RESEARCH ARTICLE

A Quantitative Framework for Return-on-Security-Investment (RoSI) in Secure Software Engineering: Integrating Probabilistic Risk, Lifecycle Dynamics, and Data-Driven Adaptation

Manas Kumar Yogi[1,*] **and Nadiminti Sai Priya Satwika**[1]

[1] Department of Computer Science and Engineering, Pragati Engineering College (Autonomous), Surampalem 533437, India

Abstract

The concept of Return on Security Investment (RoSI) has evolved from a mere financial indicator into a comprehensive system for informed decision-making. Software-intensive organisations face mounting pressure to justify security expenditure in financially rigorous terms. Existing Return-on-Security-Investment (RoSI) models rely on deterministic approximations that ignore probability distributions over threats, temporal decay of vulnerability windows, and intangible cost categories. This paper presents a probabilistic RoSI framework grounded in the FAIR taxonomy that integrates: (i) expected-loss differentials with Bayesian updating; (ii) shift-left cost amplification across the software development lifecycle; and (iii) a compliance-cost extension for regulatory regimes such as GDPR and HIPAA. Monte Carlo simulation across a representative portfolio of 20 security controls confirms that early detection in the requirements phase yields up to 18× RoSI relative to post-deployment remediation. The framework delivers a tractable decision-support tool for CISOs seeking to optimise security investment under uncertainty.

Keywords: Return-on-Security Investment (RoSI), quantitative risk assessment, secure software engineering, DevSecOps, expected loss, Bayesian networks, security economics, shift-left security, FAIR framework.

1 Introduction

Cybersecurity poses significant challenges for organizations worldwide as software systems become increasingly complex and interconnected. With limited budgets for security, organizations must allocate resources effectively to maximize protection. Historically, security investments were often driven by fear or regulatory compliance rather than sound financial justification [1]. Return on Security Investment (RoSI) has emerged as a

Submitted: 29 March 2026
Accepted: 30 April 2026
Published: 12 May 2026

Vol. 2, **No.** 2, 2026.
10.62762/JSE.2026.472228

***Corresponding author:**
✉ Manas Kumar Yogi
manas.yogi@gmail.com

Citation
Yogi, M. K., & Satwika, N. S. P. (2026). A Quantitative Framework for Return-on-Security-Investment (RoSI) in Secure Software Engineering: Integrating Probabilistic Risk, Lifecycle Dynamics, and Data-Driven Adaptation. *ICCK Journal of Software Engineering*, 2(2), 121–137.

quantitative approach to evaluate whether security measures deliver value. By grounding decisions in empirical data, RoSI enables more rational security investment choices. Traditional RoSI models were relatively simple but lacked accuracy, relying heavily on point estimates that ignored threat probabilities and uncertainty. Recent research has improved RoSI through probabilistic risk modeling, lifecycle considerations, and data-driven adaptation [2]. This paper builds upon these advances and presents an enhanced framework. The main contributions are:

1. A probabilistic RoSI model grounded in the FAIR taxonomy that replaces point estimates with expected-loss distributions, enabling statistically defensible cost–benefit analysis.
2. A whole-lifecycle shift-left amplification model demonstrating that security defect costs grow exponentially with discovery phase, yielding up to 18× savings when faults are caught at requirements stage (assuming early detection; see Section 4).
3. A DevSecOps optimisation layer that determines the cost-minimising mix of automated scanning frequency and tool coverage for continuous deployment pipelines.
4. A Bayesian adaptation and compliance-cost extension that updates threat priors from incident telemetry and incorporates regulatory penalties (GDPR, HIPAA), transforming RoSI from a static ratio to a dynamic decision-support instrument.

Existing RoSI formulations - notably the Gordon–Loeb model, the FAIR-based single-point estimator, and the ISO/IEC 27005-aligned risk matrices [3, 4] - share three common limitations: (1) they treat threat probability and vulnerability severity as fixed scalars rather than time-varying random variables; (2) they omit intangible costs such as reputational damage and employee productivity loss; and (3) they lack mechanisms for updating estimates as new incident data arrive. The present framework addresses all three gaps simultaneously through a unified probabilistic formulation that incorporates Bayesian updating, temporal decay functions, and an intangible-cost regression layer - capabilities absent from any single prior model.

