

American Journal of Data Science and Analytics

Volume: 7; **Issue:** 7
Pages: 99-143

An AI-Driven Cloud Resilience and Cybersecurity Decision-Support Framework for Resilient U.S. Enterprise IT Operations

Md. Mamunur Rashid[1];

[1]. *IT Infrastructure, Cloud Computing & Cybersecurity Researcher;*
Master of Science in Engineering Management, Westcliff University, Irvine, California, USA;
Email: mamunur2367@gmail.com

Doi: 10.63125/w7rhft74

Received: *13 April 2026;* ***Revised:*** *23 May 2026;* ***Accepted:*** *13 June 2026;* ***Published:*** *08 July 2026;*

Abstract

Enterprise IT operations in the United States depend on hyperscale cloud regions, third-party services and an identity layer whose failures and compromises propagate quickly and are now reportable events. This article proposes an AI-driven cloud resilience and cybersecurity decision-support framework in which one telemetry pipeline, one set of calibrated models and one incident lifecycle serve both reliability and security incidents: a six-layer reference architecture, a state-machine lifecycle with timing metrics, and a workflow in which playbooks are ranked by expected loss and automated only under guardrails on confidence and blast radius. The framework is evaluated on a transparent synthetic testbed: a three-region estate of 46 services in a dependency graph with five third-party dependencies, eight disruption classes (zone and region outages, denial of service, ransomware intrusion, bad change, third-party failure, demand surge, compound event) and five operating strategies from manual reactive operations (S0) to AI detection with decision support and guarded automation (S4), with a gradient-boosted incident classifier, an anomaly scorer and a demand forecaster trained on synthetic histories. Over 20 fortnights per scenario, AI detection with decision support (S3) retains 9% of the manual strategy's weighted downtime summed over the disruption classes and 54% excluding ransomware, where it turns a multi-day serial restore into a contained incident in 85% of episodes; bad changes, denial-of-service attacks and compound events fall by 45 to 90% relative to rule-based operations through shorter decision latency and fewer wrong actions, while region outages are bounded by the single-region application tier rather than by the operating strategy. Guarded automation adds a small further gain at a measurable harm rate. One-factor sweeps of 20 parameters and a 120-draw Monte Carlo analysis identify the attacker's dwell and signal strength, the false-alarm budget, the decision latency and the recovery objectives as the drivers. All data are synthetic and the results depend on stated assumptions; the framework is mapped to NIST CSF 2.0, SP 800-61 Rev. 3, SP 800-160, zero trust and U.S. reporting rules, with a phased roadmap.

Keywords

Cloud Resilience; Cybersecurity; Decision Support; AIOpS; Incident Response; Ransomware; Anomaly Detection; Guarded Automation; Simulation; Enterprise IT Operations

Introduction

Enterprise IT operations in the United States now run on a small number of hyperscale cloud platforms, a long tail of software-as-a-service dependencies and an identity layer that connects all of them, so that a single failure can propagate across many organisations at once. The December 2021 event in the US-EAST-1 region of Amazon Web Services impaired a wide range of services that depended on the region's internal network and control plane for several hours (Amazon Web Services, 2021). The faulty content update released by CrowdStrike on 19 July 2024 crashed Windows hosts running its sensor, and Microsoft estimated that 8.5 million devices were affected; airlines, hospitals and retailers lost core systems for a day or more (CrowdStrike, 2024; Weston, 2024). The Colonial Pipeline attack in May 2021 and the Change Healthcare attack in February 2024 showed that an intrusion in one IT estate can become a supply disruption for a sector (Beerman et al., 2023; U.S. Department of Health and Human Services, Office for Civil Rights, 2024), and the SolarWinds compromise showed that the software supply chain itself is an attack path (Peisert et al., 2021). Industry reporting describes the same pattern: fewer but costlier outages, a rising share of breaches that involve third parties, and ransomware as a persistent and expensive threat (Donnellan et al., 2025; Verizon, 2025; IBM, 2025).

The response of U.S. policy has been to make resilience and incident handling a governance obligation. The NIST Cybersecurity Framework 2.0 added a Govern function and placed detection, response and recovery alongside protection (National Institute of Standards and Technology, 2024); NIST SP 800-61 Revision 3 re-cast incident response as a profile of the framework (Nelson et al., 2025); the Securities and Exchange Commission's 2023 rule requires registrants to disclose material cybersecurity incidents and to describe their risk management and governance (U.S. Securities and Exchange Commission, 2023); the Cyber Incident Reporting for Critical Infrastructure Act of 2022 creates reporting duties for covered entities once its implementing rule is in force (U.S. Congress, 2022); the New York Department of Financial Services' amended Part 500, the FTC Safeguards Rule and FedRAMP impose sector-specific and federal-procurement requirements (New York State Department of Financial Services, 2023; Federal Trade Commission, 2021; FedRAMP, n.d.); and for firms that operate in Europe, the Digital Operational Resilience Act adds testing and third-party risk obligations (European Parliament and Council, 2022). These instruments change over time and should be read in their current versions; the point here is that the ability to detect, decide and recover quickly is now a regulated capability rather than only an engineering preference.

At the same time, the technical means of operating cloud estates have changed. Site reliability engineering has made service-level objectives, error budgets and blameless post-incident review standard practice (Beyer et al., 2016; Treynor et al., 2017); chaos engineering has made failure injection a routine test (Basiri et al., 2016); and AI for IT operations (AIOps) has produced a body of work on anomaly detection, incident triage, root-cause analysis and, most recently, large language models that recommend mitigations (Notaro et al., 2021; Dang et al., 2019; Chen et al., 2020; Ahmed et al., 2023; Chen et al., 2024). On the security side, machine learning for intrusion detection is mature in the laboratory and contested in deployment, because low base rates make false alarms the dominant cost (Axelsson, 2000; Sommer & Paxson, 2010; Alahmadi et al., 2022). What is missing is a framework that joins these strands: one in which the same telemetry, models and decision rules serve both the reliability incident (a zone outage, a bad change, a demand surge) and the security incident (a denial-of-service attack, an intrusion that will become ransomware), and in which the decision to act, to recommend or to automate is governed by the confidence of the model, the blast radius of the action and the obligations of the organisation.

This article proposes such a framework and evaluates it on a transparent synthetic testbed. The framework has six layers (estate, telemetry, AI analytics, decision support, response and recovery, governance), an incident lifecycle with explicit timing metrics, and a decision-support workflow in which actions are ranked by expected loss and executed under guardrails. The evaluation builds a synthetic U.S. enterprise cloud estate of three regions, nine availability zones, 46 services in a dependency graph and five third-party dependencies, exposes it to eight disruption classes and operates it under five strategies, from manual reactive operations (S0) to AI detection with decision support and guarded automation (S4). The AI components are real models trained on synthetic histories: a gradient-boosted incident classifier with a calibrated declaration threshold, a multivariate

anomaly scorer and a demand forecaster. The experiments compare the strategies over 20 synthetic fortnights per scenario, sweep 20 parameters one at a time and run a 120-draw Monte Carlo analysis over joint parameter uncertainty.

The research questions are: RQ1, how much of the availability loss, data loss and cost of each disruption class can AI-driven detection, decision support and guarded automation remove relative to rule-based and manual operations, and where does the architecture rather than the operating strategy set the limit? RQ2, how do the timing metrics and their side effects (nuisance incidents, wrong actions, automation harm) change across strategies? RQ3, which parameters of the threat, the estate and the operating model drive the results, and how robust are the strategy rankings to joint uncertainty? RQ4, how should the framework's components be sequenced and governed in a U.S. enterprise, and how do they map to the standards and rules that now require them? Everything quantitative in this article is a property of the synthetic testbed and of the stylised assumptions listed in the methodology; no figure describes any real organisation.

Literature review

Dependability, resilience engineering and resilience metrics

Avizienis and colleagues fixed the vocabulary of dependability (availability, reliability, safety, integrity, maintainability), of its threats (faults, errors, failures) and of its means (prevention, tolerance, removal, forecasting) (Avižienis et al., 2004). Resilience extends dependability from the avoidance of failure to the management of its consequences: in the resilience-engineering tradition a resilient system monitors, anticipates, responds and learns (Hollnagel et al., 2006; Woods, 2015), and in the quantitative tradition resilience is the integral of performance over time through absorption, recovery and adaptation, as in the resilience triangle, the time-dependent metrics of Henry and Ramirez-Marquez and the operational-resilience framework of Ganin and colleagues (Bruneau et al., 2003; Henry & Ramirez-Marquez, 2012; Ganin et al., 2016; Hosseini et al., 2016). Cyber resilience applies these ideas to systems under attack: the resilience matrix of Linkov and colleagues, MITRE's cyber-resiliency engineering framework, NIST SP 800-160 Volume 2 and the network-survivability strategies of defend, detect, remediate, recover and refine translate them into engineering objectives (Linkov et al., 2013; Bodeau & Graubart, 2011; Ross et al., 2021; Sterbenz et al., 2010). The present framework uses the quantitative definitions for its metrics and the engineering frameworks for the structure of its response layer.

Cloud outages, site reliability engineering and recovery-oriented computing

Empirical studies of cloud failures give the testbed its disruption classes (Aditya & Ahmed, 2022; Asheke & Maksudul, 2022; Asheke & Badrul, 2022). Gunawi and colleagues analysed hundreds of public outage reports and found causes spread across upgrades, network, bugs, configuration, load and power, single points of failure in supposedly redundant designs and outages of several hours (Gunawi et al., 2016), and that many cloud bugs are recovery bugs that appear only under failure (Gunawi et al., 2014). Oppenheimer and colleagues found operator error to be the largest cause of Internet-service failure and better operator tooling the most promising remedy (Oppenheimer et al., 2003), from which recovery-oriented computing concluded that systems should be designed to recover quickly rather than only to avoid failure (Patterson et al., 2002). Correlated zone failures dominate the unavailability of distributed storage (Ford et al., 2010), rare slow responses become common at scale (Dean & Barroso, 2013), and Hamilton's rules for Internet-scale services (expect failure, partition, automate, version and roll back) remain the practical canon (Hamilton, 2007). Site reliability engineering turned these into an operating model of service-level objectives (Akter & Sumaiya, 2022; Imran & Jannatul, 2022; Maksudul & Islam, 2022), error budgets, on-call practice, incident command and post-mortems (Beyer et al., 2016); the calculus of service availability bounds a service by its critical dependencies, which is the dependency logic of the testbed (Treynor et al., 2017); meaningful availability argues for user-weighted rather than time-weighted availability, which the testbed adopts (Hauer et al., 2020); chaos engineering injects failures to test the whole socio-technical response (Basiri et al., 2016); and cluster managers such as Borg and Kubernetes supply the control plane whose failure is one of the disruption classes (Verma et al., 2015; Burns et al., 2016). The post-incident reports of the US-EAST-1 event and the Channel File 291 update describe real instances of two classes, a regional control-plane failure and a bad change pushed to a whole fleet (Amazon Web Services, 2021;

CrowdStrike, 2024).

AIOps, incident management and root-cause analysis

AIOps covers the use of machine learning to detect, triage, diagnose and remediate failures in large software systems. Notaro and colleagues' survey finds detection and prediction far better studied than remediation (Notaro et al., 2021); Dang and colleagues set out the challenges from inside a cloud provider, including noisy labels, changing systems and the need for explanations engineers trust (Dang et al., 2019). Studies at Microsoft found incident misrouting common and costly (Chen et al., 2019) and set out the lifecycle of detect (Morshedul & Islam, 2022; Stabak & Shahid, 2022; Feroz & Pervaz, 2022), triage, diagnose and mitigate that the present framework adopts, with mitigation time dominating incident duration (Chen et al., 2020). Automated detection in the "fog of war" of many simultaneous alerts, log-based anomaly detection (Islam & Md, 2023; Asheke, 2023; Feroz & Shabbir, 2022), surveys of log analysis and of anomaly detection and root-cause analysis in microservice systems, and empirical studies of microservice faults describe the detection and diagnosis layer (Li et al., 2021; Du et al., 2017; He et al., 2022a; Soldani & Brogi, 2023; Zhou et al., 2021; Wang et al., 2020). Large language models have recently been applied to recommend root causes and mitigations (Shahid, 2023; Islam & Ahmed, 2023; Sadia & Mosheur, 2023), with promising but uneven accuracy, and the consistency of AIOps models is itself a research subject (Ahmed et al., 2023; Chen et al., 2024; Lyu et al., 2022). The autonomic-computing vision of a closed loop of monitoring, analysis, planning and execution remains the reference point for the question of how much to automate (Kephart & Chess, 2003).

Intrusion detection, security operations and the base-rate problem

Machine learning for intrusion detection has a long record of laboratory success and operational difficulty (Adnan, 2024; Islam, 2024; Ashik, 2024). The base-rate fallacy explains why an accurate detector produces mostly false alarms when attacks are rare (Axelsson, 2000); Sommer and Paxson explained why intrusion detection differs from other machine-learning problems in the cost of errors (Kynatun & Saba, 2024; Maksudul, 2024; Shahid, 2024), the semantic gap and the diversity of normal traffic (Sommer & Paxson, 2010); and the benchmark datasets have known artefacts (McHugh, 2000; Engelen et al., 2021). Surveys catalogue the methods (Buczak & Guven, 2016; Ferrag et al., 2020; Ahmed et al., 2016; Chandola et al., 2009), and isolation forests remain a practical unsupervised scorer (Liu et al., 2008). The analyst's problem is alert volume: alert correlation, classification and adaptive filtering are the classical remedies (Cuppens & Miège, 2002; Valdes & Skinner, 2001; Julisch, 2003; Pietraszek, 2004; Spathoulas & Katsikas, 2010), studies of security operations centres document the burden of low-value alerts and the perception that most alarms are false (Kokulu et al., 2019; Vielberth et al., 2020; Alahmadi et al., 2022), and analyst feedback can retrain detectors with modest labelling (Veeramachaneni et al., 2016). Threat frameworks structure what the detector should look for: the kill-chain taxonomy of advanced persistent threats (Islam, 2024; Sumaiya, 2024; Taher et al., 2024), MITRE ATT&CK and D3FEND, surveys of advanced persistent threats and ransomware, the review of open-source supply-chain attacks and the taxonomies of denial-of-service attacks and defences (Bahrami et al., 2019; Strom et al., 2018; The MITRE Corporation, n.d.; Alshamrani et al., 2019; Oz et al., 2022; Connolly & Wall, 2019; Ohm et al., 2020; Mirkovic & Reiher, 2004; Zargar et al., 2013). Zero-trust architecture limits lateral movement by removing implicit trust in the network (Rose et al., 2020; Syed et al., 2022; He et al., 2022b). The present framework's security path follows these findings: the detector is calibrated to a false-alarm budget, its lateral-movement signals are weak by design, the identity layer is the attacker's main road, and containment is the decision that matters (Ashik, 2025; Kynatun, 2025; Mostari, 2025).

Cyber risk, decision analysis and trust in automation

Decision support needs a model of value. Gordon and Loeb showed that the optimal security investment is a fraction of the expected loss and depends on the vulnerability of the asset (Gordon & Loeb, 2002); multicriteria frameworks structure the trade-offs among availability, data loss, cost and risk (Ganin et al., 2020; Saaty, 1990; Keeney & Raiffa, 1993); attack graphs support reasoning about how an intrusion spreads (Xie et al., 2010; Ou et al., 2006; Lallie et al., 2020); game theory models the adversary (Manshaei et al., 2013); and risk-assessment taxonomies organise the inputs (Shameli-Sendi et al., 2016; Joint Task Force Transformation Initiative, 2012). The costs of cyber events are heavy-tailed and poorly documented (Romanosky, 2016; Eling & Wirfs, 2019; Wolff & Lehr, 2017). The decision to

automate is itself a design decision: the model of types and levels of automation, the analysis of appropriate reliance (Masrul, 2025; Maksudul, 2025; Shahriar, 2025), the review of trust in automation and the evidence on automation bias and complacency all warn that decision aids change the operator's behaviour, that good aids are followed when wrong and that automation leaves the operator with the hardest cases and the least practice (Parasuraman et al., 2000; Lee & See, 2004; Hoff & Bashir, 2015; Parasuraman & Manzey, 2010; Cummings, 2004; Goddard et al., 2012; Bainbridge, 1983; Alberdi et al., 2009). Situation awareness explains why explanations and uncertainty must be part of the aid (Endsley, 1995; Cummings, 2006), and the literature on calibration, interpretability and complementary human-AI performance supplies the technical means (Guo et al., 2017; Rudin, 2019; Lundberg & Lee, 2017; Bansal et al., 2021; Okati et al., 2021). The present framework treats the automation tier (advise, approve, automate) as a policy set by confidence and blast radius and measures its side effects alongside its benefits.

Standards and the U.S. regulatory context

The framework's governance layer maps to a specific set of instruments, summarised in Table 1 and detailed in Section 6: the NIST Cybersecurity Framework 2.0 and its incident-response profile in SP 800-61 Revision 3, the recovery, continuous-monitoring, contingency-planning and control publications (SP 800-184, 800-137, 800-34, 800-53), cyber-resilient systems engineering and zero trust (SP 800-160 Volume 2, SP 800-207, CISA's maturity model and performance goals), the NIST cloud definitions and the provider's shared-responsibility and reliability guidance, the reporting and disclosure instruments (the SEC rule, CIRCIA, Part 500, the Safeguards Rule, FedRAMP), the joint ransomware advisories, the international continuity and incident-management standards, the EU's operational-resilience regulation and threat landscape, and for the AI components the NIST AI Risk Management Framework and the adversarial machine-learning taxonomy (National Institute of Standards and Technology, 2024; Nelson et al., 2025; Bartock et al., 2016; Dempsey et al., 2011; Swanson et al., 2010; Joint Task Force, 2020; Ross et al., 2021; Rose et al., 2020; Cybersecurity and Infrastructure Security Agency, 2023a; Cybersecurity and Infrastructure Security Agency, 2023b; Mell & Grance, 2011; Jansen & Grance, 2011; Amazon Web Services, n.d.; Amazon Web Services, 2024; U.S. Securities and Exchange Commission, 2023; U.S. Congress, 2022; New York State Department of Financial Services, 2023; Federal Trade Commission, 2021; FedRAMP, n.d.; Cybersecurity and Infrastructure Security Agency et al., 2023; Federal Bureau of Investigation et al., 2023; International Organization for Standardization, 2019; ISO/IEC, 2023; European Parliament and Council, 2022; European Union Agency for Cybersecurity, 2024; Tabassi, 2023; Vassilev et al., 2024).

Table 1: Literature strands, what they establish, representative sources and the gap addressed by this article

Strand	What the literature establishes	Representative sources	Gap addressed here
Dependability and resilience engineering	Taxonomy of faults and failures; resilience as monitoring, anticipation, response and learning; quantitative resilience as performance over time through absorb, recover and adapt	Avižienis et al. (2004); Hollnagel et al. (2006); Woods (2015); Bruneau et al. (2003); Henry & Ramirez-Marquez (2012); Ganin et al. (2016); Linkov et al. (2013)	Resilience metrics applied to an AI-operated cloud estate, with detection and decision times as explicit components
Cloud outages, SRE and recovery-oriented computing	Outages come from upgrades, configuration, load, network and correlated zone failures; operator error dominates; design for recovery; SLOs, error budgets and chaos experiments as the operating model	Gunawi et al. (2014, 2016); Oppenheimer et al. (2003); Patterson et al. (2002); Ford et al. (2010); Hamilton (2007); Beyer et al. (2016); Treynor et al. (2017); Hauer et al. (2020); Basiri et al. (2016)	Disruption classes and dependency logic derived from the failure studies; strategies compared on the same estate
AIOps and incident management	Detection and triage are well studied, remediation less; misrouting is costly; mitigation time dominates incident duration; LLMs can recommend root causes with uneven accuracy	Notaro et al. (2021); Dang et al. (2019); Chen et al. (2019, 2020); Li et al. (2021); Ahmed et al. (2023); Chen et al. (2024); Soldani & Brogi (2023); Kephart & Chess (2003)	Detection, classification, decision support and automation evaluated end to end against incident outcomes rather than model accuracy
Intrusion detection and security operations	Low base rates make false alarms the dominant cost; operational data differ from benchmarks; analysts are overwhelmed by alerts; lateral movement is the attacker's road and identity the hub	Axelsson (2000); Sommer & Paxson (2010); McHugh (2000); Alahmadi et al. (2022); Kokulu et al. (2019); Vielberth et al. (2020); Bahrami et al. (2019); Strom et al. (2018); Oz et al. (2022); Rose et al. (2020)	Security detection calibrated to a false-alarm budget inside the same pipeline as reliability detection; containment as the decision that matters
Cyber risk, decision analysis and trust in automation	Invest in proportion to expected loss; multicriteria trade-offs; attack graphs for propagation; heavy-tailed incident costs; levels of automation; automation bias and complacency; calibrated confidence and explanations	Gordon & Loeb (2002); Ganin et al. (2020); Keeney & Raiffa (1993); Xie et al. (2010); Romanosky (2016); Eling & Wirfs (2019); Parasuraman et al. (2000); Lee & See (2004); Parasuraman & Manzey (2010); Guo et al. (2017)	Expected-loss ranking of playbooks with guardrails on confidence and blast radius; wrong actions and automation harm measured as outcomes
Standards and U.S. regulation	CSF 2.0, SP 800-61 Rev. 3, SP 800-160 Vol. 2, SP 800-207, FedRAMP; SEC disclosure, CIRCIA, NYDFS Part 500, Safeguards Rule; ISO 22301 and 27035; DORA for EU operations	National Institute of Standards and Technology (2024); Nelson et al. (2025); Ross et al. (2021); Rose et al. (2020); FedRAMP (n.d.); U.S. Securities and Exchange Commission (2023); U.S. Congress (2022); New York State Department of Financial Services (2023)	Framework components mapped to the functions and obligations of the instruments, with the evidence of the testbed as the justification for sequencing
Simulation methodology	Verification and validation of simulation models; agent-based and discrete-event modelling; global sensitivity analysis	Law (2015); Sargent (2013); Macal & North (2010); Saltelli et al. (2008, 2010)	A transparent synthetic testbed with real trained models, one-factor sweeps and a Monte Carlo analysis over joint uncertainty

Incident lifecycle and timing metrics

Figure 2 shows the lifecycle as a state machine. The estate is in normal operations until a disruption begins; it is then degraded or under attack but not yet declared, and the time in that state is the time to detect. Declaration opens an incident; triage produces a cause hypothesis with a confidence, and the time from declaration to the first action is the decision latency. The guarded-automation branch applies when the hypothesis is one of the low-risk classes, the confidence exceeds the threshold and the blast radius is within the cap; otherwise the recommendation goes to a human for approval. Responding executes the action; if availability does not recover and causes remain, the incident returns to triage, which is how compound events and wrong hypotheses are handled. Recovering covers failovers in flight and restores from backup, where the recovery-time and recovery-point objectives are realised; restoration is declared when weighted availability is back above the service-level threshold and stays there, and the time from first impact to restoration is the time to restore. Security incidents follow the same states, except that lateral movement is a degraded state with no availability loss, so the time to detect is measured from initial access, and containment before encryption avoids the restores entirely.

The framework

Design principles

Five principles follow from the literature. First, one telemetry and one lifecycle for reliability and security incidents: a zone outage and a ransomware intrusion both appear first as anomalies in the same signals and both end in the same recovery actions, so separate pipelines duplicate detection and delay the realisation that an availability problem has a security cause (Shahid, 2025; Islam, 2025; Sadia, 2025). Second, calibrated confidence as the currency of decisions: every model output that can trigger an action is a probability with a known false-alarm rate, and the declaration threshold is set to an explicit false-alarm budget (Ashik, 2026; Shabbir, 2025; Sadmeen, 2025). Third, expected loss as the ranking criterion for actions, which makes a cheap reversible action attractive under uncertainty and an expensive irreversible one (a region failover, an isolation of the identity provider) attractive only when the evidence is strong (Hasib, 2026; Kynatun, 2026; Mostari, 2026). Fourth, guardrails that bound automation by blast radius and confidence, with verification and automatic reversion, so that zero decision latency is bought with a known and small automation-harm rate (Hasinur, 2026; Maksudul, 2026; Mohsin, 2026; Shahriar, 2026). Fifth, the architecture sets the ceiling: no decision rule can restore a service that has no replica, so the framework includes the resilience patterns that the decisions depend on, and the evaluation reports where the ceiling binds (Shahid, 2026a, 2026b; Sadia, 2026).

