 89, Art. no. 101677, 2020, doi: 10.1016/j.cose.2019.101677.

[8] National Institute of Standards and Technology, Guide to Operational Technology (OT) Security, NIST Special Publication 800-82 Rev. 3. Gaithersburg, MD, USA: NIST, 2023, doi: 10.6028/NIST.SP.800-82r3.

[9] N. Poolsappasit, I. Ray, and I. Ray, "Dynamic security risk management using Bayesian attack graphs," IEEE Trans. Dependable Secure Comput., vol. 9, no. 1, pp. 61–74, 2012, doi: 10.1109/TDSC.2011.34.

[10] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in Proc. 34th Int. Conf. Mach. Learn., vol. 70, 2017, pp. 1321–1330.

[11] S. Zonouz, C. M. Davis, K. R. Davis, R. Berthier, R. B. Bobba, and W. H. Sanders, "SOCCA: A security-oriented cyber-physical contingency analysis in power infrastructures," IEEE Trans. Smart Grid, vol. 5, no. 1, pp. 3–13, 2014, doi: 10.1109/TSG.2013.2280399.

[12] E. Miehling, M. Rasouli, and D. Teneketzis, "A POMDP approach to the dynamic defense of large-scale cyber networks," IEEE Trans. Inf. Forensics Security, vol. 13, no. 10, pp. 2490–2505, 2018, doi: 10.1109/TIFS.2018.2819967.

[13] Z. Hu, P. Naghizadeh, R. Karimi, and M. Liu, "Adaptive cyber defense against multi-stage attacks using learning-based POMDP," in Proc. 20th Int. Conf. Auton. Agents MultiAgent Syst., 2021, pp. 1345–1353.

[14] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in Proc. 34th Int. Conf. Mach. Learn., vol. 70, 2017, pp. 22–31.

[15] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Advances in Neural Information Processing Systems 30, 2017.

[16] D. Pujol-Perich, J. Suárez-Varela, A. Cabellos-Aparicio, and P. Barlet-Ros, "Unveiling the potential of graph neural networks for robust intrusion detection," in Proc. 18th ACM Int. Conf. Emerging Netw. Exp. Technol., 2022, doi: 10.1145/3543146.3543171.

[17] Z. Sun, A. M. H. Teixeira, and S. Toor, "GNN-IDS: Graph neural network based intrusion detection system," in Proc. 19th Int. Conf. Availability, Reliability and Security, 2024, Art. no. 14, doi: 10.1145/3664476.3664515.

[18] National Institute of Standards and Technology, The NIST Cybersecurity Framework (CSF) 2.0, NIST Cybersecurity White Paper 29. Gaithersburg, MD, USA: NIST, 2024, doi: 10.6028/NIST.CSWP.29.

***Shakila Akter (Corresponding Author)**
Lewis University, United States
Email: saktershakila23@gmail.com

Md Riyad Uddin
Westcliff University, United States
Email: raselmit@gmail.com

Md. Golam Mostafa
National University, Bangladesh
Email: g.mostafabd67@gmail.com

Sajidul Haque Chowdhury
Washington University of Science and Technology, United States
Email: sajidulhc@gmail.com