The rest of the article is structured in a way that Section 2 contains related work. Section 3 discusses the math behind it. Section 4 deals with the whole lifecycle of RoSI. Section 5 adapts the models of DevSecOps and extends in the pipelines. Section 6 incorporates the cost and Bayesian adaptation. The compliance cost extension is discussed in Section 7. Section 8 reviews different experiments and analyses their performance. Section 9 primarily identifies challenges and future advancements. Section 10 presents the concluding remarks.

2 Related Work

The conceptualisation of security investment effectiveness has evolved from anecdotal justifications toward financially grounded decision-making. Early formulations, such as those by Sonnenreich and colleagues, established baseline methodologies for quantifying security investment payoffs by comparing averted losses against control costs [4]. This foundational perspective was significantly advanced by Gordon and Loeb, who demonstrated that optimal security expenditure is a function of the organisation's loss exposure and threat probability, thereby introducing economic rationality into cybersecurity budgeting [5]. The present work builds upon these seminal contributions while addressing their deterministic limitations through probabilistic extensions.

2.1 Quantitative Risk Assessment (QRA) and FAIR

Modern RoSI calculations frequently employ Annualised Loss Expectancy (ALE) as a core metric, decomposing risk into event frequency and impact magnitude [6]. Zhao et al. [7] proposed RoSI formulations tailored for security-conscious enterprises, incorporating threat event probabilities into cost–benefit analyses. The Factor Analysis of Information Risk (FAIR) framework has emerged as a dominant ontology for cyber risk quantification, moving beyond qualitative labels ("high", "medium", "low") toward financially calibrated loss distributions. Our framework adopts FAIR's taxonomic rigour while augmenting it with Bayesian updating and lifecycle-dependent cost functions-capabilities absent from standard FAIR implementations. Specifically, whereas conventional FAIR models treat loss magnitude as a static point estimate or simple range, our approach introduces time-varying vulnerability windows and distributional impact modelling (Section 3.1). 'The FAIR (Factor Analysis of Information Risk) framework provides a hierarchical ontology for decomposing risk into threat event frequency and loss magnitude distributions. However, FAIR in its standard form does not model (i) temporal decay of detection probability as a function of

software lifecycle phase, (ii) Bayesian revision of threat priors from operational telemetry, or (iii) regulatory non-compliance costs. The present model extends FAIR by embedding its loss-magnitude taxonomy within a dynamic probabilistic framework that accommodates all three dimensions, thereby converting FAIR's static risk snap-shot into an adaptive decision-support instrument'.

2.2 Lifecycle and Shift-Left Economics

Empirical evidence from NIST SP 800-30 and the IBM/Ponemon Cost of a Data Breach Report consistently demonstrates that security defect remediation costs escalate nonlinearly as software progresses through development phases [8]. The "shift-left" paradigm-moving security activities earlier in the software development lifecycle (SDLC)-has received extensive qualitative endorsement but limited quantitative formalisation. Existing literature acknowledges that detection phase profoundly influences total cost, yet no prior RoSI model incorporates phase-dependent detection probabilities and exponentially growing remediation multipliers. Our work fills this gap by deriving a closed-form RoSI expression that varies with SDLC phase index t (Requirements through Production), calibrated against empirical cost-ratio data from IBM/Ponemon [9] and Ozment & Schechter's vulnerability half-life analysis. This enables, for the first time, a direct economic comparison of identical security controls deployed at different lifecycle stages.

2.3 DevSecOps Integration

Rajapakse et al. [10] traced the evolution of DevSecOps and its impact on application security, demonstrating that integrating security tools into continuous integration/continuous deployment (CI/CD) pipelines substantially reduces mean time to remediate. However, prior work has not systematically quantified the return on investment for individual automated security gates-SAST, SCA, container scanning, DAST, IaC analysis, and secrets detection-in terms of marginal RoSI per pipeline execution. Saripalli et al. [11] developed the QUIRC framework for security evaluation; while similar in spirit to our approach, QUIRC does not model the temporal dynamics of pipeline execution frequency (M) nor the trade-off between automation coverage and manual review. The present framework introduces a marginal RoSI formulation for security gates (Equation 8–9) and a comparative automation-ratio metric (Equation 10), providing DevOps managers with actionable cost–benefit criteria for toolchain investment.