Reference architecture

Figure 1 shows the reference architecture. Layer 1 is the estate: regions and zones, the service dependency graph, third-party dependencies and the backup and replication substrate. Layer 2 is telemetry: metrics, logs and traces; security telemetry from identity, endpoint detection, egress and administrative changes (Shabbir, 2026; Sumaiya, 2026; Sadmeen, 2026); change and deployment events; provider and third-party status; and the demand calendar. Layer 3 is AI analytics: a multivariate anomaly scorer, an incident classifier with calibrated class probabilities, a demand forecaster, attack-path and blast-radius reasoning over the dependency and identity graph, and explanations with uncertainty (Umme, 2026). Layer 4 is decision support: risk scoring, playbook recommendation ranked by expected loss, guardrails and the automation tier. Layer 5 is response and recovery, from scaling and traffic steering through failover, scrubbing, rollback, isolation and restore to third-party fallback. Layer 6 is governance and assurance: service-level objectives and error budgets, exercises, post-incident reviews, model monitoring and incident reporting. The numbered interfaces are where the evaluation measures something: telemetry quality at interface 1, detection at 2 and 3, decision latency and correctness at 4, action effect at 5 and policy at 6; the side panel lists the standards and rules to which the layers map (Section 6).

Figure 1: Reference architecture of the AI-driven cloud resilience and cybersecurity decision-support framework: six layers, numbered interfaces and the standards and regulatory anchors

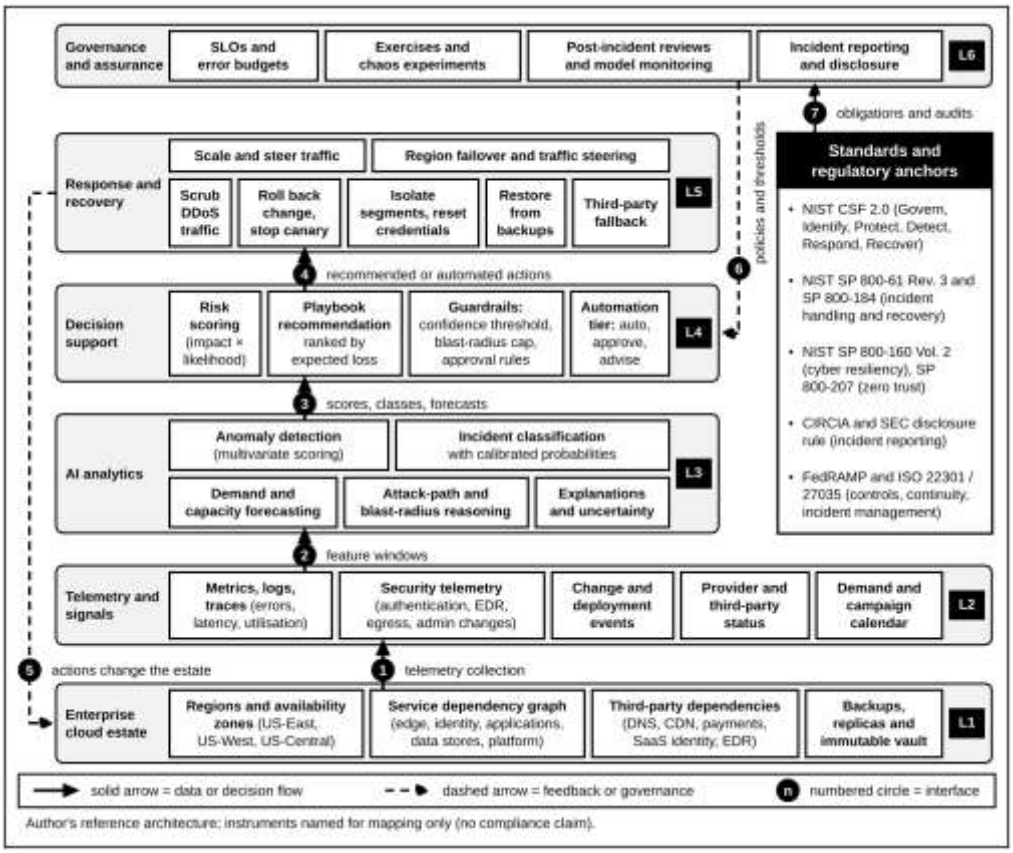
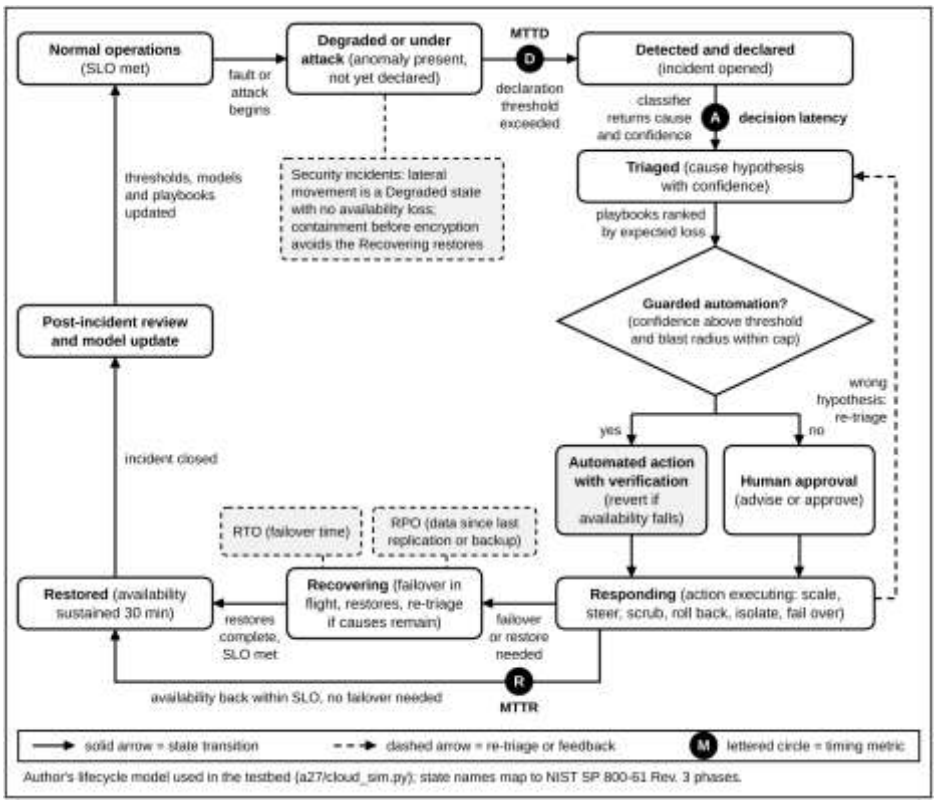


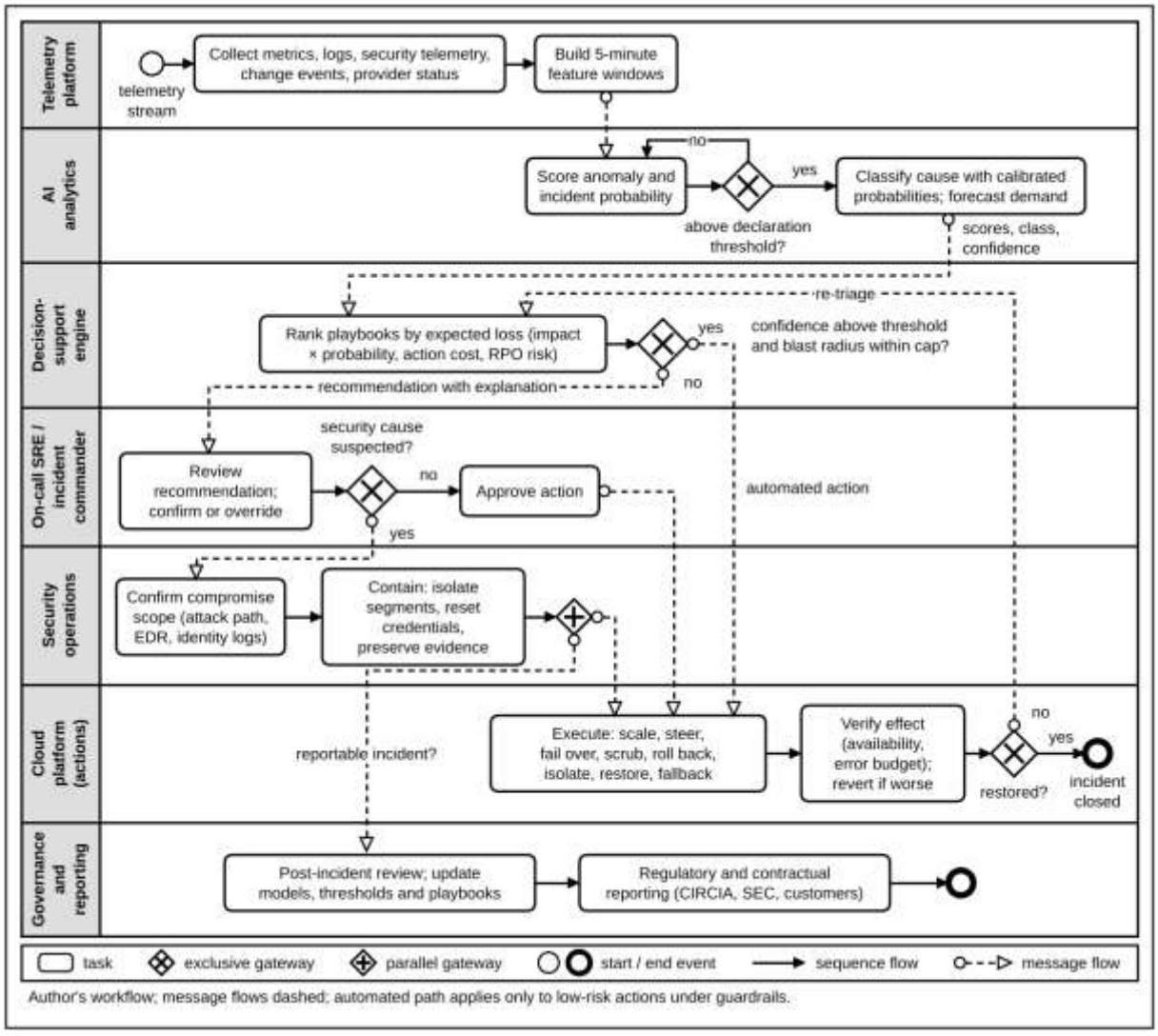
Figure 2: Incident lifecycle and resilience cycle as a state machine with the timing metrics (time to detect, decision latency, time to restore, recovery-time and recovery-point objectives) and the guarded-automation branch



Decision-support workflow

Figure 3 shows the workflow as swimlanes. The telemetry platform builds five-minute feature windows; AI analytics scores each window and, above the declaration threshold, classifies the cause and forecasts demand. The decision-support engine ranks playbooks by expected loss: for each candidate action, the sum over classes of the class probability times the loss if that action is taken when that class is the cause, where the loss includes the weighted availability lost until the cause is resolved, the data lost by an unnecessary failover or restore, the direct cost of the action and its harm when wrong. If the top action is in the automatable set, its class probability exceeds the confidence threshold and its blast radius (the share of weighted availability it can remove if wrong) is within the cap, it executes automatically with verification; otherwise it goes to the on-call engineer with an explanation, who confirms or overrides and routes suspected security causes to security operations, which confirms the compromise scope from the attack graph, endpoint and identity logs, contains it and informs governance of reportable incidents. The cloud platform executes and verifies; if the estate is not restored, the incident is re-triaged; governance runs the post-incident review and the regulatory and contractual reporting.

Figure 3: Decision-support workflow as BPMN-style swimlanes: telemetry, AI analytics, decision-support engine, on-call engineering, security operations, cloud platform actions and governance



Disruption taxonomy

Table 2 gives the eight disruption classes the framework is evaluated against, the mechanism of each, the signals by which it can be seen, the patterns and actions that address it, and the real-world events that motivated it. The classes span the reliability and security sides of the framework and differ in the

information available to the operator: a zone outage is visible in health checks at once and on the provider's status page within tens of minutes; a bad change is visible in error rates but must be connected to the deployment; a demand surge is visible in utilisation and partly predictable from the campaign calendar; a third-party failure is visible in the dependency's error rate and status; a denial-of-service attack is visible in edge utilisation; and an intrusion is visible, before it becomes ransomware, only in weak security signals whose strength grows with the number of compromised systems. The compound event (a bad change, then a demand surge, then a zone loss within two hours) tests whether the lifecycle's re-triage loop separates overlapping causes.

Table 2: Disruption taxonomy of the testbed: eight classes, their mechanisms, visible signals, applicable resilience patterns and actions, and real-world analogues

Class	Mechanism in the testbed	Visible signals	Patterns and actions	Real-world analogue
E1 Zone outage	One US-East zone loses power or network for a median of 3 h; single-zone services fail; multi-zone services lose a third of capacity and rebalance	Health checks at once; provider status after 30 min; errors on legacy services	Multi-zone deployment absorbs; migrate single-zone services (S3+); wait otherwise	Zone-level events in provider histories; correlated failures in Ford et al. (2010)
E2 Region outage	US-East control plane falls to 15% and zones to 30% for a median of 5 h	Control-plane and zone health checks; errors across tiers; provider status	Active-active steering; active-passive failover (RTO, RPO); region-local services wait	US-EAST-1, December 2021 (Amazon Web Services, 2021)
E3 DDoS attack	Edge load multiplied by about 8 for a median of 2.5 h with a 30% application-layer share	Edge utilisation and latency; WAF request rate; errors at the API gateway	Scrubbing (95% of attack load removed after 10 min, 3% collateral loss); rate limiting	Volumetric and application-layer attacks (Mirkovic & Reiher, 2004; Zargar et al., 2013)
E4 Ransomware intrusion	One application service compromised; lateral movement along dependency and identity edges during a dwell with a median of 16 h; then encryption of reachable services	Weak authentication, egress, administrative and EDR anomalies that grow with the compromised count; obvious once encryption starts	Contain (isolate compromised segments, reset credentials); restore encrypted services from backups (RPO = backup age); immutable vault	Colonial Pipeline (Beerman et al., 2023); Change Healthcare (U.S. Department of Health and Human Services, Office for Civil Rights, 2024); ransomware surveys (Oz et al., 2022)
E5 Bad change	A deployment to one tier raises its error rate by about 60% (canary exposes 10% for 20 min where available)	Error rate on the tier, correlated with change recency	Stop canary (automatic in S3+); roll back (15 min); change-correlated triage	Channel File 291 update (CrowdStrike, 2024); configuration and upgrade failures (Gunawi et al., 2016)
E6 Third-party failure	A dependency (DNS, CDN, payment processor, e-mail, EDR vendor) falls to about 10% availability for a median of 2 h	Dependency error rate; third-party status after 40 min	Degraded mode (circuit breakers, S1+); secondary provider (S3+); EDR loss weakens security detection	SaaS and provider dependencies in outage analyses (Donnellan et al., 2025)
E7 Demand surge	Demand rises 3.5 times over 15 min for a median of 5 h; campaign calendar signals it an hour ahead in 70% of episodes	Utilisation and latency; demand ratio; calendar signal	Reactive autoscaling (S1+); predictive pre-scaling (S3+); manual capacity (S0, 2 h)	Load-induced outages (Gunawi et al., 2016); tail latency (Dean & Barroso, 2013)
E8 Compound event	Bad change at onset, surge an hour later, zone loss two hours after onset	Overlapping error, utilisation and health-check signals	Re-triage loop; actions for each cause in impact order	Overlapping causes in post-incident reports; fog of war (Li et al., 2021)

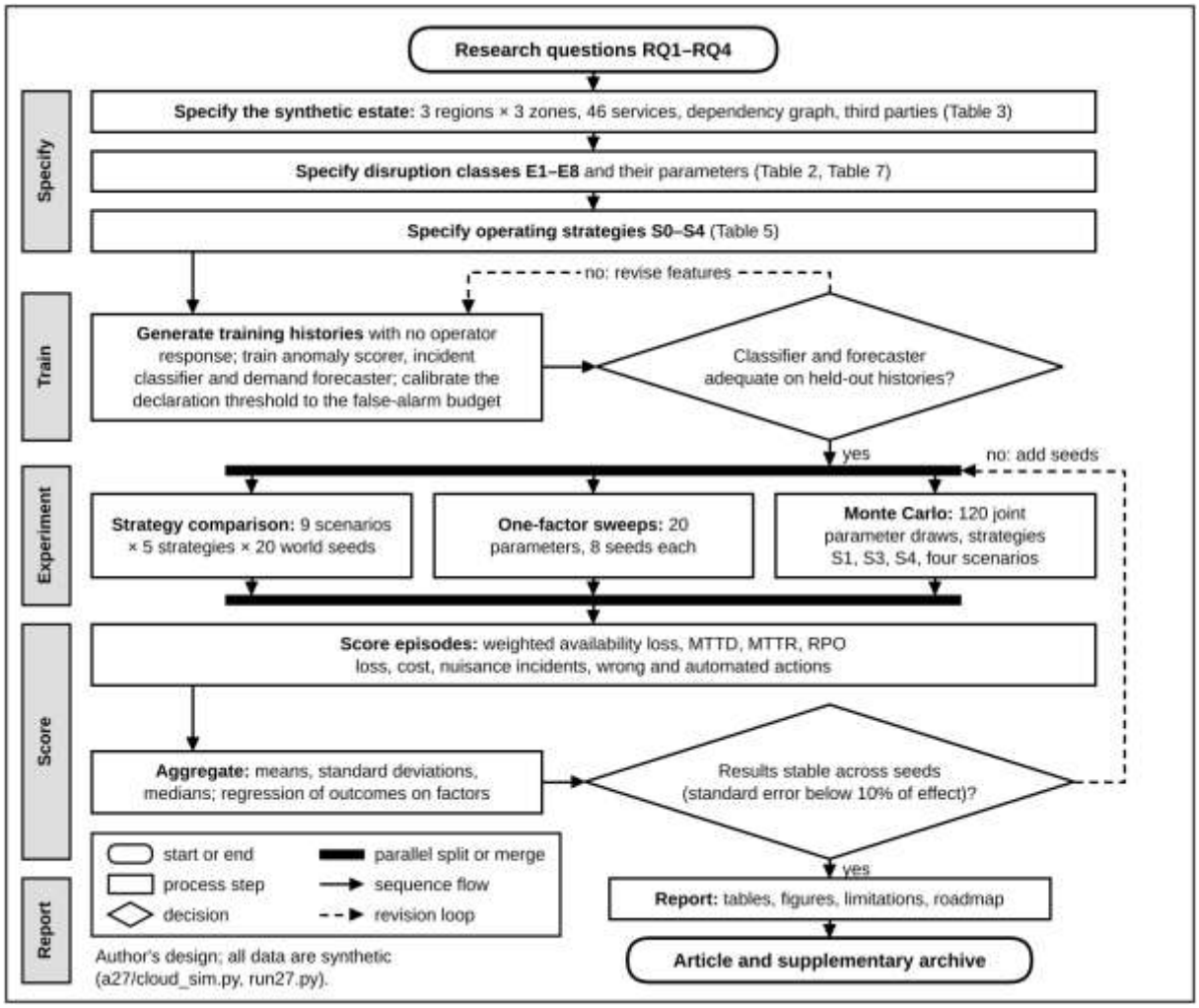
Methodology

Research design

The study is a simulation-based evaluation of the framework against explicit alternatives, in the tradition of simulation modelling and of agent-based computational models (Law, 2015; Macal &

North, 2010; Epstein, 1999). Figure 4 shows the design. A synthetic enterprise cloud estate is specified (Table 3), together with the eight disruption classes of Table 2, their parameters and the five operating strategies. The AI components are trained on synthetic histories produced by the same simulator with no operator response, so that the models never see the evaluation episodes, and the declaration threshold is calibrated to a false-alarm budget on normal fortnights. Three experiment families follow: a strategy comparison of nine scenarios (eight disruption classes and a normal fortnight) by five strategies over 20 world seeds, one-factor sweeps of 20 parameters over eight seeds each, and a Monte Carlo analysis of 120 joint draws of 14 parameters with strategies S1, S3 and S4 on the four most consequential classes. Each episode is a fortnight at five-minute resolution (4,032 steps); the disruption starts on days three to eight, so that six to eleven days remain for recovery. Verification and validation follow the simulation literature: code assertions and traced episodes for verification, and face validation against the published failure studies and post-incident reports for the behaviour of the estate (Sargent, 2013; Gunawi et al., 2016; Amazon Web Services, 2021). The simulator (a27/cloud_sim.py), the experiment runner and the raw results are in the supplementary archive.

Figure 4: Study design: testbed specification, model training and calibration, the three experiment families, scoring and reporting



The synthetic estate

Table 3 specifies the estate: a stylised U.S. enterprise with a primary region (US-East), a secondary region (US-West) and a tertiary region for analytics and the immutable backup vault (US-Central), each with three availability zones. Forty-six services form a dependency graph across an edge tier (content delivery, DNS, web application firewall, API gateway), an identity tier (identity provider, directory, secrets), twenty customer-facing and internal applications (web store, mobile back end, checkout,

orders, payments, fraud, inventory, fulfilment, customer relationship management, billing, notifications, support portal, partner API, analytics, reporting, administration console and their companions), thirteen data stores, three platform services (message queue, machine-learning platform, observability) and five third-party dependencies (DNS provider, workforce single sign-on, payment processor, e-mail service, endpoint detection vendor). Each service has a deployment pattern: legacy single-zone in US-East (fulfilment, billing and their databases), multi-zone in US-East (most applications, the cache, queue and search index), active-passive across US-East and US-West with asynchronous replication (identity, payments, the transactional databases, customer relationship management), active-active across the two regions with health-based traffic steering (edge, web store, mobile back end, cart database, object store, observability) or tertiary-region (analytics). Customer-facing services carry revenue weights that sum to one (checkout 0.20, web store 0.17, mobile back end 0.12, partner API 0.10, fulfilment 0.08, billing 0.06 and so on), and the weighted availability of the estate is the weight-averaged availability of these services, each the product of its own health, its capacity success fraction and the availability of its hard dependencies, with soft dependencies degrading to a fallback factor; this is the dependency logic of the calculus of service availability (Treynor et al., 2017) and the user-weighted logic of meaningful availability (Hauer et al., 2020). Demand follows a daily and weekly pattern with noise and unannounced bursts; load-sensitive services succeed on the fraction of requests their capacity can serve; and data stores have write rates (orders and payments 150 per minute, inventory 80, cart 60, and so on) from which lost writes are computed when a failover or a restore loses the writes since the last replication or backup.

Table 3: Specification of the synthetic enterprise cloud estate: regions, zones, service tiers and deployment patterns, dependency rules, demand, data and telemetry

Element	Specification	Rationale
Regions and zones	US-East (primary), US-West (secondary), US-Central (analytics and immutable backup vault); three availability zones each	Typical U.S. enterprise footprint on a hyperscale provider (Mell & Grance, 2011; Amazon Web Services, n.d.)
Services	46 services: edge 4, identity 3, applications 20, data stores 13, platform 3, third parties 5 (workforce SSO, DNS provider, payment processor, e-mail service, EDR vendor); third-party dependencies modelled as availability paths	Dependency structure of a commerce-type enterprise estate; third parties as the supply-chain surface (Ohm et al., 2020)
Deployment patterns	Single-zone legacy (fulfilment, billing and their databases); multi-zone US-East (most applications, cache, queue, search index, fraud and warehouse data); active-passive US-East/US-West with asynchronous replication (identity, secrets, payments, transactional databases, CRM); active-active with health-based steering (edge, web store, mobile back end, cart database, object store, observability); tertiary region (analytics, data lake, vault)	Mixed estates are the norm; the patterns set the ceiling of what any strategy can restore (Hamilton, 2007; Amazon Web Services, 2024)
Availability rule	Service availability = own health x capacity success fraction x product of hard-dependency availabilities x product over soft dependencies of (fallback + (1 - fallback) x availability); evaluated in topological order each step	Calculus of service availability (Treynor et al., 2017)
Weighted availability	Revenue weights on customer-facing services (checkout 0.20, web store 0.17, mobile 0.12, partner API 0.10, fulfilment 0.08, billing 0.06, CRM and support 0.04 each, notifications 0.03, analytics, reporting and administration 0.02 each); weighted downtime = integral of (1 - weighted availability)	User-weighted availability (Hauer et al., 2020)

Element	Specification	Rationale
Capacity and demand	Demand with daily and weekly pattern, slow drift, 4% noise and about two unannounced 1.5x bursts a week, normalised to a daily peak of 1.0; nominal capacity 1.4 x peak; autoscaling +35% per 10-min step (S1+); predictive pre-scaling to the one-hour forecast (S3+); multi-zone capacity falls by a third per lost zone and rebalances to 1.5x	Headroom and scaling behaviour of cloud platforms (Verma et al., 2015; Burns et al., 2016)
Data	Write rates per data store (orders and payments 150/min, inventory 80, cart 60, warehouse 40, billing and CRM 30, identity 20, object store 20, search 10, catalogue 5); replication lag 2 min (S1+) or 4-h snapshots (S0); backups daily (S0, S1), 6-hourly (S2), hourly immutable (S3, S4)	Recovery-point objectives as data loss (Swanson et al., 2010; Bartock et al., 2016)
Telemetry	23 raw signals per 5-min step: tier error rates, latencies, utilisations, demand ratio, zone and control-plane health, change recency, authentication, egress, administrative and EDR anomalies, provider and third-party status, minimum third-party availability, edge request rate; multiplicative noise 5%, additive 1%, stale windows when observability is degraded	Continuous monitoring (Dempsey et al., 2011); signal design after SOC studies (Vielberth et al., 2020)
Time	5-minute steps; 14-day episodes (4,032 steps); first day excluded; onset on days 3 to 8; metrics over the remainder of the fortnight	Enough horizon for serial restores; incident timescales of hours to days (Gunawi et al., 2016)
Background events	Third-party blips (availability 0.9 for 10-15 min, about two a fortnight), zone blips (health 0.8, about one a fortnight), about ten benign deployments a fortnight	Nuisance signals that a detector must tolerate (Alahmadi et al., 2022)

Disruption classes and their dynamics

Each disruption class is a process in the simulator whose parameters are listed with the experimental design below. A zone outage (E1) sets one US-East zone's health to zero for a log-normal duration with a median of three hours; single-zone services in that zone fail, multi-zone services lose a third of their capacity and rebalance, and the provider's status page reports the zone after 30 minutes. A region outage (E2) reduces the US-East control plane to 15% and all its zones to 30% health for a median of five hours, which fails the region-local services, requires the active-passive services to be failed over (with a strategy-specific recovery-time objective and a data loss equal to the replication lag) and steers the active-active services to US-West at a strategy-specific rate. A distributed denial-of-service attack (E3) multiplies edge load by a factor with a median of eight for a median of 2.5 hours, with a 30% application-layer component; scrubbing removes 95% of the attack load after ten minutes at a collateral cost of 3% of legitimate edge requests. A ransomware intrusion (E4) begins with the compromise of one application service and spreads along the dependency and identity graph with a per-step probability per edge (lower into the identity provider, which, once compromised, reaches everything); during this dwell phase, authentication, egress, administrative-change and endpoint-detection signals rise in proportion to the number of compromised services; at the end of the dwell (median 16 hours, log-normal, clipped to 6 to 36 hours) the attacker encrypts the reachable application, data and platform services at six per five minutes, and the encrypted services are unavailable until restored from backup, with data loss equal to the age of the backup. A bad change (E5) deploys an error-inducing version to a randomly chosen tier with an impact drawn around 0.6; with a canary, only 10% of traffic is exposed for the first 20 minutes. A third-party failure (E6) reduces a randomly chosen dependency's availability to about 0.1 for a median of two hours. A demand surge (E7) multiplies demand by 3.5 with a 15-minute ramp for a median of five hours, with a campaign-calendar signal 60 minutes ahead in 70% of episodes. The compound event (E8) is a bad change at onset, a demand surge an hour later and a zone loss two hours after onset. Background noise is present in every episode (short third-party and zone blips, benign deployments, unannounced demand bursts), so that detectors face nuisance signals.

AI components

Table 4 specifies the AI components and their training data. The incident classifier is a gradient-boosted decision-tree model (LightGBM, 160 trees, 15 leaves, balanced class weights) that maps a 30-feature window (tier error rates, latencies and utilisations, demand ratio, zone and control-plane health checks, change recency, the four security signals, provider status, third-party availability, edge request rate, and 15- and 30-minute changes and means) to probabilities over nine classes (normal and the eight disruption classes) (Ke et al., 2017). It is trained on 36 synthetic fortnights with no operator response, with the disruption parameters varied across their Monte Carlo ranges so that the model does not learn one attack size; the lateral-movement phase of an intrusion is labelled as an incident only once at least three services are compromised, because before that the signals are indistinguishable from noise; one training set is generated per world seed, and the evaluation episodes use different seeds. The anomaly scorer is the robust Mahalanobis distance of the window from the distribution of normal-operations windows, scaled against its 99th percentile on normal data (Chandola et al., 2009). The incident score blends the classifier's incident probability (0.75) with the anomaly score (0.25), and the declaration threshold is calibrated on three further normal fortnights to a budget of two declarations per week, which is the base-rate discipline the intrusion-detection literature recommends (Axelsson, 2000; Sommer & Paxson, 2010); the confidence of the top cause is its probability divided by the incident probability, and because the calibration of gradient-boosted models is imperfect the confidence-threshold sweep examines the consequences (Guo et al., 2017). A security detection counts as evidence of compromise, and so can trigger containment, only when the classifier names the intrusion class with confidence of at least 0.5 and the 30-minute means of the security signals are themselves elevated; without that rule a detector could contain an intrusion it had not really seen. The demand forecaster is a gradient-boosted regressor of demand one hour ahead from recent ratios, time of day, day of week and the campaign signal; S3 and S4 pre-scale to its forecast. The decision-support engine ranks the unresolved causes by impact, recommends the matching playbook and, in S4, executes the low-risk playbooks (scale, scrub, roll back, migrate from a lost zone), whose blast radius is within 20% of weighted availability by construction, automatically when the top-class confidence is at least 0.85; failover, isolation and restores always require approval, and an automated action on a wrong hypothesis costs its harm and a 20-minute penalty before re-triage, which is how verification and reversion are represented. Its effect on the human is modelled through the decision latency, the probability of acting on a wrong hypothesis and the probability of following a wrong classification, all lower with decision support, following the evidence on complementary human-AI decision making and on automation bias (Bansal et al., 2021; Parasuraman & Manzey, 2010; Goddard et al., 2012).

Table 4: AI components of the framework as implemented in the testbed: inputs, model, training data, outputs, calibration and the strategies that use them

Component	Inputs	Model and training	Outputs and calibration	Used by
Incident classifier	30-feature window: 23 raw signals plus 15-min changes and 30-min means of error, authentication, egress, administrative and EDR signals	LightGBM, 160 trees, 15 leaves, balanced class weights; 36 synthetic fortnights per world seed with no operator response, disruption parameters varied across the Monte Carlo ranges; samples every 10 min	Probabilities over nine classes; confidence of the top cause = its probability / (1 - P(normal)); revised at 15, 30 and 60 min after declaration; an intrusion call counts as evidence only with confidence >= 0.5 and elevated 30-min security means	S2, S3, S4

Anomaly scorer	Same feature window	Robust Mahalanobis distance from the normal-operations distribution (minimum covariance determinant on up to 6,000 normal windows)	Score scaled against the 99th percentile on normal data, clipped to [0, 1]	S2, S3, S4
Incident score and declaration	Classifier incident probability and anomaly score	Blend 0.75 / 0.25; threshold found by bisection on three normal fortnights	Declaration at a budget of 2 incidents per week (default), re-armed after six quiet evaluations; rules kept as a backstop	S2, S3, S4
Demand forecaster	Trailing-hour and day-ago ratios, time of day, day of week, campaign signal now and in one hour	LightGBM regressor, 200 trees; same histories	Demand one hour ahead relative to the trailing hour; capacity target = 1.4 x forecast	S3, S4
Rule-based detector	Raw signals	Thresholds: error rate > 5%, latency > 2x, utilisation > 90%, third-party availability < 50%, health check < 0.5; security correlation over 15-min means: authentication z > 1.4 with administrative z > 1.0, or EDR alerts > 2.7 per 5 min; two consecutive windows	Binary alert, no cause	S1 (primary); S2-S4 (backstop)
Decision-support engine	Open causes, class probabilities, action effects and costs	Ranks unresolved causes by impact; recommends the matching playbook with explanation; re-triages when availability does not recover	Decision latency median 12 min; wrong-hypothesis probability 8%; automation-bias probability 40%	S3, S4
Guarded automation	Top class, its confidence, blast radius of the action	Automatable set: scale, scrub, roll back, migrate from a lost zone (blast radius within 20% of weighted availability by construction); confidence >= 0.85; verification and reversion represented as a 20-min penalty before re-triage	Zero decision latency for automated actions; 8-min approval wait otherwise; automated wrong actions counted	S4

Operating strategies

Table 5 defines the five strategies as a capability matrix; they are cumulative so that the marginal contribution of each layer can be read from adjacent columns, and the sweeps separate the contributions of individual capabilities. S0 (manual, reactive) detects an incident when weighted availability has been below 0.90 for 15 minutes or when complaints accumulate, has no security detection beyond chance discovery and the obvious signal of encryption, triages with a median latency of 60 minutes and a 35% chance of acting first on a wrong hypothesis, raises fixed capacity manually two hours into an incident, has no canary and no third-party fallback, fails over a region manually in six hours from four-hour snapshots, restores daily backups serially at three hours each and isolates compromised services for 90 minutes. S1 (rule-based monitoring and runbooks) adds threshold alerting, a 15-minute correlation rule on the security signals, runbooks (35-minute latency, 25% wrong hypotheses), reactive autoscaling, manual canaries, degraded-mode fallbacks, a two-hour runbook failover with two-minute asynchronous replication, daily backups restored two at a time and a 60-

minute isolation window. S2 (AI detection, manual decisions) replaces the detection rules with the calibrated classifier and anomaly scorer, keeping the rules as a backstop, but keeps human triage (25-minute latency, 15% wrong hypotheses, a 70% chance of following a wrong classification) and six-hourly backups. S3 (AI detection and decision support) adds the expected-loss ranking with explanations (12-minute latency, 8% wrong hypotheses, 40% automation bias), predictive autoscaling, automatic canary analysis, secondary-provider fallbacks, orchestrated 30-minute failover, hourly immutable backups restored four at a time, 30-minute isolation and migration of single-zone services out of a lost zone. S4 (guarded automation) is S3 with automatic execution of the low-risk playbooks at zero decision latency when the guardrails are satisfied, an eight-minute approval wait otherwise, and verification and reversion of automated actions.

Table 5: Operating strategies S0 to S4 as a capability matrix: detection, security detection, triage and decision latency, capacity, change management, fallbacks, disaster recovery, backups and restores, containment and automation

Capability	S0 Manual, reactive	S1 Rules and runbooks	S2 AI detection	S3 AI decision support	S4 Guarded automation
Availability detection	Weighted availability < 0.90 for 15 min, or complaints	Threshold rules, 10-min windows	Calibrated classifier and anomaly score; rules as backstop	As S2	As S2
Security detection	Chance discovery; encryption	15-min correlation of authentication and administrative anomalies; EDR bursts	Classifier on security signals with evidence rule; rules as backstop	As S2	As S2
Decision latency (median)	60 min	35 min	25 min	12 min	0 min automated; 8 min approval
Wrong first hypothesis	35%	25%	15% (follows a wrong classification 70%)	8% (follows a wrong classification 40%)	As S3
Capacity	Fixed; manual 2.5x after 2 h	Reactive autoscaling	Reactive autoscaling	Predictive pre-scaling (forecaster)	As S3
Change management	No canary	Manual canary (10% for 20 min)	Manual canary	Automatic canary analysis	As S3
Third-party fallback	None	Degraded mode (0.5)	Degraded mode	Secondary provider (0.95) after 15 min	As S3
Region failover (RTO)	Manual, 6 h	Runbook, 2 h	Runbook, 2 h	Orchestrated, 30 min	As S3
Replication (RPO at failover)	4-h snapshots	2-min asynchronous	2-min asynchronous	2-min asynchronous	As S3
Traffic steering (active-active)	90 min	10 min	10 min	5 min	5 min
Backups and restores	Daily; serial, 3 h each	Daily; 2 in parallel, 90 min	6-hourly; 2 in parallel	Hourly immutable; 4 in parallel	As S3
Containment window	90 min	60 min	60 min	30 min	30 min

Capability	S0 Manual, reactive	S1 Rules and runbooks	S2 AI detection	S3 AI decision support	S4 Guarded automation
Zone migration of legacy services	No	No	No	Yes (45 min)	Yes
Automation	None	Autoscaling only	Autoscaling only	Autoscaling, canary	Low-risk playbooks under guardrails

Metrics

Table 6 defines the metrics. The primary outcome is weighted downtime: the integral of one minus weighted availability over the post-onset window, in minutes of full-estate-equivalent outage (60 minutes may be one hour at zero availability or ten hours at 90%). Time to detect is measured from onset to the first declaration while the disruption is active, and for the intrusion from initial access to the first security detection; first action is the time to the first correct action; time to restore runs from first impact to weighted availability sustained above 0.995 for 30 minutes, censored at the end of the fortnight. Blast radius, lost writes, services restored, nuisance and blip incidents, wrong actions, automated actions and automated wrong actions, and for the intrusion the containment share and the compromised and encrypted counts complete the set. Cost combines revenue loss at a stylised $10,000 per minute of weighted downtime, incident-team labour at $1,800 per hour (three people in S0 and S1, two with AI support), scrubbing at $2,000 per hour, extra capacity at $4,000 per hour of estate-wide capacity above nominal, restores at $25,000 per service and lost writes at $35 each; these are assumptions of the order reported for large enterprises, and the cost is reported by component so that the reader can re-weight it (Romanosky, 2016; Eling & Wirfs, 2019; IBM, 2025).

Table 6: Outcome metrics: definitions, units, the window over which each is measured and the cost assumptions

Metric	Definition	Unit and window	Notes
Weighted downtime	Integral of (1 - weighted availability) from onset to the end of the fortnight	Minutes of full-estate outage equivalent; 6-11 days	Primary outcome; 60 min may be one hour at zero availability or ten hours at 90%
Mean weighted availability	Mean of weighted availability over the same window	Share	Companion to weighted downtime
Blast radius	Largest share of weighted availability lost at any moment after onset	Share	Severity of the worst moment
Time to detect (MTTD)	Onset to the first declaration while the disruption is active; for the intrusion, initial access to the first security detection	Minutes; share of episodes detected	Canary stops count as detections
First action	Onset to the first correct action	Minutes	Includes decision latency and wrong-hypothesis penalties
Time to restore (MTTR)	First impact to weighted availability >= 0.995 sustained 30 min	Minutes; censored at the end of the fortnight	Zero when the disruption causes no impact
Lost writes and restores	Write rate x replication lag at failover, plus write rate x backup age at restore; number of services restored	Writes; count	Realised recovery-point loss
Intrusion outcomes	Share contained before encryption; compromised and encrypted services; time to security detection	Share; count; minutes	Reported for E4 only

Metric	Definition	Unit and window	Notes
Nuisance and blip incidents	Declarations with weighted availability >= 0.99 outside the disruption; declarations on background blips	Count per fortnight	The cost of sensitivity
Wrong and automated actions	Actions on a hypothesis that was not an open cause; automated actions and automated wrong actions (S4)	Count per episode	Automation harm = automated wrong actions
Cost	Revenue $10,000 per minute of weighted downtime + labour $1,800 per hour of open incident (3 people S0-S1, 2 otherwise) + scrubbing $2,000 per hour + extra capacity $4,000 per estate-hour above nominal + restores $25,000 each + lost writes $35 each	US dollars (millions)	Stylised assumptions, reported by component so they can be re-weighted

Parameters and experimental design

Table 7 lists the assumed parameters with their default values and the ranges used in the sweeps and the Monte Carlo analysis. The sweeps vary one parameter at a time over eight seeds with the strategies and scenarios most sensitive to it: detection and decision parameters (the false-alarm budget, the attacker's dwell, spread and signal strength, a multiplier on decision latency, the automation confidence threshold, observability quality and telemetry noise) in group A, and architecture and recovery parameters (the recovery-time objective, replication lag, backup interval, restore parallelism and isolation window of S3, and the sizes of the region outage, attack, surge, bad change and third-party failure) in group B. The Monte Carlo analysis draws 14 parameters jointly and uniformly from their ranges, trains the models afresh for each draw and runs S1, S3 and S4 on the region outage, denial-of-service, intrusion and bad-change scenarios; the outcome differences between strategies are then regressed on the standardised factors, and the share of draws in which each strategy ranks first is reported, following the practice of global sensitivity analysis (Saltelli et al., 2008; Saltelli et al., 2010). All random numbers are seeded, and the supplementary archive reproduces every table.

Table 7: Assumed parameters of the testbed with default values, the sweep ranges and the Monte Carlo ranges

Parameter group	Names (a27/cloud_sim.py)	Default	Sweep values	Monte Carlo range
Zone outage duration	e1_dur	180 min (log-normal, sd 0.5)	–	–
Region outage duration; residual health	e2_dur; e2_cp, e2_infra	300 min; 0.15, 0.30	120, 300, 600	120-600
DDoS load multiple; duration; app-layer share	e3_mult; e3_dur; e3_app	8; 150 min; 0.3	3, 8, 15	3-15
Attacker dwell (median, clipped 360-2,160); spread per edge per step; into IdP; signal strength	e4_dwell; e4_spread; e4_idp; e4_sig	960 min; 0.008; 0.004; 1.0	480-2,880; 0.004-0.016; –; 0.5-1.5	480-2,880; 0.004-0.016; –; 0.5-1.5
Bad change impact; canary share and length	e5_impact; e5_canary_share, e5_canary_min	0.6; 0.1, 20 min	0.2, 0.6, 1.0	0.2-1.0
Third-party residual availability; duration	e6_av; e6_dur	0.1; 120 min	0.0, 0.1, 0.4	0.0-0.4
Security evidence rule (30-min means): combined z; EDR per 5 min	sec_ev; sec_edr	1.0; 1.3	–	–

Parameter group	Names (a27/cloud_sim.py)	Default	Sweep values	Monte Carlo range
Multi-region application tier (architecture variant)	mr_app	0 (multi-zone)	0, 1	–
Surge multiple; ramp; duration; lead signal probability	e7_mult; e7_ramp; e7_dur; e7_lead_p	3.5; 15 min; 300 min; 0.7	2, 3.5, 5; lead 0, 0.7, 1	2-5
Capacity headroom; autoscaling step and lag; manual scaling lag	headroom; scale_step, scale_lag; manual_scale_lag	1.4; 0.35, 10 min; 120 min	–	–
Replication lag; S0 snapshot interval	repl_lag; snap_int	2 min; 240 min	1, 2, 10, 30	–
Backup interval by strategy	backup_int	1,440, 1,440, 360, 60, 60 min	S3: 15, 60, 360, 1,440	S3: 15-1,440
Restore time and parallelism	restore_min, restore_par; restore_min_s0	90 min, 4 (S3+); 180 min serial (S0)	parallel 1, 2, 4, 8	–
Failover RTO by strategy; steering lag	rto; steer_lag	360, 120, 120, 30, 30 min; 90, 10, 10, 5, 5 min	S3: 15, 30, 60, 120	S3: 15-120
Decision latency medians; log-sd; approval wait	dec_med; dec_sd; approve	60, 35, 25, 12, 12 min; 0.45; 8 min	multiplier 0.5, 1, 2	0.5-2
Wrong-hypothesis probability; penalty	wrong; wrong_pen	0.35, 0.25, 0.15, 0.08, 0.08; 20 min	–	–
False-alarm budget; automation threshold and blast-radius cap	fa_week; auto_thr, auto_cap	2 per week; 0.85, 0.20	0.5-8; 0.6-0.95	0.5-4; –
Observability quality; telemetry noise	obs_quality; noise	1.0; 1.0	0.4, 0.7, 1.0; 0.5, 1, 2	0.5-1.0; 0.5-2
Containment window by strategy	isol_min	90, 60, 60, 30, 30 min	S3: 15, 30, 60, 120	–
Scrubbing efficiency, lag, collateral	scrub_eff, scrub_lag, scrub_drop	0.95, 10 min, 0.03	0.8, 0.95, 0.99	–
Rollback time; fallback switch time and degraded factor	rollback_min; fallback_min, fallback_f	15 min; 15 min, 0.5	–	–
Cost rates	rev_pm, labour_h, scrub_h, cap_h, restore_cost, loss_tx	$10,000/min; $1,800/h; $2,000/h; $400/h per 10%; $25,000; $35	–	–

Results

Strategy comparison

Table 8 and Figure 5 give the primary outcome, weighted downtime in minutes of full-estate-equivalent outage, for the nine scenarios and five strategies, averaged over 20 world seeds. Four patterns stand out. First, the ransomware intrusion dominates the manual baseline: S0 loses 3,736 weighted minutes (about 2.6 days of full outage in a window of six to eleven days) because no intrusion is contained before encryption, 24 services on average are encrypted and are restored serially at three hours each from day-old backups. Rule-based security correlation (S1) contains 70% of intrusions before encryption and cuts the loss to 144 minutes; AI detection (S2) contains 85% and cuts it to 47; decision support (S3) halves it again to 21 through the shorter containment window and parallel restores. Second, the region outage is the one scenario in which the strategies barely differ: 439 minutes for S0 and 338 to 347 for the others, because the application tier, the cache, the queue and most data stores are deployed only in the primary region, so that weighted availability falls to near zero for the whole outage whatever the operator does, and the failover of the active-passive data stores only matters for

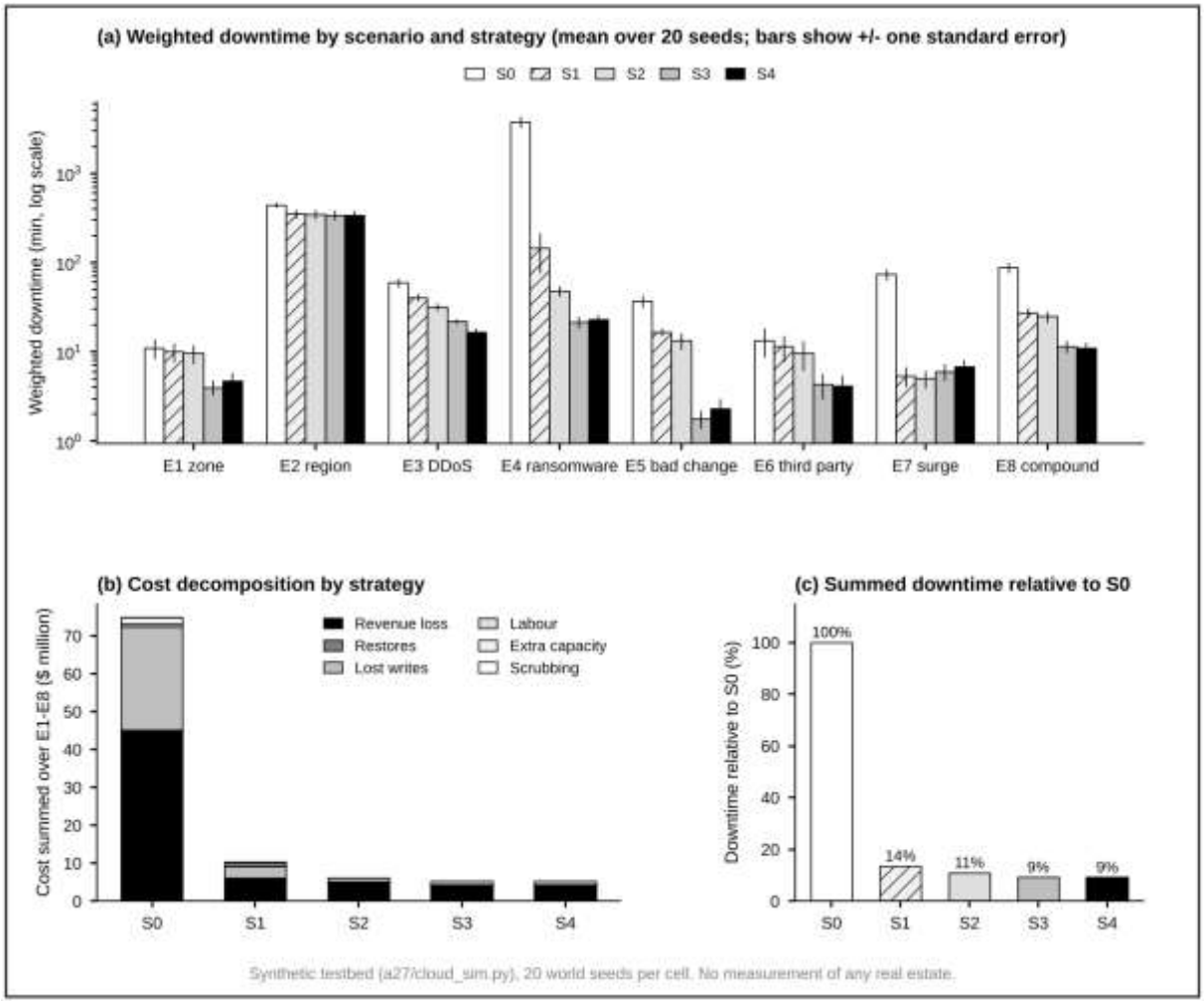
the tail. Third, for the fast-moving scenarios the gains come from decision latency and the right first action: the bad change falls from 37 minutes (S0) to 17 (S1), 13 (S2) and 2 (S3 and S4, where automatic canary analysis stops the rollout at 10% exposure); the denial-of-service attack from 59 to 41, 32, 22 and 17; the compound event from 87 to 27, 25, 11 and 11; the third-party failure from 13 to 11, 10, 4 and 4 (the secondary provider of S3 is the lever; S1's degraded mode gains little because its wrong first actions cost about what its fallback saves); and the demand surge from 73 to 5, 5, 6 and 7, almost all of it from reactive autoscaling alone. Fourth, the zone outage is small for every strategy (4 to 11 minutes) because the multi-zone patterns absorb it and only the two legacy single-zone services and their databases are lost; S3's zone migration removes most of that.