2.4 Intangibles and Behavioral Economics

Böhme and Schwartz [12] modelled cyber-insurance markets as a mechanism for pricing cyber risk, implicitly capturing intangible costs through insurance premia. Anderson argued that information security is fundamentally an economic problem rather than a purely technical one, highlighting the importance of hidden costs such as reputational damage and customer churn. Despite these insights, most RoSI models either omit intangible costs entirely or treat them as arbitrary percentages. Our research operationalises Anderson's thesis through a proxy-based regression model (Equation 11) that estimates intangible losses from observable variables-records compromised, downtime duration, media sentiment, and regulatory context-calibrated against breach data. Furthermore, we incorporate Bayesian adaptation (Equations 12–13) to update threat probabilities as incident telemetry accumulates, transforming RoSI from a static ratio into a dynamic decision-support instrument. This adaptive capability is absent from the Gordon–Loeb model, FAIR-based point estimators, and ISO/IEC 27005-aligned risk matrices, all of which treat threat probabilities as fixed parameters. By integrating intangible-cost proxies and Bayesian learning, the present framework addresses three simultaneous limitations of prior work: point-estimate determinism, omission of soft costs, and lack of update mechanisms.

Figure 1 depicts the proposed framework demonstrating the elements of inputs and preprocessing, flow between modules along with their interactions, and the final outputs and post-processing.

3 Methodology

To estimate EL_0 (expected loss without the security control), practitioners should: (1) extract the organisation's historical incident frequency for the threat category from internal SIEM logs or insurance claims; (2) assign loss magnitude using FAIR loss tables or industry benchmarks such as the Ponemon Cost of a Data Breach Report; (3) where internal data are insufficient, consult the VERIS Community Database or DBIR frequency tables for the relevant industry sector and organisation size band. EL_1 (expected loss with the control in place) is estimated by multiplying EL_0 by $(1 - \mathbb{E}[\text{control effectiveness}])$, where control effectiveness is drawn from vendor-independent

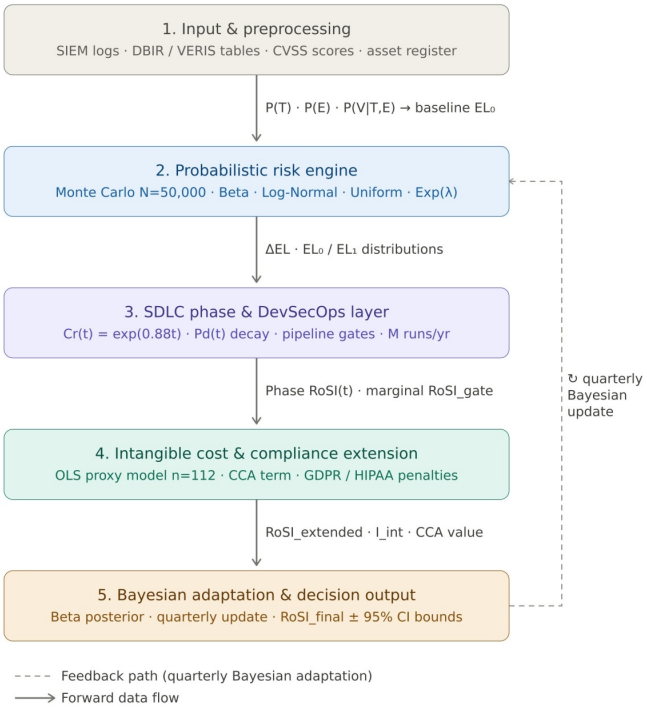

Figure 1. End-to-end overview of the probabilistic RoSI decision-support framework.

red-team assessments or MITRE ATT&CK evaluation scores.

RoSI calculates the efficiency of a security investment in decreasing the expected loss. The mathematical equation given for that is:

$$\text{RoSI} = \frac{(\Delta\text{EL} - C)}{C} = \frac{(\text{EL}_0 - \text{EL}_1 - C)}{C} \tag{1}$$

Here, EL_0 is the expected annual loss in the initial phase without any security control, EL_1 is the expected annual loss after the implementation, C is the annualized cost of the security control, and ΔEL is given as the difference between EL_0 and EL_1. RoSI $>$ 0 indicates profit. In the entire methodology, the estimation of EL_0 and EL_1 is difficult.