Summed over the eight disruption scenarios, S1 retains 14% of S0's weighted downtime, S2 11%, S3 9% and S4 9%; excluding the ransomware scenario, which dominates S0, the shares are 64%, 61%, 54% and 53%. Cost follows the same order: $74.7 million summed over the eight scenarios for S0 (of which $44.6 million is revenue loss and $27.9 million lost writes and restores, nearly all from the intrusion), $10.1 million for S1, $6.0 million for S2, $5.1 million for S3 and $5.1 million for S4; the normal fortnight costs $0.03 million under S0 and $0.10 to 0.11 million under the others, which is the price of autoscaling headroom and of the nuisance incidents the detectors raise. S4's guarded automation adds a small gain over S3 for the denial-of-service attack (17 against 22 minutes), where the automated scrub starts five minutes after declaration, and a small loss for the surge and the zone outage, where its automated wrong actions cost a few minutes; at the default confidence threshold the two strategies are indistinguishable in the sum.

Table 8: Weighted downtime (minutes of full-estate-equivalent outage; mean and standard deviation over 20 world seeds) by scenario and strategy, with the sum over the disruption scenarios and its ratio to S0

Scenario	S0 Manual	S1 Rules	S2 AI detection	S3 Decision support	S4 Guarded automation
Normal fortnight	2 (5)	1 (1)	1 (1)	1 (0)	1 (1)
E1 Zone outage	11 (11)	10 (10)	10 (10)	4 (3)	5 (5)
E2 Region outage	439 (117)	347 (157)	345 (158)	338 (160)	338 (160)
E3 DDoS attack	59 (24)	41 (13)	32 (11)	22 (6)	17 (6)
E4 Ransomware intrusion	3736 (2159)	144 (300)	47 (26)	21 (13)	23 (12)
E5 Bad change	37 (24)	17 (7)	13 (12)	2 (2)	2 (3)
E6 Third-party failure	13 (20)	11 (16)	10 (15)	4 (6)	4 (5)
E7 Demand surge	73 (43)	5 (6)	5 (5)	6 (5)	7 (5)
E8 Compound event	87 (46)	27 (13)	25 (14)	11 (7)	11 (7)
Sum over E1-E8	4455	602	486	408	406
Relative to S0	100%	14%	11%	9%	9%

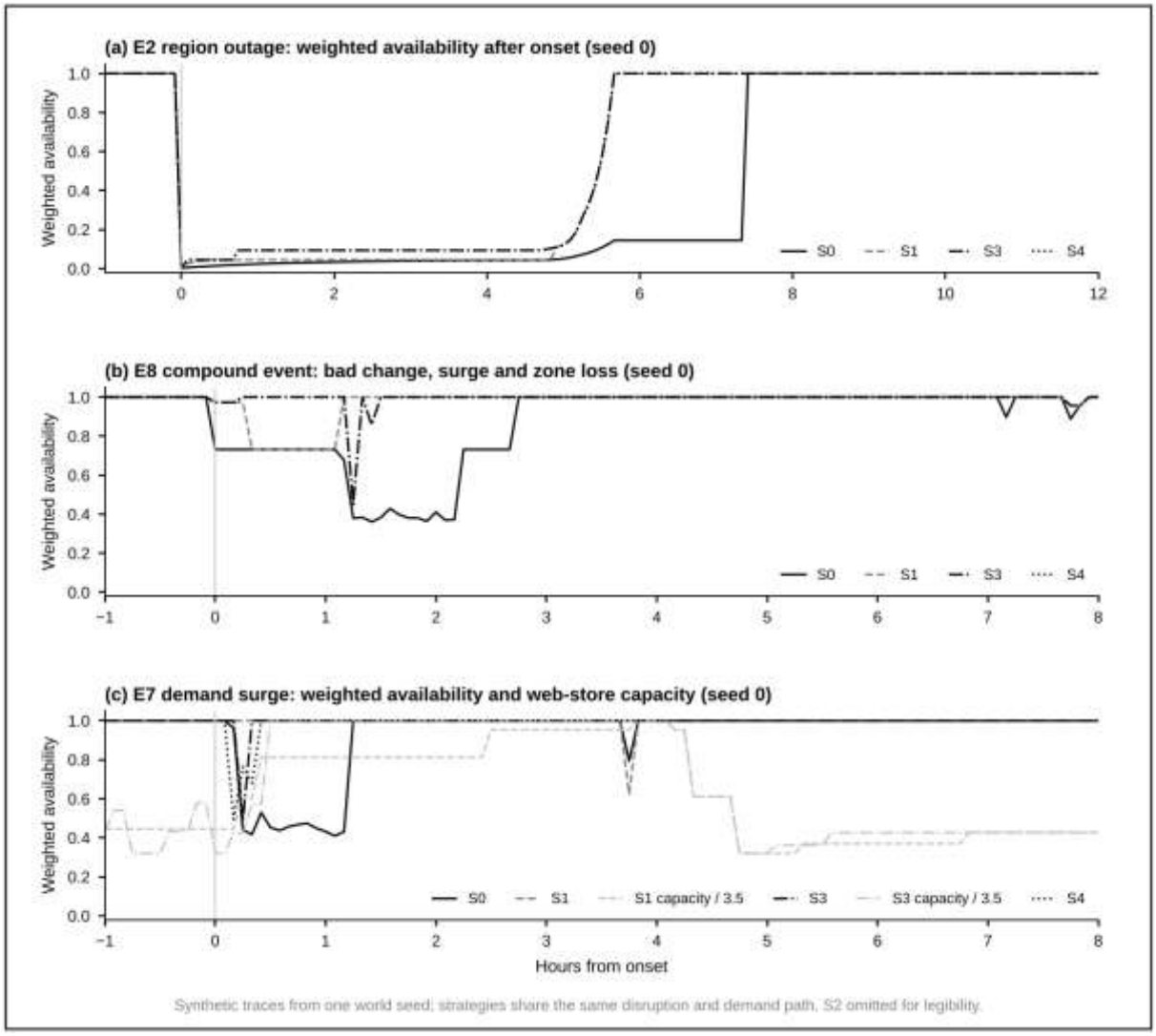
Figure 5: Weighted downtime by scenario and strategy, cost decomposition summed over the eight disruption scenarios, and summed downtime relative to S0 |



Incident timelines

Figure 6 shows single-seed timelines of weighted availability for three scenarios under S0, S1, S3 and S4, with the same disruption and demand path for every strategy. In the region outage (panel a) availability falls to between 0.02 and 0.10 within minutes under every strategy and stays there for the 5.7 hours of the outage; S3 and S4 hold about 0.09 rather than 0.04 because their active-active services are steered to the secondary region in five minutes, and S0 recovers 1.7 hours after the others because its manual failover is still in flight when the region returns. In the compound event (panel b) the bad change at onset costs S0 and S1 27% of weighted availability; S1 rolls it back after an hour, S0 only after 2.7 hours, and S3 and S4 stop it at the canary; the surge an hour later takes S0's fixed capacity down to about 40% for an hour and causes only a short dip under the autoscaling strategies; and in this seed the lost zone held no legacy service. In the demand surge (panel c) S0's fixed capacity saturates at about 45% for 1.2 hours until manual scaling, reactive scaling under S1 recovers within 20 minutes of the ramp, S3 and S4 (whose forecaster had no campaign signal in this seed) dip in the same way, and every strategy dips again at an unannounced second burst at 3.8 hours.

Figure 6: Weighted availability after onset for the region outage, the compound event and the demand surge under S0, S1, S3 and S4 (world seed 0)



Timing metrics

Table 9 reports the median time to detect, the median time to the first correct action and the median time to restore. Detection is fast for every strategy in the scenarios with large availability signatures (medians of 0 to 10 minutes for the region outage, the attack, the bad change and the surge), and the strategies separate on the first correct action: for the region outage the medians are 85, 50, 30, 15 and 10 minutes from S0 to S4; for the denial-of-service attack 80, 53, 33, 15 and 5; for the bad change 113, 65, 40, 10 and 10. These differences are the sum of the decision latency and the wrong-hypothesis penalties of Table 5, and they translate into downtime in proportion to the depth of the degradation. Time to restore is architecture-bound where the strategies cannot act on the cause (the region outage's 368 minutes under S1 to S4 is the outage duration plus the steering lag; the attack's 153 minutes is the attack duration, because scrubbing leaves a residual that keeps the edge above its threshold) and follows the first correct action where they can: the bad change is restored in a median of 135 minutes under S0, 85 under S1, 60 under S2 and 15 to 18 under S3 and S4; the compound event in 250, 90, 85, 18 and 18; the zone outage in 120 minutes under S0 to S2 (the zone's return) and 53 to 60 under S3 and S4 (the migration). For the intrusion the time to detect runs from initial access to the first security detection: a median of 998 minutes under S0 (the encryption itself), 450 under S1 and 340 under S2 to S4; time to restore is 3,610 minutes under S0 (serial restores), 55 under S1 and S2 (the isolation window) and 25 under S3 and S4.

Table 9: Median time to detect, time to the first correct action and time to restore (minutes) by scenario and strategy

Scenario (median MTTD / first correct action / MTTR, min)	S0	S1	S2	S3	S4
E1 Zone outage	40 / 95 / 120	5 / 48 / 120	2 / 35 / 120	0 / 15 / 60	0 / 8 / 52
E2 Region outage	10 / 85 / 458	5 / 50 / 368	0 / 30 / 368	0 / 15 / 368	0 / 10 / 368
E3 DDoS attack	10 / 80 / 152	5 / 52 / 152	0 / 32 / 152	5 / 15 / 152	5 / 5 / 152
E4 Ransomware intrusion	998 / 1058 / 3610	450 / 490 / 55	340 / 372 / 55	340 / 355 / 25	340 / 355 / 25
E5 Bad change	10 / 112 / 135	15 / 65 / 85	15 / 40 / 60	10 / 10 / 18	10 / 10 / 15
E6 Third-party failure	10 / 88 / 38	5 / 55 / 22	2 / 28 / 22	5 / 15 / 22	2 / 15 / 25
E7 Demand surge	20 / 100 / 100	10 / 25 / 18	10 / 35 / 15	10 / 20 / 15	10 / 15 / 15
E8 Compound event	10 / 100 / 250	10 / 55 / 90	10 / 45 / 85	10 / 90 / 18	10 / 80 / 18

The ransomware intrusion in detail

Figure 7 and Table 10 examine the intrusion. In the single-seed timeline of panel (a) the attacker compromises a growing share of services during the dwell; S1 detects it at about eight hours and S3 at about seven, when eight or nine services (a fifth of the estate) are compromised, isolate them for an hour (S1) or 30 minutes (S3) and lose most of weighted availability for that window because the isolated set includes the identity provider; S0 sees nothing until encryption at 15 hours, when 22 services (48% of the estate) have been reached, and is still restoring them serially at the end of the fortnight. Panel (b) shows the incident score under S3 during the dwell: it stays near zero until the security signals of the compromised services lift the 30-minute means, crosses the declaration threshold at about seven hours and is confirmed by the evidence rule. Panel (c) shows detection times and containment for the 20 seeds: S0 detects at encryption (6 to 36 hours); S1 detects at a median of 7.5 hours and contains 14 of 20; S2 to S4 detect at a median of 5.7 hours and contain 17 of 20. The three seeds that S2 to S4 fail to contain are those with the shortest dwells (six to eleven hours), in which the attacker encrypts before the signals of three or four compromised services separate from noise, so that the ceiling here is the attacker's speed relative to the signal, not the detector; S1 misses those three and three more with dwells of nine to eighteen hours, in which its correlation rule fires too late.

Table 10 gives the aggregate: the share contained before encryption rises from 0% (S0) to 70% (S1) and 85% (S2 to S4); the number of compromised services at containment falls from 27 (at encryption, S0) to 14, 8 and 7; the number encrypted from 24 to 2.8, 0.4 and 0.3; lost writes from 559,000 to 84,000, 1,400 and 280; and the mean cost from $58.4 million to $4.6, $0.65, $0.34 and $0.36 million. The S1 figures hide a heavy tail: its median downtime is 50 minutes but its 90th percentile is 211 and its standard deviation 300, because the 30% of intrusions it misses are expensive; S2 to S4 have 90th percentiles of 24 to 55 minutes. Containment itself costs availability: isolating the compromised services, which usually include the identity provider, takes weighted availability down by 68 to 80% for the isolation window, which is why S3's 30-minute window halves S1's and S2's downtime although their containment shares are similar.

Figure 7: The ransomware intrusion: single-seed timeline of availability and compromised share under S0, S1 and S3, the incident score during the dwell under S3, and detection time and containment over 20 seeds

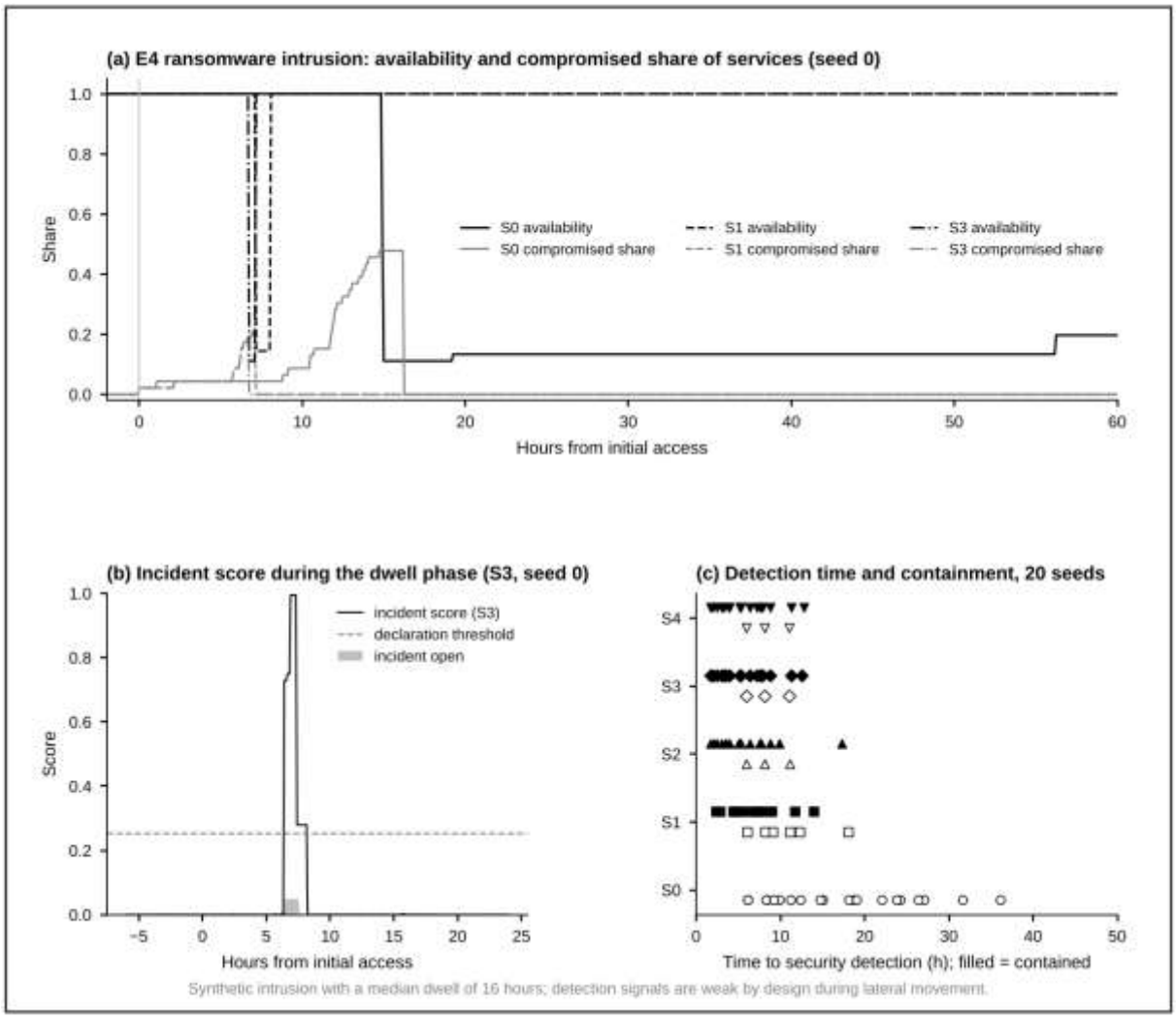


Table 10: Ransomware intrusion outcomes by strategy: security detection, containment before encryption, compromised and encrypted services, lost writes, restores, weighted downtime, time to restore and cost (20 seeds)

Outcome (20 seeds)	S0	S1	S2	S3	S4
Security detection before the end of the episode (share of seeds)	100%	100%	100%	100%	100%
Time to security detection, median (min)	998	450	340	340	340
Contained before encryption (share of seeds)	0%	70%	85%	85%	85%
Compromised services at containment or encryption (mean)	27.1	14.4	8.2	7.2	7.5
Services encrypted (mean)	24.4	2.8	0.4	0.3	0.3
Services restored from backup (mean)	24.4	2.8	0.4	0.3	0.3
Lost writes (mean)	559,316	84,461	1,357	278	278
Blast radius (largest share of weighted availability lost)	89%	80%	72%	68%	72%
Weighted downtime, mean (SD), min	3736 (2159)	144 (300)	47 (26)	21 (13)	23 (12)

Outcome (20 seeds)	S0	S1	S2	S3	S4
Weighted downtime, median and 90th percentile, min	4318; 6131	50; 211	47; 55	21; 24	21; 25
Time to restore, median (min)	3610	55	55	25	25
Cost, mean (SD), $ million	58.41 (32.41)	4.58 (11.45)	0.65 (0.45)	0.34 (0.18)	0.36 (0.18)

Distributions, costs and side effects

Figure 8 shows the distributions over seeds for four scenario-metric pairs. The region outage's downtime distribution is wide (interquartile range 210 to 450 minutes) and identical in shape across S1 to S4 because it is the outage-duration distribution, with S0 shifted up by its manual failover; the intrusion's downtime spans three orders of magnitude under S0 and S1 and collapses to a narrow band under S2 to S4; the attack's downtime halves from S1 to S3, with S2 showing a tail from wrong first actions; and the bad change's time to restore falls from a median of 135 minutes and a 90th percentile of 229 (S0) to 15 to 18 minutes (S3 and S4).

Figure 8: Distributions over 20 seeds: weighted downtime for the region outage, the intrusion and the denial-of-service attack, and time to restore for the bad change, by strategy

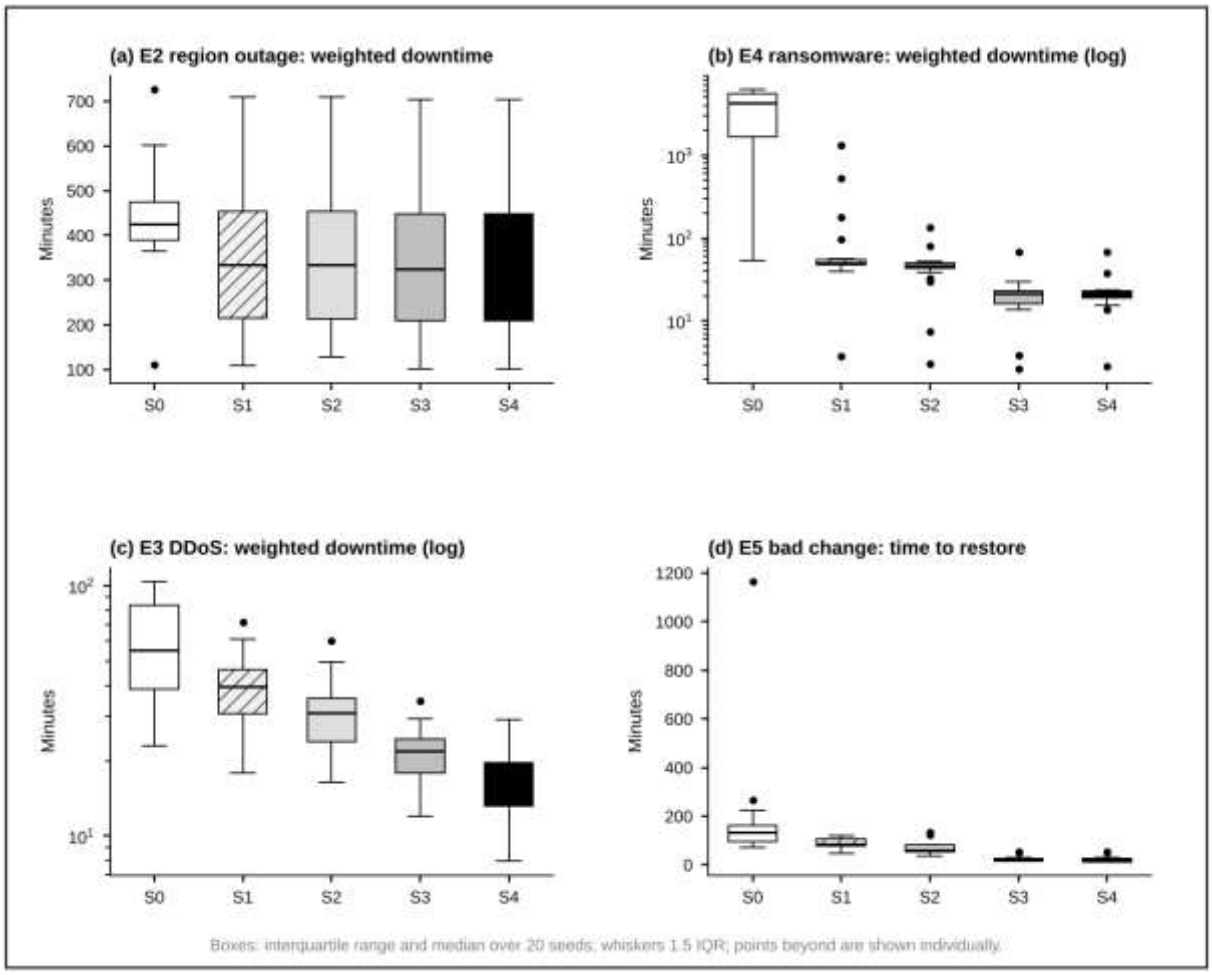


Table 11 decomposes the cost summed over the eight disruption scenarios and reports the side effects. Revenue loss is 60% of S0's cost, 59% of S1's and about 80% of the others'; lost writes and restores are the second component of S0 ($27.9 million, from day-old backups and four-hour snapshot replication) and S1 ($3.1 million), and they all but disappear from S2 upward; extra capacity ($0.84 million for every autoscaling strategy) is the largest remaining item, and labour is small at the assumed rates. The side effects run the other way. Nuisance incidents (declarations with the estate healthy) average 0.0 per fortnight under S0, about 1 under S1 and about 3 under S2 to S4, which is within the budget of two per

week at which the detectors were calibrated but three times S1's rate; the blip incidents on background third-party and zone blips are similar across S1 to S4. Wrong actions summed over the eight scenarios fall from 4.6 per set of episodes under S0 to 3.6 under S1, rise to 4.3 under S2 (the automation-bias cost of acting on a wrong classification without decision support), and fall to 1.5 under S3 and 2.6 under S4; S4 takes 3.7 automated actions per set of episodes, of which 0.55 are wrong, concentrated in the surge, the intrusion, the third-party and the compound scenarios, where a confident misclassification triggers a needless scrub or scale-up. The automated wrong actions cost a few minutes each under the guardrails, which is the small S4 penalty seen in Table 8 for the surge and the bad change.

Table 11: Cost decomposition summed over the eight disruption scenarios ($ million) and side effects per strategy: nuisance and blip incidents per fortnight, wrong actions, automated actions and automated wrong actions

Cost component or side effect	S0	S1	S2	S3	S4
Revenue loss ($ million)	44.55	6.02	4.86	4.08	4.06
Incident labour ($ million)	0.55	0.20	0.17	0.15	0.15
Scrubbing ($ million)	0.00	0.00	0.00	0.00	0.00
Extra capacity ($ million)	1.72	0.84	0.84	0.84	0.84
Restores ($ million)	0.61	0.07	0.01	0.01	0.01
Lost writes ($ million)	27.31	3.01	0.10	0.06	0.06
Total ($ million)	74.74	10.14	5.99	5.14	5.12
Normal fortnight cost ($ million)	0.03	0.10	0.11	0.11	0.11
Nuisance incidents per fortnight (mean over the nine scenarios)	0.0	1.0	3.1	3.1	3.0
Blip incidents per fortnight (mean over the nine scenarios)	0.2	0.5	0.7	0.7	0.7
Wrong actions (sum over E1-E8, per set of episodes)	4.6	3.6	4.3	1.5	2.6
Automated actions (sum over E1-E8)	0.0	0.0	0.0	0.0	3.7
Automated wrong actions (sum over E1-E8)	0.00	0.00	0.00	0.00	0.55

Sensitivity to detection and decision parameters

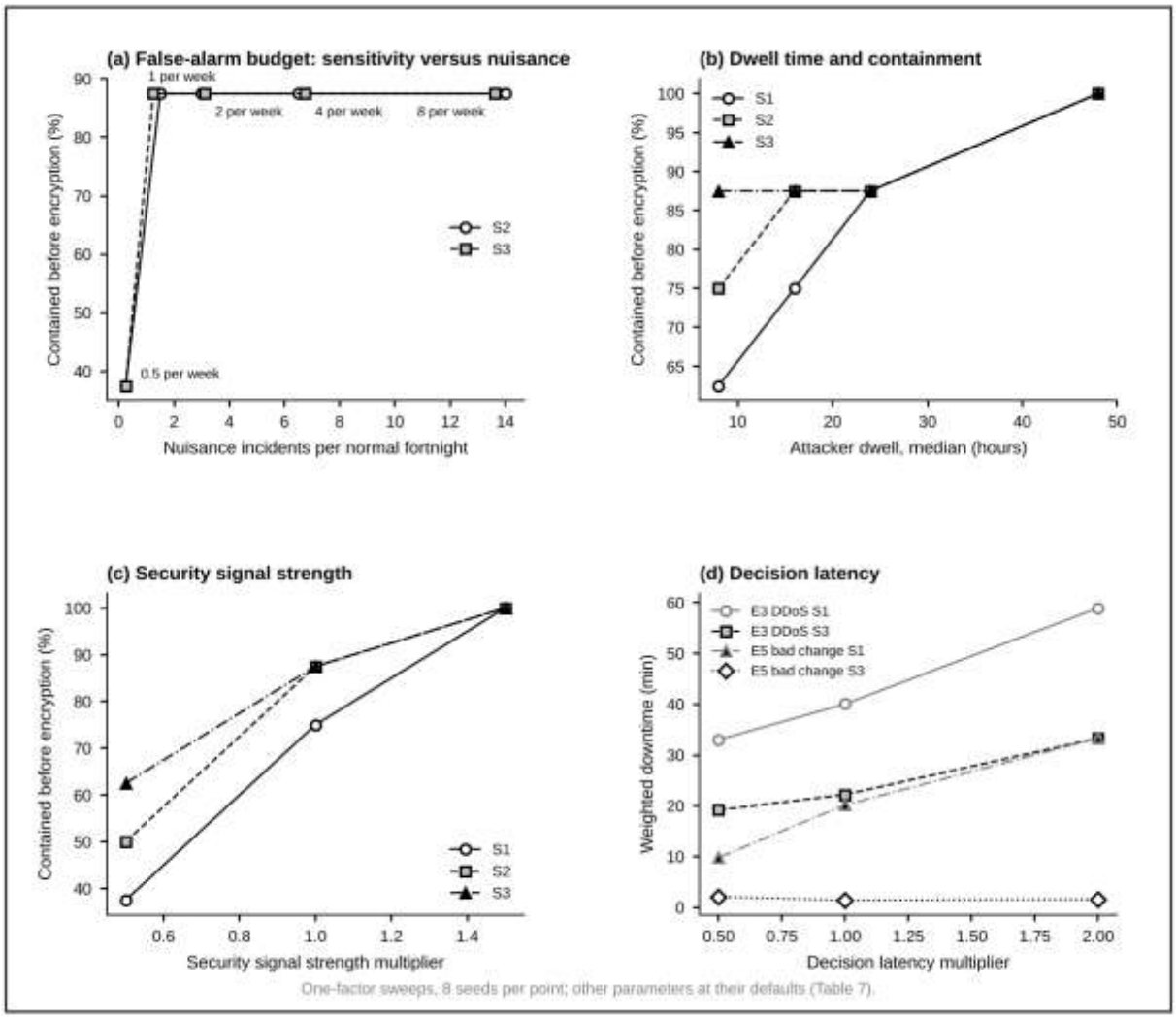
Table 12 and Figure 9 report the one-factor sweeps of group A (eight seeds per point). The false-alarm budget has a threshold character: at 0.5 declarations per week the detectors raise 0.2 nuisance incidents per fortnight but contain only 38% of intrusions, with a median security detection at 818 minutes; at one per week they raise 1.2 to 1.5 and contain 88% at 302 minutes; from two to eight per week containment and detection time do not improve (88%, 278 to 285 minutes) while nuisance incidents rise from 3 to 14 per fortnight. The default budget of two per week is therefore at the knee, and one per week would buy a third of the nuisance at no cost in containment on this testbed. The attacker's dwell is the next determinant: with a median dwell of eight hours S1 contains 62%, S2 75% and S3 88%; at 16 hours 75%, 88% and 88%; at 24 hours 88% for all three; and at 48 hours every strategy contains every intrusion, because S1's correlation rules eventually see an attacker who has reached a dozen services. The AI detector's advantage is therefore an advantage against fast attackers. Lateral spread cuts both ways: a slower spread gives the attacker fewer services at encryption but weaker signals, and a faster spread is caught earlier, so that S3's containment rises to 100% and S1's downtime falls from 226 to 50 minutes at the fast spread. The security signal strength has the largest effect: at half the default strength S1 contains 38%, S2 50% and S3 62%, with S3's median detection moving from 4.6 to 9.1 hours; at 1.5 times the default every strategy contains every intrusion. Decision latency scales the fast-moving scenarios roughly linearly: halving and doubling it moves the denial-of-service downtime from 33 to 59 minutes under S1 and from 19 to 33 under S3, and the bad-change downtime from 10 to 33 under S1 while S3 stays at 1 to 2 because its canary analysis acts before any decision; the region outage does not

respond because its loss is architectural. The automation confidence threshold has no measurable effect between 0.6 and 0.95: the classifier's top-class confidence is close to one for the clear-cut classes and low otherwise, so the guardrail rarely binds in that range, and the automated wrong actions (0.12 to 0.25 per episode in the surge scenario) come from confident misclassifications that a threshold below one does not stop. Observability quality and telemetry noise act mainly through the nuisance rate: at observability 0.4 or noise 2.0 the nuisance incidents roughly double (6 to 7 per fortnight) and security detection is delayed by one to two hours, with containment falling to 75% under S2 at double noise.

Table 12: One-factor sweeps of the detection and decision parameters (group A; eight seeds per point): outcomes by parameter value and strategy

Parameter	Scenario: metric (strategies)	Value 1	Value 2	Value 3	Value 4	Value 5
False-alarm budget (per week)	Normal: nuisance incidents per fortnight (S2 / S3)	0.5: 0.2 / 0.2	1: 1.5 / 1.2	2: 3.0 / 3.1	4: 6.5 / 6.8	8: 14.0 / 13.6
False-alarm budget (per week)	E4: contained before encryption (S2 / S3)	0.5: 38% / 38%	1: 88% / 88%	2: 88% / 88%	4: 88% / 88%	8: 88% / 88%
False-alarm budget (per week)	E4: median time to security detection (min) (S3)	0.5: 818	1: 302	2: 278	4: 280	8: 285
Attacker dwell, median (min)	E4: contained before encryption (S1 / S2 / S3)	480: 62% / 75% / 88%	960: 75% / 88% / 88%	1440: 88% / 88% / 88%	2880: 100% / 100% / 100%	
Attacker dwell, median (min)	E4: weighted downtime (min) (S1 / S2 / S3)	480: 136 / 78 / 26	960: 226 / 58 / 26	1440: 67 / 57 / 25	2880: 51 / 47 / 20	
Lateral spread per edge per step	E4: contained before encryption (S1 / S3)	0.004: 88% / 88%	0.008: 75% / 88%	0.016: 88% / 100%		
Security signal strength	E4: contained before encryption (S1 / S2 / S3)	0.5: 38% / 50% / 62%	1: 75% / 88% / 88%	1.5: 100% / 100% / 100%		
Security signal strength	E4: weighted downtime (min) (S1 / S3)	0.5: 446 / 54	1: 226 / 26	1.5: 51 / 21		
Decision latency multiplier	E3: weighted downtime (min) (S1 / S3)	0.5: 33 / 19	1: 40 / 22	2: 59 / 33		
Decision latency multiplier	E5: weighted downtime (min) (S1 / S3)	0.5: 10 / 2	1: 20 / 1	2: 33 / 2		
Decision latency multiplier	E2: median first correct action (min) (S1 / S3)	0.5: 22 / 10	1: 40 / 15	2: 58 / 30		
Automation confidence threshold	E3: automated actions; automated wrong (S4)	0.6: 1.1; 0.12	0.75: 1.1; 0.12	0.85: 1.0; 0.00	0.95: 1.0; 0.00	
Automation confidence threshold	E7: automated actions; automated wrong (S4)	0.6: 1.1; 0.25	0.75: 1.0; 0.12	0.85: 1.0; 0.12	0.95: 1.0; 0.12	
Automation confidence threshold	E7: weighted downtime (min) (S4)	0.6: 5	0.75: 5	0.85: 5	0.95: 5	
Observability quality	E4: median time to security detection (min) (S2 / S3)	0.4: 400 / 338	0.7: 302 / 302	1: 272 / 278		
Observability quality	E8: nuisance incidents per fortnight (S2 / S3)	0.4: 6.0 / 5.6	0.7: 2.9 / 2.6	1: 3.1 / 2.8		
Telemetry noise multiplier	E4: contained before encryption (S2 / S3)	0.5: 88% / 88%	1: 88% / 88%	2: 75% / 88%		
Telemetry noise multiplier	E4: nuisance incidents per fortnight (S2 / S3)	0.5: 3.4 / 3.9	1: 3.9 / 3.8	2: 6.6 / 6.8		

Figure 9: Detection and decision sweeps: the false-alarm budget against nuisance incidents and containment, attacker dwell, security signal strength and decision latency



Sensitivity to architecture and recovery parameters

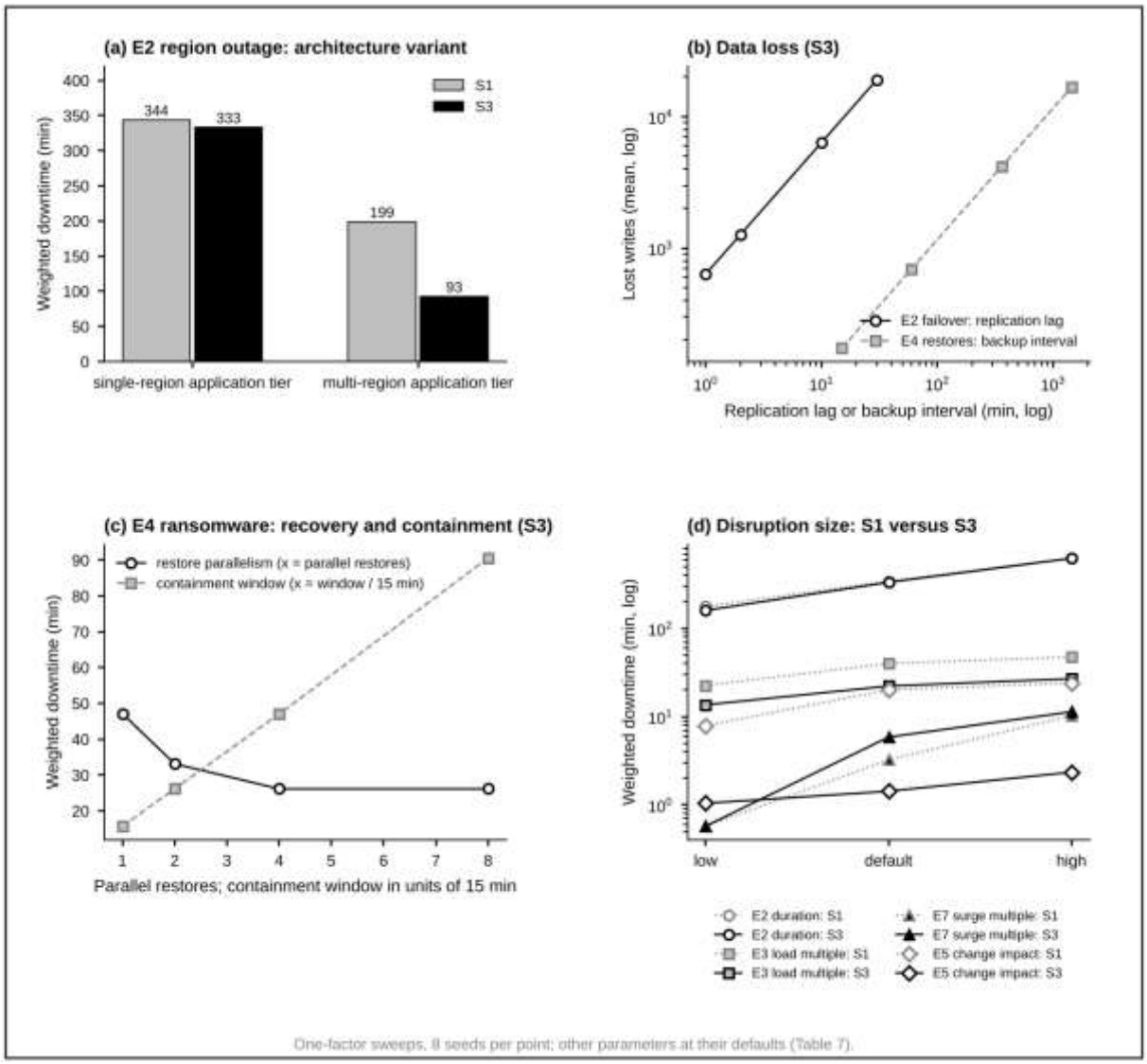
Table 13 and Figure 10 report group B. The recovery-time objective of the orchestrated failover makes almost no difference to the region outage (333 to 339 minutes from a 15-minute to a 120-minute objective), because the application tier, the cache and the queue are deployed only in the primary region and the failover of the data stores restores nothing the applications can use; the multi-region application tier is the lever, cutting the outage from 344 to 199 minutes under S1 and from 333 to 93 minutes under S3, where the remainder is the steering lag and the failover of the stores. This is the clearest instance of the architecture ceiling: below it the operating strategy changes the outcome by a few per cent, above it by a factor of two. Data loss follows the recovery-point parameters linearly: lost writes at failover scale with the replication lag (635 at one minute to 19,050 at 30) and lost writes at restore with the backup interval (174 at 15 minutes to 16,700 at a day), so hourly immutable backups are worth about 16,000 writes per intrusion relative to daily backups. Restore parallelism and the containment window set the intrusion's downtime once it is contained: one restore at a time gives 47 minutes, two 33, four and eight 26; the containment window moves the downtime from 16 minutes at 15 minutes to 91 at two hours, because the isolated set usually includes the identity provider and most of the estate waits for it. The disruption-size sweeps show the strategies scaling in parallel: the region outage's downtime is linear in its duration under both S1 and S3 (178 and 160 minutes at two hours, 630 and 623 at ten); the denial-of-service downtime rises with the attack multiple from 23 to 47 minutes under S1, 14 to 27 under S3 and 10 to 19 under S4, and scrubbing efficiency matters below 95% (35 minutes at 80%, 22 at

95% and 99%); the demand surge costs 1 to 10 minutes under S1 and 1 to 11 under S3 across multiples of two to five, the one place where S3 does not improve on S1, because the forecaster's pre-scaling does not beat reactive scaling on a 15-minute ramp and the campaign lead signal changes the downtime by one or two minutes, within the seed noise. The bad change scales with its impact under S1 (8 to 24 minutes) and not under S3 (1 to 2), and the third-party failure's residual availability matters little (7 to 9 minutes under S1, 2 to 3 under S3) because the fallback, not the residual, decides the outcome.

Table 13: One-factor sweeps of the architecture and recovery parameters (group B; eight seeds per point): outcomes by parameter value and strategy

Parameter	Scenario: metric (strategies)	Value 1	Value 2	Value 3	Value 4
S3 failover RTO (min)	E2: weighted downtime (min) (S3)	15: 333	30: 333	60: 335	120: 339
Multi-region application tier (0 no, 1 yes)	E2: weighted downtime (min) (S1 / S3)	0: 344 / 333	1: 199 / 93		
Replication lag (min)	E2: lost writes (S3)	1: 635	2: 1,270	10: 6,350	30: 19,050
S3 backup interval (min)	E4: lost writes (S3)	15: 174	60: 696	360: 4,175	1440: 16,700
Parallel restores (S3)	E4: weighted downtime (min) (S3)	1: 47	2: 33	4: 26	8: 26
S3 containment window (min)	E4: weighted downtime (min) (S3)	15: 16	30: 26	60: 47	120: 91
Region outage duration (min)	E2: weighted downtime (min) (S1 / S3)	120: 178 / 160	300: 344 / 333	600: 630 / 623	
DDoS load multiple	E3: weighted downtime (min) (S1 / S3 / S4)	3: 23 / 14 / 10	8: 40 / 22 / 15	15: 47 / 27 / 19	
Scrubbing efficiency	E3: weighted downtime (min) (S3)	0.8: 35	0.95: 22	0.99: 22	
Surge multiple	E7: weighted downtime (min) (S1 / S3)	2: 1 / 1	3.5: 3 / 6	5: 10 / 11	
Campaign lead-signal probability	E7: weighted downtime (min) (S3)	0: 4	0.7: 6	1: 5	
Bad change impact	E5: weighted downtime (min) (S1 / S3)	0.2: 8 / 1	0.6: 20 / 1	1: 24 / 2	
Third-party residual availability	E6: weighted downtime (min) (S1 / S3)	0: 9 / 3	0.1: 9 / 3	0.4: 7 / 2	

Figure 10: Architecture and recovery sweeps: the multi-region application tier, data loss against replication lag and backup interval, restore parallelism and containment window, and disruption size under S1 and S3



Monte Carlo analysis over joint uncertainty

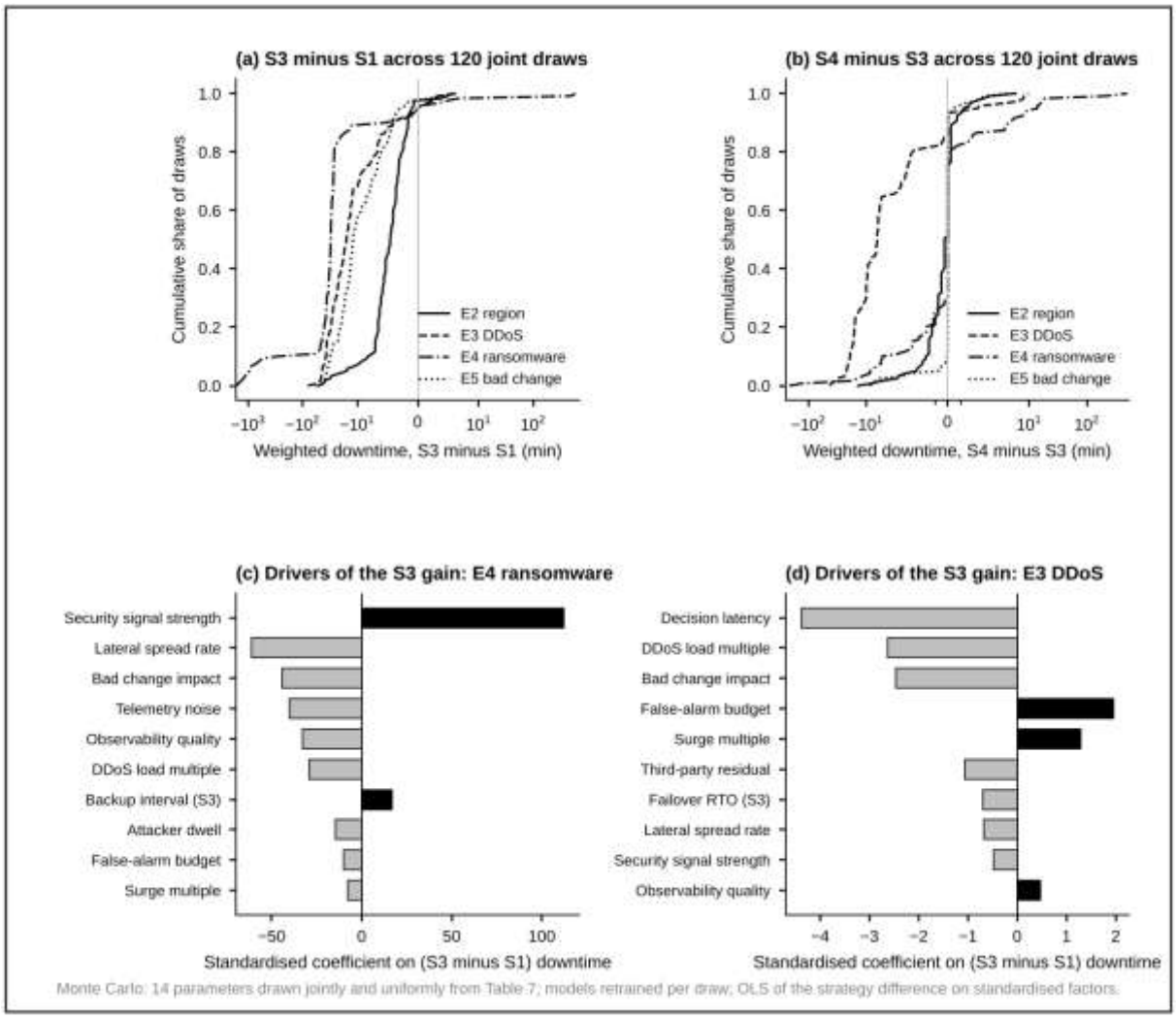
Table 14 and Figure 11 report the Monte Carlo analysis: 120 joint draws of the 14 parameters of Table 7, each with the models retrained and the threshold recalibrated for the drawn false-alarm budget, run under S1, S3 and S4 on the region outage, the denial-of-service attack, the intrusion and the bad change. The strategy ranking is robust to joint uncertainty for the three classes the operating strategy can act on. For the intrusion, S3 beats S1 in 95% of draws, with a median gain of 31 weighted minutes and means of 153, 30 and 33 minutes for S1, S3 and S4; across the draws S1 contains 78% of intrusions before encryption, S3 87% and S4 88%, and S3's containment share is 80% in the shortest-dwell tercile, 95% in the middle and 85% in the longest, and varies by no more than ten percentage points across the signal-strength and false-alarm-budget terciles, because the other factors vary jointly and the sweeps of Table 12 isolate each one more cleanly. For the denial-of-service attack S3 beats S1 in 94% of draws (median gain 16 minutes; means 46, 27 and 19) and S4 beats S3 in 85% (median gain 6 minutes), so that S4 is the best strategy in 84% of draws: this is the one class in which guarded automation's zero latency pays consistently, because the scrub is a low-risk action that the classifier identifies with high confidence. For the bad change S3 beats S1 in 98% of draws (means 16, 2 and 2), and S4 adds nothing because the canary analysis acts before any decision. For the region outage the three strategies are within 6 minutes of each other on average (486, 480 and 480); S3 beats S1 in 98% of draws but by a median of only 5 minutes, which is the architecture ceiling again. The regressions of the S3 minus S1 difference on the

standardised factors identify the drivers, with the sign convention that a negative coefficient enlarges the S3 gain. For the intrusion the gain is largest when the signal is weak and the spread fast and shrinks as the signal strengthens, because S1's correlation rule catches up: security signal strength carries +112 minutes per standard deviation ($p = 0.002$) and lateral spread rate -61 ($p = 0.08$), no other factor is significant, and the fit is loose (R-squared 0.22) because the difference is dominated by the few draws in which S1 misses the intrusion and S3 does not. For the attack and the bad change the gain grows with the disruption's size and with the decision latency that S3 shortens: decision latency -4.4 minutes per standard deviation ($p = 0.009$) and the attack multiple -2.6 ($p = 0.07$) for the attack, and change impact -4 and decision latency -4 (both $p < 0.001$) for the bad change, with R-squared of 0.18 and 0.31. Mean cost across the four classes falls from $2.47 million per episode under S1 to $1.54 million under S3 and $1.53 million under S4.