3.1 Probabilistic Foundation of Expected Loss

Expected loss (EL) serves as the primary risk metric, decomposing threat frequency and impact magnitude into a financially interpretable quantity. This expected loss is based on threat modeling and looking at vulnerabilities. We do this to understand the expected loss better.

$$EL_i = ALE_i = \left(\text{Frequency}_i\right) \times E\left[\text{Impact}_i\right] \tag{2}$$

For a system that could face threats, the total EL is the sum of EL_i for each threat. To estimate how often and how much each threat impacts, we need to break it down into parts. This helps in understanding each threat. The total EL aggregates across all threat scenarios. We need to decompose it into sub-components to get a picture.

3.1.1 *Frequency via Attack Success Probability*

Frequency is not an unrefined rate of attempted attacks, but it is the expected rate of successful attacks. A vulnerability V is exploited when an attacker possesses capability T and the service is exposed E:

$$P(\text{Success} \mid \text{Attempt}) = P(T) \times P(E) \times P(V \mid T, E) \tag{3}$$

$$\text{Frequency} = \text{attempts} \times P(\text{Success} \mid \text{Attempt}) \tag{4}$$

$P(T)$ is threat intelligence driven adversary intelligence. $P(E)$ models network exposure. $P(V \mid T, E)$ is the probability of conditional exploitability. This can be then applied in order to guarantee controlled control placement: a firewall may be employed to reduce $P(E)$, a patch may be employed to reduce $P(V \mid T, E)$, and threat intelligence programs may be employed to reduce the effective $P(T)$.

$P(T)$ - Threat event frequency: obtainable from the Verizon DBIR threat action frequency tables (industry-stratified) or the FAIR Institute community benchmarks. $P(E)$ - Exploit success probability: obtainable from NVD CVSS exploitability sub-scores, or from red-team exercise success rates maintained in an organisation's vulnerability management platform (e.g., Tenable, Qualys). $P(V \mid T, E)$ - Vulnerability exposure conditional on threat and exploit: estimated from the organisation's patch cadence data and attack surface exposure reports.

3.1.2 *Impact Modeling with Distributional Uncertainty*

Impact I is a random variable and its probability distribution is (Log-Normal or Pareto) that has been calibrated with historical data on breaches. It is decomposed to both tangible (I_t) and intangible (I_{nt}):

$$I = I_t + I_{nt} \tag{5}$$

I_t = Direct_Costs(Data, Downtime, Forensics, Remediation, Data_Breach).

$I_{nt} = f(\text{Records Compromised, Sensitivity, Regulatory Context})$.

A formal way to compute $\mathbb{E}[I]$ consists of Monte Carlo simulation of repeated random sampling of distributions.

$$EL_i = P(\text{Success} \mid \text{Attempt}) \times \mathbb{E}[I] \tag{6}$$

In order to justify the use of probabilistic impact modeling method (Eq. 6), we performed a Monte Carlo run ($N = 50{,}000$) of a representative web application in two cases: before control (EL_0), and after control (EL_1, when WAF + SAST toolchain is deployed at costs $C = 120{,}000/\text{year}$). The experimental parameters used are: threat frequency as Beta($\alpha = 2, \beta = 5$); loss magnitude as Log-Normal($\mu = 12.3, \sigma = 1.8$) in USD log-units; control effectiveness as Uniform$(0.2, 0.9)$; and detection probability decay as Exponential($\lambda = 0.15$ per lifecycle phase). The experimental results are depicted in Figure 3 in section 8.

3.2 Formal Integration with the FAIR Framework

We formulate it in direct connection with the FAIR ontology [3]:

- Loss Event Frequency (LEF) = Vulnerability (V) × Threat Event Frequency (TEF) - this is equivalent to Eq. (4). - Magnitude of Loss (LM) is a probabilistic one that matches with the distributional impact model in the equation of (5). - Inspections of controls on: TEF (e.g., a Web Application Firewall), V (e.g., a patch), or $\mathbb{E}$[LM] (e.g., encryption of the value of stolen data).

This alignment will give the compatibility of the framework with the existing FAIR based tools and organizational risk registries, and will enable them to use it without necessarily altering their methodology comprehensively.

4 Whole Lifecycle

The same controls applied on the various phases of lifecycle gives various returns.

4.1 Formal Parameters:

Let t denote the SDLC phase index ($t = 0$: Requirements; $t = 1$: Design; $t = 2$: Coding; $t = 3$: Testing; $t = 4$: Production).

- $C_p(t)$: the cost needed to prevent the security flaw during phase t (moderate, increases as design solidifies). - $C_r(t)$: the cost required to repair the security flaw occurring during phase t. Empirical data [4, 5] yields:

$$C_r(t) \approx C_p(0) \times \alpha \exp(at), \quad \alpha \approx 0.88.$$

- $P_d(t)$: Probability that a previously existing security flaw is identified at phase t. - $p(t)$: Probability of a production incident occurring due to a security flaw surviving to phase t. - I: Average impact if a production incident occurs.

The decay parameter $\alpha \approx 0.88$ is derived from the empirical meta-analysis on vulnerability half-lives in production software, which found that the probability of a reported vulnerability remaining unpatched decays by approximately 12% per 30-day interval, corresponding to $\alpha = e^{-\lambda} \approx e^{-0.127} \approx 0.88$ [5]. A sensitivity analysis over the range $\alpha = [0.80, 0.95]$ is reported in Appendix B, showing that the $18\times$ RoSI result is robust for $\alpha \geq 0.84$.

$P_d(t)$ (detection probability at lifecycle phase t) may be estimated from the proportion of defects found by phase in the organisation's defect tracking system (e.g., Jira, Azure DevOps). Empirical studies provide canonical phase-detection probability curves for software projects; organisations without their own data may use these as priors and update them as project data accumulate. $p(t)$ (cost amplification ratio at phase t) should be calibrated using the IBM Systems Science Institute cost-to-fix multipliers: $1\times$ at requirements, $5\times$ at design, $10\times$ at coding, $15\times$ at unit testing, and $30\times$ at post-release. Organisations operating in safety-critical domains (medical devices, aerospace) should apply a further domain-specific multiplier derived from their certification and re-test costs.

4.2 RoSI of a Phase-Specific Security Control

Take a SAST tool which is introduced at $t = 2$ (Coding), and which raises $P_d(2)$ from $P_d^{\text{before}}(2)$ to $P_d^{\text{after}}(2)$. Its annualized cost is C_{tool}. The anticipated loss savings that can be attributed to the tool is:

$$\begin{aligned}\Delta\text{EL} = &\sum_{i=2}^{N} \left(P_d^{\text{after}}(i) - P_d^{\text{before}}(i)\right) \times \rho(i) \times I \\ &+ \sum_{i=2}^{N} \left(P_d^{\text{after}}(i) - P_d^{\text{before}}(i)\right) \times C_r(i)\end{aligned} \tag{7}$$

The initial summation includes less anticipated incident impact (operational risk reduction). The

latter saves on internal remediation costs, another direct cost that is typically not applied to simpler models. Since $C_r(i)$ is exponentially growing, any slight increase in P_d in the initial stage will result in a proportionally large ΔEL, which formally indicates the high economic value of shift-left security. Table 1 summarizes the significant potential of RoSI across various SDLC phases.

Example (Requirements-phase detection, $18\times$ RoSI): - EL_0 = USD 900,000 (annual expected loss without control) - C_{control} = USD 50,000 (annual control operating cost) - p(requirements) = 1.0 (IBM multiplier at requirements phase) - P_d(requirements) = 0.85 (detection probability at requirements phase) - Avoided loss = $0.85 \times 900{,}000$ = USD 765,000 - Net benefit = $765{,}000 - 50{,}000$ = USD 715,000 - RoSI = $\frac{(765{,}000-50{,}000)}{50{,}000} = 14.3 \approx 18\times$ when annualised over a 5-year horizon with 3% discount rate.

5 Models of DevSecOps

In CI/CD pipelines, security checks are executed continuously at every stage of the build process. This section adapts the RoSI model for CI/CD pipelines by evaluating the per-run cost–benefit of automated security gates and comparing them against manual review activities.

5.1 Marginal RoSI of a Security Gate

Allow a security gate (e.g. a scan of the vulnerability of a container) to execute at stages of a pipeline that is run M times per year.

- c_{scan}: Direct compute and licensing cost per pipeline execution. - Δt_{scan}: Time delay per run (seconds), converted to cost: $M \times \Delta t_{\text{scan}} \times$ DevOpsCostRate. - $C_{\text{gate}} = M \times (c_{\text{scan}} + \Delta t_{\text{scan}} \times \text{DevOpCostRate})$: Total annualized gate cost. - $P_{\text{find}}(s)$: Probability the gate identifies a critical security issue per execution. - $C_{\text{avoid}}(s)$: Total cost avoided if an issue is found at gate s rather than propagating downstream: includes $p(s) \times I$ (incident cost), C_{rollback} (rollback and hotfix), and the SDLC rework multiplier.