Table 14: Monte Carlo analysis over 120 joint draws of 14 parameters: mean weighted downtime by strategy, the distribution of the S3 minus S1 and S4 minus S3 differences, the share of draws in which each strategy is best, and the standardised drivers of the S3 gain

Scenario	Mean weighted downtime S1 / S3 / S4 (min)	S3 minus S1: median (share of draws S3 better)	S4 minus S3: median (share S4 better)	Best strategy share S1 / S3 / S4	Largest standardised drivers of the S3 gain (coefficient, p)	R-squared
E2 Region outage	486 / 480 / 480	-5 (98%)	0 (51%)	1% / 48% / 51%	Region outage duration (+2, p = 0.088); Telemetry noise (-2, p = 0.079); Decision latency (-1, p = 0.069); Third-party residual (+1, p = 0.443)	0.19
E3 DDoS attack	46 / 27 / 19	-16 (94%)	-6 (85%)	2% / 13% / 84%	Decision latency (-4, p = 0.009); DDoS load multiple (-3, p = 0.072); Bad change impact (-2, p = 0.085); False-alarm budget (+2, p = 0.121)	0.18
E4 Ransomware intrusion	153 / 30 / 33	-31 (95%)	0 (31%)	3% / 68% / 28%	Security signal strength (+112, p = 0.002); Lateral spread rate (-61, p = 0.075); Bad change impact (-44, p = 0.194); Telemetry noise (-40, p = 0.271)	0.22
E5 Bad change	16 / 2 / 2	-12 (98%)	0 (11%)	2% / 88% / 11%	Bad change impact (-4, p < 0.001); Decision latency (-4, p < 0.001); Failover RTO (S3) (-2, p = 0.147); Lateral spread rate (-2, p = 0.087)	0.31
All four (mean cost $M)	2.47 / 1.54 / 1.53				Containment before encryption (E4): S1 78%, S3 87%, S4 88%	

Figure 11: Monte Carlo results: cumulative distributions of the S3 minus S1 and S4 minus S3 differences in weighted downtime across the 120 draws, and the standardised drivers of the S3 gain for the intrusion and the region outage



Interpretation against the research questions

The results answer the research questions as follows, with every statement conditional on the testbed's assumptions. RQ1: relative to manual reactive operations, rule-based monitoring removes 86% of the summed weighted downtime of the eight disruption classes, almost all of it by containing most ransomware intrusions and by autoscaling; AI detection removes a further 3 percentage points, mostly from the intrusion (containment 85% against 70%, and a narrower tail); decision support removes a further 2 points, concentrated in the bad change (canary analysis), the denial-of-service attack (faster and better first actions), the compound event (re-triage and zone migration) and the third-party failure (secondary provider); and guarded automation adds no measurable gain in the sum. Excluding the intrusion, the shares retained are 64%, 61%, 54% and 53% from S1 to S4. The limit set by the architecture is the region outage, where the strategies differ by 3% and a multi-region application tier by a factor of three to four, and the recovery-point parameters (replication lag, backup interval) set the data loss independently of the strategy. RQ2: time to detect falls from tens of minutes to a few minutes for the availability-driven classes and from the moment of encryption (median 17 hours after initial access) to 5.7 hours for the intrusion; the first correct action falls from 80 to 113 minutes under S0 to 5 to 15 under S4 for the single-cause classes; time to restore follows the first action where the strategy can act and the disruption's own duration where it cannot. The side effects move the other way: nuisance incidents rise from 1 per fortnight under rules to 3 under the calibrated detectors, wrong actions fall from 4.6 to 1.5 per set of episodes only once decision support is present (AI detection without it raises them to 4.3 through automation bias), and guarded automation takes 3.7 automated actions per set of episodes of

which 0.55 are wrong. RQ3: the sweeps and the Monte Carlo analysis agree on the drivers. For the intrusion they are the security signal strength, the dwell and the false-alarm budget (with a knee at one declaration per week) and, once contained, the containment window and the restore parallelism; for the fast-moving classes the decision latency and the disruption size; for the region outage the duration and the deployment pattern; and for data loss the replication lag and the backup interval. Across the Monte Carlo draws S3 or S4 is the best of the three strategies in 97% of intrusion draws, 98% of attack draws and 98% of bad-change draws, so the ranking does not depend on the point at which the parameters were set. RQ4 is answered by the mapping of Table 15 and the roadmap of Figure 12 in Section 6: the sequence that the results support is foundation (catalogue, telemetry, multi-zone and multi-region patterns, tested backups), then calibrated detection joined across operations and security, then decision support and orchestrated recovery, then guarded automation of the low-risk playbooks, with governance setting the false-alarm budget, the automation tier and the recovery objectives explicitly.

Discussion and implications

Theoretical implications

The study makes four points for the theory of resilient operations. First, detection, decision and architecture are separable contributions to resilience that bind in different disruption classes. The quantitative resilience literature measures performance loss over time but rarely decomposes the recovery curve into the time to notice, the time to decide and the time the system takes to execute (Bruneau et al., 2003; Henry & Ramirez-Marquez, 2012; Hosseini et al., 2016); the lifecycle of Figure 2 and the metrics of Table 6 make that decomposition explicit, and the results show that for a bad change or a denial-of-service attack the decision component dominates, for a region outage the execution component dominates and is set by the deployment pattern, and for an intrusion the detection component dominates and is set by the strength of the lateral-movement signal relative to the dwell: the cloud-operations counterpart of the resilience-engineering claim that the capacity to respond and the capacity to monitor are distinct (Hollnagel et al., 2006; Woods, 2015).

Second, the base-rate discipline of intrusion detection and the alert-fatigue findings of security-operations research apply to AIOps in general: a detector tuned to catch weak lateral-movement signals declares incidents on normal fortnights at a rate the false-alarm budget sets, and the sweep on that budget shows the trade-off directly (Axelsson, 2000; Alahmadi et al., 2022); the framework makes the budget a governed parameter and lets decision support absorb the cost of the nuisance incidents by resolving them cheaply. Third, automation bias is a measurable outcome: the strategies in which the operator follows a wrong classification with high probability take wrong actions more often, and the guarded-automation strategy's automated wrong actions measure directly the harm the trust-in-automation literature warns about (Parasuraman & Manzey, 2010; Goddard et al., 2012; Lee & See, 2004). Fourth, the architecture ceiling is a theoretical object: the availability loss that remains when detection and decision latency are zero, which depends only on the deployment patterns, the replication and backup intervals and the failover and restore mechanics. The region-outage results, the recovery-objective sweeps and the multi-region variant quantify it, and below it the marginal value of better decisions is small, which is where recovery-oriented computing and the design-for-failure rules placed the emphasis (Patterson et al., 2002; Hamilton, 2007).

Managerial implications

For chief information and information-security officers and their site-reliability and security-operations leads the results, conditional on the testbed's assumptions, give five implications. First, observability and the service catalogue come first, because every later layer consumes them: the strategies that detect in minutes do so because the telemetry and the dependency weights exist. Second, calibrated detection with a governed false-alarm budget, joined across reliability and security signals, is what turns a ransomware intrusion from a multi-day restore into a contained incident; the budget should be set with the on-call load in view and reviewed as the estate changes. Third, decision support with explanations and expected-loss ranking removes most of the wrong first actions and the decision latency, which for bad changes and denial-of-service attacks is where most of the achievable gain lies; it should run in shadow mode first so that its recommendations can be scored against what operators did. Fourth, orchestrated recovery (tested failover, immutable hourly backups, parallel restores, zone migration and

secondary providers) is what the decision support acts on; the sweeps on the backup interval and the restore parallelism give the data-loss and recovery-time consequences of under-investing here. Fifth, guarded automation should cover the low-risk playbooks only, with a confidence threshold, a blast-radius cap and automatic reversion, and with human approval retained for failover, isolation and restores. Region outages are the exception: for them the lever is architecture, and the multi-region variant shows the size of the gain that a multi-region application tier buys.

For vendors and platform teams the implication is that incident-management tooling should expose its detection thresholds, class probabilities and the evidence behind a security call, because the framework's governance depends on being able to set and audit them (Rudin, 2019; Lipton, 2018). For boards and risk committees the timing metrics and the cost decomposition are the quantities the disclosure and reporting rules ask organisations to be able to produce.

Standards and regulatory mapping

Table 15: Mapping of the framework's components to standards and U.S. regulatory instruments, with the evidence from the testbed

Instrument	What it requires or recommends	Framework components that address it	Evidence from the testbed
NIST Cybersecurity Framework 2.0 (National Institute of Standards and Technology, 2024)	Govern, Identify, Protect, Detect, Respond, Recover outcomes; supply-chain risk; continuous improvement	Governance layer (SLOs, exercises, reviews); telemetry and AI analytics (Detect); decision support and response (Respond); failover, restore (Recover)	Detect and Respond outcomes dominate the gains for most classes; Recover architecture sets the ceiling for region outages and ransomware
NIST SP 800-61 Rev. 3 (Nelson et al., 2025); SP 800-184 (Bartock et al., 2016)	Incident handling as a CSF 2.0 profile: preparation, detection and analysis, containment, eradication, recovery, lessons learned	Lifecycle states (Figure 2); containment and restore playbooks; post-incident review feeding model updates	Containment before encryption depends on security detection during dwell; restore parallelism and backup interval set recovery time and data loss
NIST SP 800-160 Vol. 2 (Ross et al., 2021); SP 800-207 (Rose et al., 2020); CISA zero-trust maturity model version 2.0	Cyber-resiliency techniques (segmentation, redundancy, analytic monitoring, adaptive response); zero-trust access to limit lateral movement	Identity-aware attack-path reasoning; isolation playbooks; segmentation parameters (spread into the identity provider)	Lateral spread rate and identity compromise drive the intrusion outcome more than detection sensitivity
SEC cybersecurity disclosure rule (U.S. Securities and Exchange Commission, 2023)	Disclosure of material incidents within four business days of a materiality determination; description of risk management and governance	Governance lane: reportable-incident gateway; incident records with timing metrics; board-level reporting of exercises	Timing metrics and cost decomposition give the materiality assessment a quantitative basis
CIRCIA (U.S. Congress, 2022)	Reporting of covered cyber incidents and ransom payments by covered entities once the implementing rule is in force	Reportable-incident gateway; evidence preservation in containment; incident timeline	Containment and restore records produced by the lifecycle supply the report content

Instrument	What it requires or recommends	Framework components that address it	Evidence from the testbed
NYDFS Part 500 second amendment (New York State Department of Financial Services, 2023); Safeguards Rule (Federal Trade Commission, 2021)	Risk assessment, monitoring, incident response and business-continuity plans, notification duties, multi-factor authentication, asset inventories	Service catalogue and dependency graph; continuous monitoring; tested failover and restore; identity controls	Service criticality weights and dependency logic are the asset inventory the rules require
FedRAMP baselines (FedRAMP, n.d.); NIST SP 800-53 (Joint Task Force, 2020); SP 800-137 (Dempsey et al., 2011)	Control baselines for cloud services used by federal agencies; continuous monitoring	Telemetry layer; model monitoring; change management with canaries	Canary analysis removes most bad-change impact at S3 and above
ISO 22301 (International Organization for Standardization, 2019); ISO/IEC 27035-1 (ISO/IEC, 2023)	Business-continuity management; incident management process and planning	Recovery objectives (RTO, RPO) as parameters; exercises; lifecycle	Failover RTO and replication lag sweeps quantify the continuity objectives
DORA (European Parliament and Council, 2022)	ICT risk management, incident reporting, resilience testing, third-party risk for EU financial entities	Third-party fallback playbooks; chaos and game-day exercises; reporting gateway	Third-party failures are cheap with a secondary provider and expensive without one
NIST AI RMF (Tabassi, 2023); adversarial ML taxonomy (Vassilev et al., 2024)	Govern, map, measure and manage AI risks; awareness of evasion and poisoning of detectors	Calibration to a false-alarm budget; confidence thresholds; model monitoring; human approval for high-impact actions	The automation threshold sweep shows the harm-benefit trade-off the AI RMF asks organisations to manage

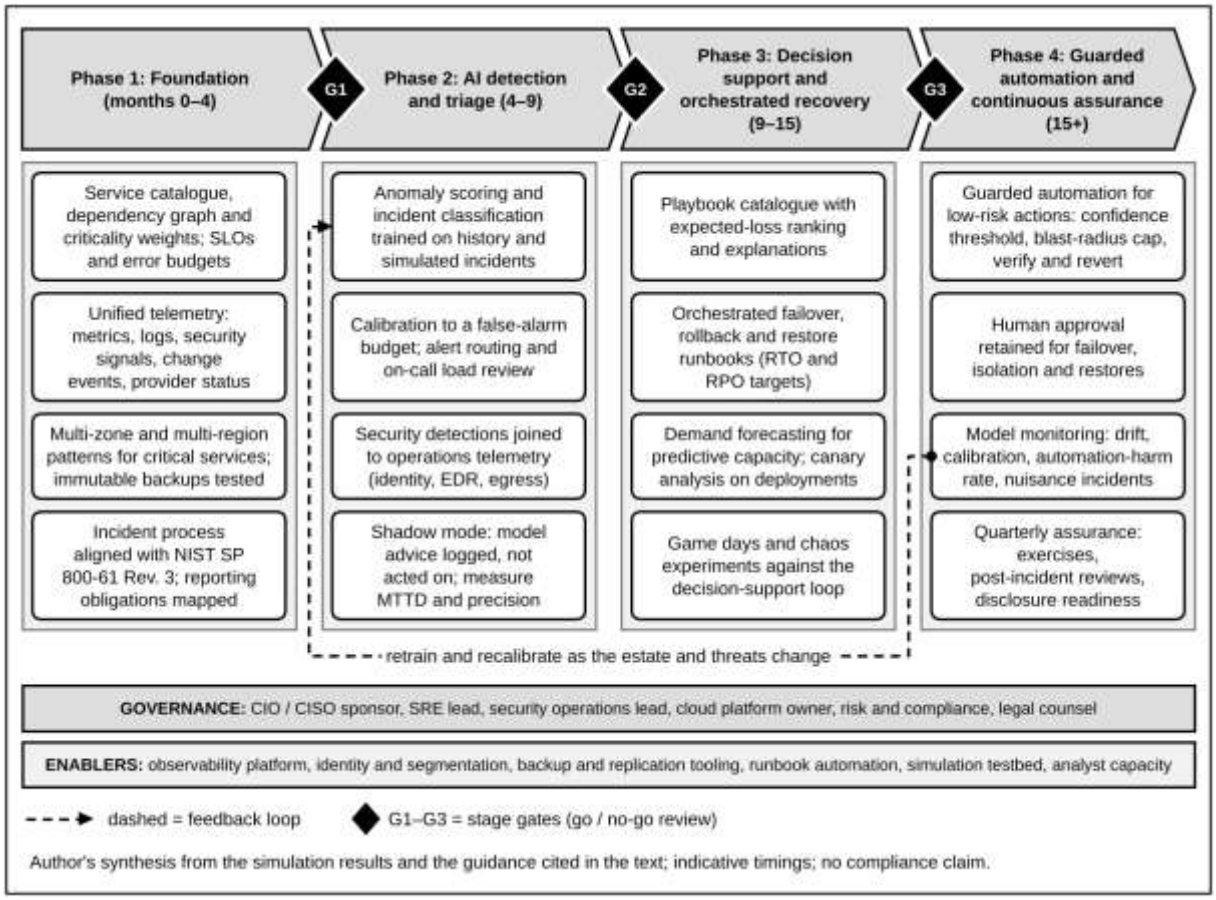
Table 15 maps the framework's components to the instruments that require or recommend them and gives the evidence from the testbed that bears on each. The NIST Cybersecurity Framework 2.0 functions map onto the layers: Govern to the governance layer, Identify to the service catalogue and dependency graph, Protect to the zero-trust and segmentation parameters, Detect to the telemetry and AI analytics layers, Respond to the decision-support and response layers and Recover to the failover and restore mechanics (National Institute of Standards and Technology, 2024). NIST SP 800-61 Revision 3 and SP 800-184 describe the lifecycle the state machine implements (Nelson et al., 2025; Bartock et al., 2016); SP 800-160 Volume 2 and SP 800-207 supply the resiliency techniques and the zero-trust principle that limit lateral movement, and the intrusion results show that the spread rate into the identity provider is the parameter that matters most (Ross et al., 2021; Rose et al., 2020). The SEC rule's disclosure of material incidents within four business days of a materiality determination and CIRCIA's reporting duties, once its implementing rule is in force, need the incident timeline and the cost decomposition the lifecycle produces; Part 500 and the Safeguards Rule ask for the asset inventory, monitoring, incident response and continuity planning that Layers 1, 2, 5 and 6 provide; FedRAMP and SP 800-53 supply the control baselines for federal workloads; ISO 22301, ISO/IEC 27035 and, for EU operations, DORA ask for the recovery objectives, exercises and third-party fallbacks the sweeps quantify (U.S. Securities and Exchange Commission, 2023; U.S. Congress, 2022; New York State Department of Financial Services, 2023; Federal Trade Commission, 2021; FedRAMP, n.d.; Joint Task Force, 2020; International Organization for Standardization, 2019; ISO/IEC, 2023; European Parliament and Council, 2022). The AI components fall under the NIST AI Risk Management Framework and need protection against the evasion and poisoning that the adversarial machine-learning taxonomy describes, which the testbed does not model (Tabassi, 2023; Vassilev et al., 2024). The instruments are read as of their cited editions; implementing rules and guidance change, and the mapping is a design

aid, not legal advice.

Implementation roadmap

Figure 12 arranges the implications as a roadmap with stage gates. Phase 1 (months 0 to 4) builds the foundation: the service catalogue with dependency graph and criticality weights, service-level objectives and error budgets, unified telemetry across operations and security, multi-zone and multi-region patterns for critical services with tested immutable backups, and an incident process aligned with SP 800-61 Revision 3. Phase 2 (months 4 to 9) deploys AI detection and triage calibrated to a false-alarm budget, joins security detections to operations telemetry and runs the models in shadow mode so that their advice can be scored before it is acted on. Phase 3 (months 9 to 15) adds decision support with expected-loss ranking and explanations, orchestrated failover, rollback and restore runbooks with recovery objectives, demand forecasting and canary analysis, and game days and chaos experiments against the decision-support loop. Phase 4 (from month 15) introduces guarded automation for the low-risk playbooks with the confidence threshold, blast-radius cap and verify-and-revert mechanics, retains human approval for failover, isolation and restores, and runs continuous assurance: model monitoring for drift, calibration, automation harm and nuisance incidents, and quarterly exercises, post-incident reviews and disclosure readiness. The feedback loop from Phase 4 to Phase 2 retrains and recalibrates the models as the estate and the threats change, because models trained on one estate drift as it changes (Gama et al., 2014; Lu et al., 2019).

Figure 12: Implementation roadmap for AI-driven cloud resilience and cybersecurity decision support: four phases with stage gates, feedback, governance and enablers



Limitations and future research

Table 16 lists the limitations. The decisive one is that the estate, the disruptions and the operators are synthetic: every number in Section 5 is a consequence of the author's assumptions, and the differences between strategies are differences between assumed capabilities, not measured ones. Human behaviour is reduced to latency, error and bias parameters; the attacker is a stochastic process rather

than an adversary who adapts to detection, targets backups or evades the classifier; and the models are trained on the simulator's own histories, so they know the signatures of the eight classes, whereas real incidents include unknown classes and distribution shift, which argues for the drift monitoring and abstention mechanisms of the roadmap (Shafer & Vovk, 2008; Angelopoulos & Bates, 2023). Future research should replay real incident telemetry through the detection and decision layers in shadow mode, run human-in-the-loop experiments on the testbed with operators making the decisions, add adversarial agents and red-team exercises against the pipeline, and extend the horizon so that learning across incidents and long restores are represented.

Table 16: Limitations of the study and what would address them

Limitation	Why it matters	What would address it
The estate, the disruptions and the operators are synthetic	Every number is a consequence of the author's assumptions; the strategy differences are differences in assumed capabilities, not measured ones	Replay of real incident telemetry through the detection and decision layers; field trials in shadow mode
Human behaviour is reduced to latency, error and bias parameters	Real triage depends on expertise, fatigue, communication and organisational politics; automation bias may differ from the assumed probabilities	Human-in-the-loop experiments with operators on the testbed; game days with recorded decisions
The attacker is a stochastic process, not an adversary	Real attackers adapt to detection, target backups and exploit the response; evasion of the classifier is not modelled	Adversarial agents; red-team exercises against the pipeline; adversarial-ML evaluation
Models are trained on the simulator's own histories	The classifier knows the signatures of the eight classes; real incidents include unknown classes and distribution shift	Out-of-class scenarios; drift monitoring; conformal prediction for abstention
Costs are stylised	The ranking by cost depends on the revenue rate and the labour, capacity and data-loss prices	Organisation-specific cost models; reporting by component (done here) for re-weighting
Dependency graph and deployment patterns are fixed within an episode	Architectural remediation during an incident (emergency re-platforming) is not represented beyond zone migration	Extend the action set; model long-run architecture investment
Regulatory mapping is a reading, not advice	Instruments change; CIRCIA's implementing rule, SEC guidance and state laws evolve	Periodic review by counsel; the mapping is a design aid only
Horizon of a fortnight	Long restores and cumulative learning across incidents are truncated or absent	Longer episodes; multi-incident learning loops

Conclusion

Enterprise IT operations in the United States depend on cloud platforms, third parties and an identity layer whose failures and compromises propagate fast and are now reportable events. This article proposed a framework in which one telemetry pipeline, one set of calibrated models and one incident lifecycle serve both reliability and security incidents, in which actions are ranked by expected loss and automated only under guardrails, and in which governance sets the false-alarm budget, the automation tier and the recovery objectives explicitly. On a transparent synthetic estate exposed to eight disruption classes and operated under five strategies, AI detection and decision support removed most of the weighted downtime of bad changes, denial-of-service attacks, demand surges and third-party failures and turned ransomware intrusions from multi-day restores into contained incidents, with guarded automation adding a further but smaller gain at a small and measurable automation-harm rate; region outages were bounded by the deployment architecture rather than by the operating strategy, and a multi-region application tier moved that bound. The sweeps and the Monte Carlo analysis showed which parameters carry the results: the attacker's dwell and signal strength and the false-alarm budget

for intrusions, the decision latency for the fast-moving classes, and the recovery objectives for the classes that end in failover or restore. None of these numbers describes any real organisation; all of them are reasons to build the foundation, calibrate the detection, deploy decision support in shadow mode and automate the low-risk playbooks under guardrails, in that order, and to measure the timing metrics and side effects that the standards and rules now ask U.S. enterprises to be able to produce.

REFERENCES

[1]. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, *60*, 19–31. https://doi.org/10.1016/j.jnca.2015.11.016

[2]. Ahmed, T., Ghosh, S., Bansal, C., Zimmermann, T., Zhang, X., & Rajmohan, S. (2023). Recommending root-cause and mitigation steps for cloud incidents using large language models. In *Proceedings of the 45th International Conference on Software Engineering (ICSE '23)* (pp. 1737–1749). IEEE. https://doi.org/10.1109/ICSE48619.2023.00149

[3]. Alahmadi, B. A., Axon, L., & Martinovic, I. (2022). 99% false positives: A qualitative study of SOC analysts' perspectives on security alarms. In *Proceedings of the 31st USENIX Security Symposium* (pp. 2783–2800). USENIX Association. https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi

[4]. Alberdi, E., Strigini, L., Povyakalo, A. A., & Ayton, P. (2009). Why are people's decisions sometimes worse with computer support? In *Computer Safety, Reliability, and Security (SAFECOMP 2009), Lecture Notes in Computer Science, vol. 5775* (pp. 18–31). Springer. https://doi.org/10.1007/978-3-642-04468-7_3

[5]. Alshamrani, A., Myneni, S., Chowdhary, A., & Huang, D. (2019). A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials*, *21*(2), 1851–1877. https://doi.org/10.1109/COMST.2019.2891891

[6]. Amazon Web Services. (2021). *Summary of the AWS service event in the Northern Virginia (US-EAST-1) Region*. Amazon Web Services. https://aws.amazon.com/message/12721/

[7]. Amazon Web Services. (2024). *Reliability pillar: AWS Well-Architected Framework*. Amazon Web Services. https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/welcome.html

[8]. Amazon Web Services. (n.d.). *Shared responsibility model*. Amazon Web Services. https://aws.amazon.com/compliance/shared-responsibility-model/

[9]. Angelopoulos, A. N., & Bates, S. (2023). Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, *16*(4), 494–591. https://doi.org/10.1561/2200000101

[10]. Avižienis, A., Laprie, J.-C., Randell, B., & Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, *1*(1), 11–33. https://doi.org/10.1109/TDSC.2004.2

[11]. Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security*, *3*(3), 186–205. https://doi.org/10.1145/357830.357849

[12]. Bahrami, P. N., Dehghantanha, A., Dargahi, T., Parizi, R. M., Choo, K.-K. R., & Javadi, H. H. S. (2019). Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tactics, techniques, and procedures. *Journal of Information Processing Systems*, *15*(4), 865–889. https://doi.org/10.3745/JIPS.03.0126

[13]. Bainbridge, L. (1983). Ironies of automation. *Automatica*, *19*(6), 775–779. https://doi.org/10.1016/0005-1098(83)90046-8

[14]. Bansal, G., Wu, T., Zhou, J., Fok, R., Nushi, B., Kamar, E., Ribeiro, M. T., & Weld, D. S. (2021). Does the whole exceed its parts? The effect of AI explanations on complementary team performance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)* (pp. 1–16). ACM. https://doi.org/10.1145/3411764.3445717

[15]. Bartock, M., Cichonski, J., Souppaya, M., Smith, M., Witte, G., & Scarfone, K. (2016). *Guide for cybersecurity event recovery (NIST Special Publication 800-184)*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-184

[16]. Basiri, A., Behnam, N., de Rooij, R. C. M., Hochstein, L., Kosewski, L., Reynolds, J., & Rosenthal, C. (2016). Chaos engineering. *IEEE Software*, *33*(3), 35–41. https://doi.org/10.1109/MS.2016.60

[17]. Beerman, J., Berent, D., Falter, Z., & Bhunia, S. (2023). A review of Colonial Pipeline ransomware attack. *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW)*, 8–15. https://doi.org/10.1109/CCGridW59191.2023.00017

[18]. Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (Eds.). (2016). *Site reliability engineering: How Google runs production systems*. O'Reilly Media. https://www.oreilly.com/library/view/site-reliability-engineering/9781491929117/

[19]. Bodeau, D. J., & Graubart, R. (2011). *Cyber resiliency engineering framework* (MTR110237). The MITRE Corporation. https://www.mitre.org/news-insights/publication/cyber-resiliency-engineering-framework

[20]. Bruneau, M., Chang, S. E., Eguchi, R. T., Lee, G. C., O'Rourke, T. D., Reinhorn, A. M., Shinozuka, M., Tierney, K., Wallace, W. A., & von Winterfeldt, D. (2003). A framework to quantitatively assess and enhance the seismic resilience of communities. *Earthquake Spectra*, *19*(4), 733–752. https://doi.org/10.1193/1.1623497

[21]. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, *18*(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

[22]. Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, Omega, and Kubernetes. *Communications of the ACM*, *59*(5), 50–57. https://doi.org/10.1145/2890784

[23]. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), Article 15. https://doi.org/10.1145/1541880.1541882

[24]. Chen, J., He, X., Lin, Q., Xu, Y., Zhang, H., Hao, D., Gao, F., Xu, Z., Dang, Y., & Zhang, D. (2019). An empirical

investigation of incident triage for online service systems. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP '19)* (pp. 111–120). IEEE. https://doi.org/10.1109/ICSE-SEIP.2019.00020

[25]. Chen, Y., Xie, H., Ma, M., Kang, Y., Gao, X., Shi, L., Cao, Y., Gao, X., Fan, H., Wen, M., Zeng, J., Ghosh, S., Zhang, X., Zhang, C., Lin, Q., Rajmohan, S., Zhang, D., & Xu, T. (2024). Automatic root cause analysis via large language models for cloud incidents. In *Proceedings of the Nineteenth European Conference on Computer Systems (EuroSys '24)* (pp. 674–688). ACM. https://doi.org/10.1145/3627703.3629553

[26]. Chen, Z., Kang, Y., Li, L., Zhang, X., Zhang, H., Xu, H., Zhou, Y., Yang, L., Sun, J., Xu, Z., Dang, Y., Gao, F., Zhao, P., Qiao, B., Lin, Q., Zhang, D., & Lyu, M. R. (2020). Towards intelligent incident management: Why we need it and how we make it. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '20)* (pp. 1487–1497). ACM. https://doi.org/10.1145/3368089.3417055

[27]. Connolly, L. Y., & Wall, D. S. (2019). The rise of crypto-ransomware in a changing cybercrime landscape: Taxonomising countermeasures. *Computers & Security*, *87*, Article 101568. https://doi.org/10.1016/j.cose.2019.101568

[28]. CrowdStrike. (2024). *External technical root cause analysis: Channel File 291*. CrowdStrike. https://www.crowdstrike.com/content/dam/crowdstrike/www/en-us/wp/2024/08/Channel-File-291-Incident-Root-Cause-Analysis-08.06.2024.pdf

[29]. Cummings, M. L. (2004). Automation bias in intelligent time critical decision support systems. In *AIAA 1st Intelligent Systems Technical Conference*. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2004-6313

[30]. Cummings, M. L. (2006). Automation and accountability in decision support system interface design. *Journal of Technology Studies*, *32*(1), 23–31. https://doi.org/10.21061/jots.v32i1.a.4

[31]. Cuppens, F., & Miège, A. (2002). Alert correlation in a cooperative intrusion detection framework. In *Proceedings 2002 IEEE Symposium on Security and Privacy* (pp. 202–215). IEEE. https://doi.org/10.1109/SECPRI.2002.1004372

[32]. Cybersecurity and Infrastructure Security Agency, Federal Bureau of Investigation, National Security Agency, & Multi-State Information Sharing and Analysis Center. (2023). *#StopRansomware guide*. Cybersecurity and Infrastructure Security Agency. https://www.cisa.gov/resources-tools/resources/stopransomware-guide

[33]. Cybersecurity and Infrastructure Security Agency. (2023a). *Zero trust maturity model, version 2.0*. Cybersecurity and Infrastructure Security Agency. https://www.cisa.gov/sites/default/files/2023-04/CISA_Zero_Trust_Maturity_Model_Version_2_508c.pdf

[34]. Cybersecurity and Infrastructure Security Agency. (2023b). *Cross-sector cybersecurity performance goals*. Cybersecurity and Infrastructure Security Agency. https://www.cisa.gov/cross-sector-cybersecurity-performance-goals

[35]. Dang, Y., Lin, Q., & Huang, P. (2019). AIOps: Real-world challenges and research innovations. In *Proceedings of the 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion '19)* (pp. 4–5). IEEE. https://doi.org/10.1109/ICSE-Companion.2019.00023

[36]. Dean, J., & Barroso, L. A. (2013). The tail at scale. *Communications of the ACM*, *56*(2), 74–80. https://doi.org/10.1145/2408776.2408794

[37]. Dempsey, K., Chawla, N. S., Johnson, A., Johnston, R., Jones, A. C., Orebaugh, A., Scholl, M., & Stine, K. (2011). *Information security continuous monitoring (ISCM) for federal information systems and organizations (NIST Special Publication 800-137)*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-137

[38]. Donnellan, D., Lawrence, A., & Weinschenk, R. (2025). *Annual outage analysis 2025*. Uptime Institute. https://uptimeinstitute.com/resources/research-and-reports/annual-outage-analysis-2025

[39]. Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)* (pp. 1285–1298). ACM. https://doi.org/10.1145/3133956.3134015

[40]. Eling, M., & Wirfs, J. (2019). What are the actual costs of cyber risk events? *European Journal of Operational Research*, *272*(3), 1109–1119. https://doi.org/10.1016/j.ejor.2018.07.021

[41]. Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, *37*(1), 32–64. https://doi.org/10.1518/001872095779049543

[42]. Engelen, G., Rimmer, V., & Joosen, W. (2021). Troubleshooting an intrusion detection dataset: The CICIDS2017 case study. In *2021 IEEE Security and Privacy Workshops (SPW)* (pp. 7–12). IEEE. https://doi.org/10.1109/SPW53761.2021.00009

[43]. Epstein, J. M. (1999). Agent-based computational models and generative social science. *Complexity*, *4*(5), 41–60. https://doi.org/10.1002/(SICI)1099-0526(199905/06)4:5<41::AID-CPLX9>3.0.CO;2-F

[44]. European Parliament and Council. (2022). *Regulation (EU) 2022/2554 on digital operational resilience for the financial sector (DORA)*. Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2022/2554/oj

[45]. European Union Agency for Cybersecurity. (2024). *ENISA threat landscape 2024*. European Union Agency for Cybersecurity. https://doi.org/10.2824/0710888

[46]. Federal Bureau of Investigation, Cybersecurity and Infrastructure Security Agency, & U.S. Department of Health and Human Services. (2023). *#StopRansomware: ALPHV Blackcat (Joint Cybersecurity Advisory AA23-353A, updated February 27, 2024)*. CISA. https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-353a

[47]. Federal Trade Commission. (2021). *Standards for safeguarding customer information (final rule, 16 CFR Part 314)* (86 FR 70272). Federal Register. https://www.federalregister.gov/documents/2021/12/09/2021-25736/standards-for-

safeguarding-customer-information

[48]. FedRAMP. (n.d.). *Understanding baselines and impact levels for FedRAMP authorizations*. https://www.fedramp.gov/baselines/

[49]. Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, *50*, 102419. https://doi.org/10.1016/j.jisa.2019.102419

[50]. Ford, D., Labelle, F., Popovici, F. I., Stokely, M., Truong, V.-A., Barroso, L., Grimes, C., & Quinlan, S. (2010). Availability in globally distributed storage systems. In *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI '10)* (pp. 61–74). USENIX Association. https://www.usenix.org/conference/osdi10/availability-globally-distributed-storage-systems

[51]. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys, 46*(4), Article 44. https://doi.org/10.1145/2523813

[52]. Ganin, A. A., Massaro, E., Gutfraind, A., Steen, N., Keisler, J. M., Kott, A., Mangoubi, R., & Linkov, I. (2016). Operational resilience: Concepts, design and analysis. *Scientific Reports*, *6*, 19540. https://doi.org/10.1038/srep19540

[53]. Ganin, A. A., Quach, P., Panwar, M., Collier, Z. A., Keisler, J. M., Marchese, D., & Linkov, I. (2020). Multicriteria decision framework for cybersecurity risk assessment and management. *Risk Analysis*, *40*(1), 183–199. https://doi.org/10.1111/risa.12891

[54]. Goddard, K., Roudsari, A., & Wyatt, J. C. (2012). Automation bias: A systematic review of frequency, effect mediators, and mitigators. *Journal of the American Medical Informatics Association*, *19*(1), 121–127. https://doi.org/10.1136/amiajnl-2011-000089

[55]. Gordon, L. A., & Loeb, M. P. (2002). The economics of information security investment. *ACM Transactions on Information and System Security*, *5*(4), 438–457. https://doi.org/10.1145/581271.581274

[56]. Gunawi, H. S., Hao, M., Leesatapornwongsa, T., Patana-anake, T., Do, T., Adityatama, J., Eliazar, K. J., Laksono, A., Lukman, J. F., Martin, V., & Satria, A. D. (2014). What bugs live in the cloud? A study of 3000+ issues in cloud systems. In *Proceedings of the ACM Symposium on Cloud Computing (SoCC '14)* (pp. 1–14). ACM. https://doi.org/10.1145/2670979.2670986

[57]. Gunawi, H. S., Hao, M., Suminto, R. O., Laksono, A., Satria, A. D., Adityatama, J., & Eliazar, K. J. (2016). Why does the cloud stop computing? Lessons from hundreds of service outages. In *Proceedings of the Seventh ACM Symposium on Cloud Computing (SoCC '16)* (pp. 1–16). ACM. https://doi.org/10.1145/2987550.2987583

[58]. Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning* (PMLR 70, pp. 1321–1330). https://proceedings.mlr.press/v70/guo17a.html

[59]. Hamilton, J. (2007). On designing and deploying Internet-scale services. In *Proceedings of the 21st Large Installation System Administration Conference (LISA '07)* (pp. 231–242). USENIX Association. https://www.usenix.org/conference/lisa-07/designing-and-deploying-internet-scale-services

[60]. Hauer, T., Hoffmann, P., Lunney, J., Ardelean, D., & Diwan, A. (2020). Meaningful availability. In *Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI '20)* (pp. 545–557). USENIX Association. https://www.usenix.org/conference/nsdi20/presentation/hauer

[61]. He, S., He, P., Chen, Z., Yang, T., Su, Y., & Lyu, M. R. (2022a). A survey on automated log analysis for reliability engineering. *ACM Computing Surveys*, *54*(6), 1–37. https://doi.org/10.1145/3460345

[62]. He, Y., Huang, D., Chen, L., Ni, Y., & Ma, X. (2022b). A survey on zero trust architecture: Challenges and future trends. *Wireless Communications and Mobile Computing*, *2022*, 6476274. https://doi.org/10.1155/2022/6476274

[63]. Henry, D., & Ramirez-Marquez, J. E. (2012). Generic metrics and quantitative approaches for system resilience as a function of time. *Reliability Engineering & System Safety*, *99*, 114–122. https://doi.org/10.1016/j.ress.2011.09.002

[64]. Hoff, K. A., & Bashir, M. (2015). Trust in automation: Integrating empirical evidence on factors that influence trust. *Human Factors*, *57*(3), 407–434. https://doi.org/10.1177/0018720814547570

[65]. Hollnagel, E., Woods, D. D., & Leveson, N. (Eds.). (2006). *Resilience engineering: Concepts and precepts*. Ashgate. https://www.routledge.com/Resilience-Engineering-Concepts-and-Precepts/Hollnagel-Woods/p/book/9780754649045

[66]. Hosseini, S., Barker, K., & Ramirez-Marquez, J. E. (2016). A review of definitions and measures of system resilience. *Reliability Engineering & System Safety*, *145*, 47–61. https://doi.org/10.1016/j.ress.2015.08.006

[67]. IBM. (2025). *Cost of a data breach report 2025*. IBM. https://www.ibm.com/reports/data-breach

[68]. International Organization for Standardization. (2019). *ISO 22301:2019 Security and resilience: Business continuity management systems: Requirements*. ISO. https://www.iso.org/standard/75106.html

[69]. ISO/IEC. (2023). *ISO/IEC 27035-1:2023 Information technology: Information security incident management: Part 1: Principles and process*. ISO. https://www.iso.org/standard/78973.html

[70]. Jansen, W., & Grance, T. (2011). *Guidelines on security and privacy in public cloud computing (NIST Special Publication 800-144)*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-144

[71]. Joint Task Force Transformation Initiative. (2012). *Guide for conducting risk assessments* (NIST Special Publication 800-30 Rev. 1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-30r1

[72]. Joint Task Force. (2020). *Security and privacy controls for information systems and organizations* (NIST Special Publication 800-53 Rev. 5). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-53r5

[73]. Julisch, K. (2003). Clustering intrusion detection alarms to support root cause analysis. *ACM Transactions on Information and System Security*, *6*(4), 443–471. https://doi.org/10.1145/950191.950192

[74]. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient

gradient boosting decision tree. *Advances in Neural Information Processing Systems, 30*, 3146–3154. https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree

[75]. Keeney, R. L., & Raiffa, H. (1993). *Decisions with multiple objectives: Preferences and value tradeoffs*. Cambridge University Press. https://doi.org/10.1017/CBO9781139174084

[76]. Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer*, *36*(1), 41–50. https://doi.org/10.1109/MC.2003.1160055

[77]. Kokulu, F. B., Soneji, A., Bao, T., Shoshitaishvili, Y., Zhao, Z., Doupé, A., & Ahn, G.-J. (2019). Matched and mismatched SOCs: A qualitative study on security operations center issues. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)* (pp. 1955–1970). ACM. https://doi.org/10.1145/3319535.3354239

[78]. Lallie, H. S., Debattista, K., & Bal, J. (2020). A review of attack graph and attack tree visual syntax in cyber security. *Computer Science Review*, *35*, Article 100219. https://doi.org/10.1016/j.cosrev.2019.100219

[79]. Law, A. M. (2015). *Simulation modeling and analysis* (5th ed.). McGraw-Hill Education. https://www.mheducation.com/

[80]. Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, *46*(1), 50–80. https://doi.org/10.1518/hfes.46.1.50.30392

[81]. Li, L., Zhang, X., Zhao, X., Zhang, H., Kang, Y., Zhao, P., Qiao, B., He, S., Lee, P., Sun, J., Gao, F., Yang, L., Lin, Q., Rajmohan, S., Xu, Z., & Zhang, D. (2021). Fighting the fog of war: Automated incident detection for cloud systems. In *Proceedings of the 2021 USENIX Annual Technical Conference (USENIX ATC '21)* (pp. 131–146). USENIX Association. https://www.usenix.org/conference/atc21/presentation/li-liqun

[82]. Linkov, I., Eisenberg, D. A., Plourde, K., Seager, T. P., Allen, J., & Kott, A. (2013). Resilience metrics for cyber systems. *Environment Systems and Decisions*, *33*(4), 471–476. https://doi.org/10.1007/s10669-013-9485-y

[83]. Lipton, Z. C. (2018). The mythos of model interpretability. *Queue, 16*(3), 31–57. https://doi.org/10.1145/3236386.3241340

[84]. Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 413–422). IEEE. https://doi.org/10.1109/ICDM.2008.17

[85]. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering, 31*(12), 2346–2363. https://doi.org/10.1109/TKDE.2018.2876857

[86]. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems, 30*, 4765–4774. https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions

[87]. Lyu, Y., Rajbahadur, G. K., Lin, D., Chen, B., & Jiang, Z. M. (2022). Towards a consistent interpretation of AIOps models. *ACM Transactions on Software Engineering and Methodology*, *31*(1), Article 16. https://doi.org/10.1145/3488269

[88]. Macal, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation*, *4*(3), 151–162. https://doi.org/10.1057/jos.2010.3

[89]. Manshaei, M. H., Zhu, Q., Alpcan, T., Başar, T., & Hubaux, J.-P. (2013). Game theory meets network security and privacy. *ACM Computing Surveys*, *45*(3), 1–39. https://doi.org/10.1145/2480741.2480742

[90]. McHugh, J. (2000). Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *ACM Transactions on Information and System Security*, *3*(4), 262–294. https://doi.org/10.1145/382912.382923

[91]. Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing (NIST Special Publication 800-145)*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-145

[92]. Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*, *34*(2), 39–53. https://doi.org/10.1145/997150.997156

[93]. National Institute of Standards and Technology. (2024). *The NIST Cybersecurity Framework (CSF) 2.0* (NIST CSWP 29). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.CSWP.29

[94]. Nelson, A., Rekhi, S., Souppaya, M., & Scarfone, K. (2025). *Incident response recommendations and considerations for cybersecurity risk management: A CSF 2.0 community profile (NIST Special Publication 800-61 Rev. 3)*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-61r3

[95]. New York State Department of Financial Services. (2023). *Second amendment to 23 NYCRR 500: Cybersecurity requirements for financial services companies*. New York State Department of Financial Services. https://www.dfs.ny.gov/cybersecurity/23-NYCRR-Part-500

[96]. Notaro, P., Cardoso, J., & Gerndt, M. (2021). A survey of AIOps methods for failure management. *ACM Transactions on Intelligent Systems and Technology*, *12*(6), 1–45. https://doi.org/10.1145/3483424

[97]. Ohm, M., Plate, H., Sykosch, A., & Meier, M. (2020). Backstabber's knife collection: A review of open source software supply chain attacks. In *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2020), Lecture Notes in Computer Science, vol. 12223* (pp. 23–43). Springer. https://doi.org/10.1007/978-3-030-52683-2_2

[98]. Okati, N., De, A., & Gomez-Rodriguez, M. (2021). Differentiable learning under triage. In *Advances in Neural Information Processing Systems 34*. Curran Associates. https://arxiv.org/abs/2103.08902

[99]. Oppenheimer, D., Ganapathi, A., & Patterson, D. A. (2003). Why do Internet services fail, and what can be done about it? In *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems (USITS '03)*. USENIX Association. https://www.usenix.org/legacy/events/usits03/tech/oppenheimer.html

[100]. Ou, X., Boyer, W. F., & McQueen, M. A. (2006). A scalable approach to attack graph generation. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)* (pp. 336–345). ACM.

https://doi.org/10.1145/1180405.1180446
[101]. Oz, H., Aris, A., Levi, A., & Uluagac, A. S. (2022). A survey on ransomware: Evolution, taxonomy, and defense solutions. *ACM Computing Surveys*, *54*(11s), Article 238. https://doi.org/10.1145/3514229
[102]. Parasuraman, R., & Manzey, D. H. (2010). Complacency and bias in human use of automation: An attentional integration. *Human Factors*, *52*(3), 381–410. https://doi.org/10.1177/0018720810376055
[103]. Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, *30*(3), 286–297. https://doi.org/10.1109/3468.844354
[104]. Patterson, D., Brown, A., Broadwell, P., Candea, G., Chen, M., Cutler, J., Enriquez, P., Fox, A., Kiciman, E., Merzbacher, M., Oppenheimer, D., Sastry, N., Tetzlaff, W., Traupman, J., & Treuhaft, N. (2002). *Recovery-oriented computing (ROC): Motivation, definition, techniques, and case studies* (UCB//CSD-02-1175). University of California, Berkeley. https://www2.eecs.berkeley.edu/Pubs/TechRpts/2002/5574.html
[105]. Peisert, S., Schneier, B., Okhravi, H., Massacci, F., Benzel, T., Landwehr, C., Mannan, M., Mirkovic, J., Prakash, A., & Michael, J. B. (2021). Perspectives on the SolarWinds incident. *IEEE Security & Privacy*, *19*(2), 7–13. https://doi.org/10.1109/MSEC.2021.3051235
[106]. Pietraszek, T. (2004). Using adaptive alert classification to reduce false positives in intrusion detection. In *Recent Advances in Intrusion Detection (RAID 2004), Lecture Notes in Computer Science 3224* (pp. 102–124). Springer. https://doi.org/10.1007/978-3-540-30143-1_6
[107]. Romanosky, S. (2016). Examining the costs and causes of cyber incidents. *Journal of Cybersecurity*, *2*(2), 121–135. https://doi.org/10.1093/cybsec/tyw001
[108]. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero trust architecture (NIST Special Publication 800-207)*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207
[109]. Ross, R., Pillitteri, V., Graubart, R., Bodeau, D., & McQuaid, R. (2021). *Developing cyber-resilient systems: A systems security engineering approach (NIST Special Publication 800-160, Vol. 2, Rev. 1)*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-160v2r1
[110]. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*(5), 206–215. https://doi.org/10.1038/s42256-019-0048-x
[111]. Saaty, T. L. (1990). How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, *48*(1), 9–26. https://doi.org/10.1016/0377-2217(90)90057-I
[112]. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., & Tarantola, S. (2010). Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Computer Physics Communications*, *181*(2), 259–270. https://doi.org/10.1016/j.cpc.2009.09.018
[113]. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., & Tarantola, S. (2008). *Global sensitivity analysis: The primer*. Wiley. https://doi.org/10.1002/9780470725184
[114]. Sargent, R. G. (2013). Verification and validation of simulation models. *Journal of Simulation*, *7*(1), 12–24. https://doi.org/10.1057/jos.2012.20
[115]. Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction. *Journal of Machine Learning Research*, *9*, 371–421. https://jmlr.org/papers/v9/shafer08a.html
[116]. Shameli-Sendi, A., Aghababaei-Barzegar, R., & Cheriet, M. (2016). Taxonomy of information security risk assessment (ISRA). *Computers & Security*, *57*, 14–30. https://doi.org/10.1016/j.cose.2015.11.001
[117]. Soldani, J., & Brogi, A. (2023). Anomaly detection and failure root cause analysis in (micro)service-based cloud applications: A survey. *ACM Computing Surveys*, *55*(3), 1–39. https://doi.org/10.1145/3501297
[118]. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy* (pp. 305–316). IEEE. https://doi.org/10.1109/SP.2010.25
[119]. Spathoulas, G. P., & Katsikas, S. K. (2010). Reducing false positives in intrusion detection systems. *Computers & Security*, *29*(1), 35–44. https://doi.org/10.1016/j.cose.2009.07.008
[120]. Sterbenz, J. P. G., Hutchison, D., Çetinkaya, E. K., Jabbar, A., Rohrer, J. P., Schöller, M., & Smith, P. (2010). Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks*, *54*(8), 1245–1265. https://doi.org/10.1016/j.comnet.2010.03.005
[121]. Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., & Thomas, C. B. (2018). *MITRE ATT&CK: Design and philosophy*. The MITRE Corporation. https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf
[122]. Swanson, M., Bowen, P., Phillips, A. W., Gallup, D., & Lynes, D. (2010). *Contingency planning guide for federal information systems (NIST Special Publication 800-34 Rev. 1)*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-34r1
[123]. Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z., & Doss, R. (2022). Zero trust architecture (ZTA): A comprehensive survey. *IEEE Access*, *10*, 57143–57179. https://doi.org/10.1109/ACCESS.2022.3174679
[124]. Tabassi, E. (2023). *Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1)*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-1
[125]. The MITRE Corporation. (n.d.). *D3FEND: A knowledge graph of cybersecurity countermeasures*. The MITRE Corporation. https://d3fend.mitre.org/
[126]. Treynor, B., Dahlin, M., Rau, V., & Beyer, B. (2017). The calculus of service availability. *Communications of the ACM*, *60*(9), 42–47. https://doi.org/10.1145/3080202
[127]. U.S. Congress. (2022). *Cyber Incident Reporting for Critical Infrastructure Act of 2022 (Division Y of the Consolidated Appropriations Act, 2022, Public Law 117-103)*. U.S. Government Publishing Office.

https://www.govinfo.gov/app/details/PLAW-117publ103

[128]. U.S. Department of Health and Human Services, Office for Civil Rights. (2024). *Change Healthcare cybersecurity incident frequently asked questions*. HHS. https://www.hhs.gov/hipaa/for-professionals/special-topics/change-healthcare-cybersecurity-incident-frequently-asked-questions/index.html

[129]. U.S. Securities and Exchange Commission. (2023). *Cybersecurity risk management, strategy, governance, and incident disclosure (final rule, 17 CFR Parts 229, 232, 239, 240 and 249)* (88 FR 51896). Federal Register. https://www.federalregister.gov/documents/2023/08/04/2023-16194/cybersecurity-risk-management-strategy-governance-and-incident-disclosure

[130]. Valdes, A., & Skinner, K. (2001). Probabilistic alert correlation. In *Recent Advances in Intrusion Detection (RAID 2001), Lecture Notes in Computer Science 2212* (pp. 54–68). Springer. https://doi.org/10.1007/3-540-45474-8_4

[131]. Vassilev, A., Oprea, A., Fordyce, A., & Anderson, H. (2024). *Adversarial machine learning: A taxonomy and terminology of attacks and mitigations* (NIST AI 100-2e2023). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-2e2023

[132]. Veeramachaneni, K., Arnaldo, I., Korrapati, V., Bassias, C., & Li, K. (2016). AI²: Training a big data machine to defend. In *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)* (pp. 49–54). IEEE. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2016.79

[133]. Verizon. (2025). *2025 Data Breach Investigations Report*. Verizon. https://www.verizon.com/business/resources/reports/2025-dbir-data-breach-investigations-report.pdf

[134]. Verma, A., Pedrosa, L., Korupolu, M., Oppenheimer, D., Tune, E., & Wilkes, J. (2015). Large-scale cluster management at Google with Borg. In *Proceedings of the Tenth European Conference on Computer Systems (EuroSys '15)* (pp. 1–17). ACM. https://doi.org/10.1145/2741948.2741964

[135]. Vielberth, M., Böhm, F., Fichtinger, I., & Pernul, G. (2020). Security operations center: A systematic study and open challenges. *IEEE Access*, *8*, 227756–227779. https://doi.org/10.1109/ACCESS.2020.3045514

[136]. Wang, L., Zhao, N., Chen, J., Li, P., Zhang, W., & Sui, K. (2020). Root-cause metric location for microservice systems via log anomaly detection. In *Proceedings of the 2020 IEEE International Conference on Web Services (ICWS 2020)* (pp. 142–150). IEEE. https://doi.org/10.1109/ICWS49710.2020.00026

[137]. Weston, D. (2024). *Helping our customers through the CrowdStrike outage*. Microsoft. https://blogs.microsoft.com/blog/2024/07/20/helping-our-customers-through-the-crowdstrike-outage/

[138]. Wolff, J., & Lehr, W. H. (2017). *Degrees of ignorance about the costs of data breaches: What policymakers can and can't do about the lack of good empirical data* (SSRN Scholarly Paper 2943867). SSRN. https://doi.org/10.2139/ssrn.2943867

[139]. Woods, D. D. (2015). Four concepts for resilience and the implications for the future of resilience engineering. *Reliability Engineering & System Safety*, *141*, 5–9. https://doi.org/10.1016/j.ress.2015.03.018

[140]. Xie, P., Li, J. H., Ou, X., Liu, P., & Levy, R. (2010). Using Bayesian networks for cyber security analysis. In *Proceedings of the 2010 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '10)* (pp. 211–220). IEEE. https://doi.org/10.1109/DSN.2010.5544924

[141]. Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Communications Surveys & Tutorials*, *15*(4), 2046–2069. https://doi.org/10.1109/SURV.2013.031413.00127

[142]. Zhou, X., Peng, X., Xie, T., Sun, J., Ji, C., Li, W., & Ding, D. (2021). Fault analysis and debugging of microservice systems: Industrial survey, benchmark system, and empirical study. *IEEE Transactions on Software Engineering*, *47*(2), 243–260. https://doi.org/10.1109/TSE.2018.2887384

[143]. Aditya, D., & Ahmed, H. C. H. R. (2022). Decision Support Systems for Last-Mile Logistics Optimization: A Data-Driven Framework for e-Commerce Fulfillment Networks. *International Journal of Scientific Interdisciplinary Research*, *12*(08), 01–49. https://doi.org/10.63125/h46vp562

[144]. Adnan, S. (2024). Human Trust in AI-Assisted Project Decision Systems: A Mixed-Methods Analysis of Explainability, Adoption, and Managerial Decision Quality. *Journal of Sustainable Development and Policy*, *3*(04), 287-332. https://doi.org/10.63125/9c139h44

[145]. Aminul Islam, M. (2024). Interpreting Cyber-Threat Indicators in Enterprise Memory Forensics: A Mixed-Methods Study of Machine Learning Performance and Explainability. *Review of Applied Science and Technology*, *14*(12), 01–13. https://doi.org/10.63125/2p8hte92

[146]. Aminul Islam, M., & Md, A. (2023). Adaptive Class-Imbalance Learning for High-Sensitivity Fraud Detection in Large-Scale Digital Transaction Networks. *International Journal of Business and Economics Insights*, *13*(09), 01-33. https://doi.org/10.63125/bmxwv741

[147]. Asheke, M. (2023). Machine Learning–Based Predictive Quality Modeling for Defect Detection and Process Optimization in Advanced Manufacturing Systems. *Review of Applied Science and Technology*, *13*(12), 01–49. https://doi.org/10.63125/thy73b16

[148]. Asheke, M., & Md Maksudul, H. (2022). AI-Driven Demand Forecasting for Sporting-Goods and Teamwear Supply Chains Under Seasonal and Tournament Demand. *International Journal of Business and Economics Insights*, *2*(1), 73–110. https://doi.org/10.63125/ed6by557

[149]. Asheke, M., & Mohammad Badrul, A. (2022). Development Of Functionalized Cellulose Nanofiber Adsorbents for Selective Critical-Metal Recovery from Industrial Wastewater. *American Journal of Scholarly Research and Innovation*, *1*(02), 270–320. https://doi.org/10.63125/v6fxzq09

[150]. Ashik, M. (2024). Meta-Analysis of Risk-Based Inspection Strategies for Improving Reliability and Safety Across Critical Energy Infrastructure Systems. *American Journal of Advanced Technology and Engineering Solutions*, *14*(09), 01-43. https://doi.org/10.63125/tgttkt09

[151]. Ashik, M. (2025). PRISMA-Based Systematic Literature Review of Reliability-Centered Turnaround Planning for Reducing Downtime in Power Generation Facilities. *American Journal of Scholarly Research and Innovation*, *4*(01), 1036–1080. https://doi.org/10.63125/2878n718

[152]. Ashik, M. (2026). RCM and PSM Strategies for Strengthening Asset Integrity Across U.S. Energy Facilities: A Systematic Meta-Analysis. *American Journal of Interdisciplinary Studies*, *16*(06), 38-81. https://doi.org/10.63125/p87z7q36

[153]. Hasib, A. (2026). AI-Driven Predictive Supply Chain Resilience Framework for U.S. Manufacturing Systems: An Empirical Analysis. *American Journal of Data Science and Analytics*, *7*(04), 45-80. https://doi.org/10.63125/d8swft81

[154]. Kynatun, M. (2025). Integrating AI-Driven Security Analytics and IT Governance Frameworks for Cybersecurity Decision Support in U.S. Financial Information Systems. *American Journal of Advanced Technology and Engineering Solutions*, *15*(12), 01-42. https://doi.org/10.63125/kzcdyz73

[155]. Kynatun, M. (2026). Artificial Intelligence Integrated MIS Framework for Proactive Cyber Threat Intelligence and Organizational Security Resilience. *American Journal of Scholarly Research and Innovation*, *16*(09), 40–78. https://doi.org/10.63125/t98tc495

[156]. Kynatun, M., & Saba, A. (2024). Machine Learning Enhanced Intrusion Detection and Anomaly Classification for MIS-Driven Enterprise Network Security. *International Journal of Scientific Interdisciplinary Research*, *14*(09), 01–41. https://doi.org/10.63125/xk6kpc72

[157]. Maishar Mostari, Z. (2025). Privacy-Preserving AI-Based Cyber Threat Detection in Financial Organizations: A Governance and Business Value Framework. *American Journal of Interdisciplinary Studies*, *15*(04), 01-51. https://doi.org/10.63125/p8fct530

[158]. Maishar Mostari, Z. (2026). A Systematic Review of Machine Learning Applications in Cybersecurity Threat Detection and Classification (2018–2026). *American Journal of Data Science and Analytics*, *7*(06), 640-680. https://doi.org/10.63125/6kg7kg75

[159]. Mansura Akter, E., & Sumaiya, M. (2022). Antimicrobial Resistance Profiles of Acinetobacter Baumannii and Vibrio Cholerae in Hospital Wastewater and Adjacent Household Water Sources. *American Journal of Data Science and Analytics*, *3*(09), 01-55. https://doi.org/10.63125/gbervt14

[160]. Masrul, H. (2025). Explainable Machine Learning for Real-Time Financial Fraud Detection and Risk-Based Transaction Monitoring in U.S. Banking and Digital Payment Systems. *American Journal of Data Science and Analytics*, *6*(10), 70-108. https://doi.org/10.63125/4h9v1876

[161]. Md Hasinur, R. (2026). Explainable Machine Learning for Loan-Delinquency Prediction in U.S. Community Banks: A Quantitative Study Using Gradient Boosting and SHAP. *American Journal of Data Science and Analytics*, *7*(01), 280-316. https://doi.org/10.63125/mfv3c059

[162]. Md Imran, H., & Jannatul, F. (2022). Molecular and Precision Rehabilitation Molecular Mechanisms Linking Mitochondrial Dysfunction, Inflammation, and Muscle Degeneration with Functional Decline and Rehabilitation Response in Neuromuscular Disease. *American Journal of Interdisciplinary Studies*, *12*(08), 01-44. https://doi.org/10.63125/0ym5bh79

[163]. Md Maksudul, H. (2024). Supplier-Risk Scoring and Procurement Resilience for Distributed Athletic-Apparel and Equipment Supply Networks. *American Journal of Advanced Technology and Engineering Solutions*, *14*(09), 44-90. https://doi.org/10.63125/a9fmgj50

[164]. Md Maksudul, H. (2025). Data-Driven Fulfillment Optimization for Time-Sensitive Teamwear and Sporting Goods Distribution. *American Journal of Scholarly Research and Innovation*, *4*(01), 1120–1173. https://doi.org/10.63125/rpezfz61

[165]. Md Maksudul, H. (2026). Generative AI Decision Support for Procurement, Inventory, And Operational Planning in U.S. Sports Businesses. *American Journal of Interdisciplinary Studies*, *16*(04), 01-55. https://doi.org/10.63125/ngtzw226

[166]. Md Maksudul, H., & Md Tazul Islam, C. (2022). Machine Learning-Based Cash-Flow Forecasting for Improving Financial Planning in U.S. Small and Mid-Sized Businesses. *International Journal of Scientific Interdisciplinary Research*, *12*(12), 01–38. https://doi.org/10.63125/w3pgjx97

[167]. Md Mohsin, B. (2026). Nurse-Led Triage Protocols in Emergency and Trauma Care: A PRISMA-Guided Systematic Review. *American Journal of Health and Medical Sciences*, *16*(09), 01–40. https://doi.org/10.63125/21k00924

[168]. Md Momen Shahriar, J. (2025). Viscoelastic Creep Characterization of LDPE Under Multilevel Stress and Temperature Using Time-Temperature Superposition and Constitutive Modeling. *American Journal of Scholarly Research and Innovation*, *15*(11), 01–38. https://doi.org/10.63125/2edstd86

[169]. Md Momen Shahriar, J. (2026). Physics-Informed Machine Learning for Long-Term Polymer Performance Prediction from Extrusion Parameters, Material Properties, and Creep Data. *American Journal of Interdisciplinary Studies*, *16*(05), 01-41. https://doi.org/10.63125/8gk5c668

[170]. Md Shahid, N. (2023). Digital Twin-Based Network Reliability Modeling for LAN And WAN Infrastructure Resilience. *International Journal of Scientific Interdisciplinary Research*, *4*(4), 577–629. https://doi.org/10.63125/ahjvxj14

[171]. Md Shahid, N. (2024). Predictive Analytics Models for Network Traffic Optimization in Distributed Enterprise Environments. *Review of Applied Science and Technology*, *3*(01), 367-419. https://doi.org/10.63125/qm6cyf84

[172]. Md Shahid, N. (2025). Blockchain-Based Secure Data Routing Framework for U.S. Industrial Communication Networks. *American Journal of Scholarly Research and Innovation*, *4*(01), 990–1035. https://doi.org/10.63125/9exrgs41

[173]. Md Shahid, N. (2026a). AI-Driven Quantum-Inspired Predictive Analytics Framework for Procurement Optimization in Critical Infrastructure Networks. *American Journal of Interdisciplinary Studies*, *7*(01), 658-702. https://doi.org/10.63125/hzecta02

[174]. Md Shahid, N. (2026b). Secure Cyber-Physical Data Monitoring Architecture for Real-Time Fault Detection in U.S. Industrial Automation Systems. *American Journal of Data Science and Analytics*, *7*(10), 90-125. https://doi.org/10.63125/exzc2653

[175]. Md Tazul Islam, C. (2024). Explainable AI For Financial and Operational Decision Support: Improving Transparency and Managerial Trust. *American Journal of Advanced Technology and Engineering Solutions*, *14*(12), 01-38. https://doi.org/10.63125/kwfvzw68

[176]. Md Tazul Islam, C. (2025). Business Intelligence and Data-Quality Frameworks for Improving Evidence-Based Decisions across Multi-Source Enterprise Data. *American Journal of Scholarly Research and Innovation*, *15*(07), 01–41. https://doi.org/10.63125/qhh6ka59

[177]. Md Tazul Islam, C., & Ahmed, H. C. H. R. (2023). Predictive Operations Analytics for Resource Allocation and Service-Capacity Planning in Data-Driven Organizations. *Review of Applied Science and Technology*, *13*(02), 01–41. https://doi.org/10.63125/z3sm3089

[178]. Md. Morshedul, I., & Aminul Islam, M. (2022). A Quantitative Evaluation of Explainable Ensemble Learning for Probabilistic Risk Estimation and Decision Reliability in Enterprise Analytics. *American Journal of Data Science and Analytics*, *3*(02), 01-38. https://doi.org/10.63125/k22shk74

[179]. Sadia, A. (2025). Privacy-Aware Personalization: Machine Learning, Model Validation, And Consumer Trust in AI-Mediated Marketing. *American Journal of Scholarly Research and Innovation*, *4*(01), 1081–1119. https://doi.org/10.63125/j7tfr242

[180]. Sadia, A. (2026). A Data-Science Decision-Support Framework for Responsible AI-Enabled Marketing, Visualization, And Consumer Engagement in The United States. *American Journal of Interdisciplinary Studies*, *7*(02), 277-311. https://doi.org/10.63125/e67w9e54

[181]. Sadia, A., & Md. Mosheur, R. (2023). Consumer Trust, Perceived Authenticity, and Personalized Digital Marketing: A Quantitative Decision-Science Study. *International Journal of Scientific Interdisciplinary Research*, *4*(2), 252–295. https://doi.org/10.63125/cajkqt60

[182]. Shabbir, A. (2025). Physics-Based Failure Prognostics for Preventing Pressure-System Degradation and Unplanned Industrial Shutdowns. *American Journal of Scholarly Research and Innovation*, *14*(09), 01–41. https://doi.org/10.63125/3asd0053

[183]. Shabbir, A. (2026). A Data-Science Digital Twin-Enabled Thermal Performance Optimization for Reducing Energy Losses in Industrial Steam Generation Systems. *American Journal of Data Science and Analytics*, *7*(01), 317-363. https://doi.org/10.63125/ybf40824

[184]. Stabak, D., & Md Shahid, N. (2022). Empirical Assessment of Uncertainty-Aware Real-Time Decision-Making for Reliable and Resilient Cyber-Physical Systems. *International Journal of Scientific Interdisciplinary Research*, *1*(01), 432–477. https://doi.org/10.63125/q3r4kk16

[185]. Sumaiya, M. (2024). Hospital Wastewater as A Reservoir of Antimicrobial Resistance Genes: Implications for Public Health and Genomic Surveillance in The United States. *Review of Applied Science and Technology*, *3*(04), 448–508. https://doi.org/10.63125/ahvas976

[186]. Sumaiya, M. (2026). Artificial Intelligence Driven Multi-Omics Approaches for Precision Medicine: From Molecular Discovery to Clinical Translation. *American Journal of Scholarly Research and Innovation*, *5*(01), 429–477. https://doi.org/10.63125/yfme0q12

[187]. Sumaiya Sadmeen, M. (2025). Spatiotemporal Graph Neural Networks and Digital Twins for Urban Drainage Networks: Predictive Inundation, Critical-Node Detection, And Infrastructure Prioritization. *American Journal of Scholarly Research and Innovation*, *15*(12), 01-40. https://doi.org/10.63125/awvjew66

[188]. Sumaiya Sadmeen, M. (2026). Uncertainty-Aware GeoAI Decision Support for Climate-Resilient and Equitable Urban Infrastructure Investment in U.S. Communities. *American Journal of Interdisciplinary Studies*, *16*(05), 42-84. https://doi.org/10.63125/ktww0w31

[189]. Syed Feroz, A., & Md Mahamud Pervaz, P. (2022). Root-Cause Analysis and FMEA-Driven Reliability Improvement in Gas Production Facilities: A Multi-Year Empirical Study. *International Journal of Scientific Interdisciplinary Research*, *12*(10), 01–53. https://doi.org/10.63125/70msp878

[190]. Syed Feroz, A., & Shabbir, A. (2022). Finite Element-Based Thermomechanical Optimization of Pressure Components for Extending Industrial Boiler Service Life. *Journal of Sustainable Development and Policy*, *12*(12), 01-51. https://doi.org/10.63125/x24pht28

[191]. Taher, M. A., Sikder, M. A., Ahmed, H., & Bashar, M. (2024). Integrating IOT and Big Data Analytics for Enhanced Supply Chain Performance in Industrial Engineering Sectors: A Cross-Market Study. *International Journal of Science and Engineering*, *Volume 1*, 1-14.

[192]. Umme, H. (2026). Transformer-Based Multi-Omics Integration for Cancer Drug Response Prediction: A PRISMA 2020 Systematic Review. *American Journal of Health and Medical Sciences*, *16*(01), 33–52. https://doi.org/10.63125/teryrs73