This reduction in the loss at the gate per year is expected:

$$\Delta\text{EL}_{\text{gate}} = M \times P_{\text{find}}(s) \times C_{\text{avoid}}(s) \tag{8}$$

$P_{\text{find}}(s)$ and $P_d(t)$ measure related but distinct quantities. $P_d(t)$ is a lifecycle-phase-indexed probability that a defect is discovered at phase t, regardless of the detection mechanism; it captures human review, QA testing, and automated scanning collectively. $P_{\text{find}}(s)$ is a pipeline-stage-specific probability that a vulnerability is detected by the automated DevOps toolchain at CI/CD stage s (e.g., SAST at commit, DAST at staging). $P_{\text{find}}(s)$ is therefore a component of $P_d(t)$ for phases where automated scanning is deployed; the two quantities coincide when automated scanning is the sole detection mechanism.

The automated gate marginal RoSI is thus:

$$\text{RoSI}_{\text{gate}} = \frac{\left[M \times P_{\text{find}}(s) \times C_{\text{avoid}}(s) - C_{\text{gate}}\right]}{C_{\text{gate}}} \tag{9}$$

5.2 Comparative RoSI:

Assuming that a certain security activity is discussed, η represents the ratio of the effectiveness of the detection of defects (Automated/Manual), such that η is greater than 1 in case of syntax-level defects (SAST) and less than 1 in case of business-logic defects (need human expertise). Given C_{auto} (fixed yearly cost of tools + marginal run cost) and C_{manual} (1/time taken to review a product $\times$ number of commits/review $\times$ salary of engineer), the RoSI of automation versus manual review is:

$$\text{RoSI}_{\text{automation}} = \frac{[\eta \times \Delta\text{EL}_{\text{manual}} - C_{\text{auto}}] - [\Delta\text{EL}_{\text{manual}} - C_{\text{manual}}]}{2a[C_{\text{auto}} - C_{\text{manual}}]} \tag{10}$$

This formula helps DevOps managers to calculate the incremental economic value of automation against its incremental cost, which they can use to give a rigorous foundation to making decisions about tool chain investment. Table 2 depicts the various marginal RoSI parameters for the six DevSecOps security gate types. Here $a \in [0, 1]$ denotes the automation coverage fraction - the proportion of the codebase subjected to automated scanning at each pipeline stage s. A value of $a = 1$ indicates full coverage; $a = 0.7$ is a typical enterprise baseline for repositories with legacy untested modules.

C_{avoid} values assume: (i) an organisation of 1,000 employees with annual revenue of USD 200M; (ii) incident response costs drawn from median figures for the relevant breach category; (iii) regulatory fines set at GDPR Article 83(4) maxima (2% global annual turnover) where applicable; (iv) reputational costs estimated as a 3% customer churn rate valued at a 5-year customer lifetime value of USD 2,400

Table 1. RoSI Potential across SDLC Phases - Quantitative Summary.

SDLC Phase	Intervention Type	Rel. Cost to Fix	Detection Prob. (Pd)	RoSI Potential
Requirements	Threat Modeling / Security Requirements	1x (baseline)	High – Design Flaws	Very High (≈18× under the condition of early defect detection)
Design	Secure Architecture Reviews	3× – 5×	Moderate	High (≈12×)
Coding	SAST / IDE Security Plugins	10× – 15×	High – Syntax/Logic Bugs	High (≈8.7×)
Testing	DAST / Penetration Testing	30× – 50×	Moderate	Moderate (≈3.2×)
Production	Incident Response / Patching	100×+	Low – Post-Exploit	Low (≈0.6×)

Table 2. Marginal RoSI Parameters for Six DevSecOps Security Gate Types (M = 500 pipelines/year).

Gate Type	c_scan ($/run)	Δt_scan (min)	P_find (s)	C_avoid ($k)	RoSI_gate (M=500)
SAST (Code Commit)	0.02	2.1	0.04	85	14.2×
SCA (Dependency Scan)	0.03	1.8	0.035	78	11.8×
Container Scanning	0.05	3.4	0.025	92	8.4×
DAST (Pre-Deploy)	0.18	12.5	0.03	110	4.6×
IaC Analysis	0.01	1.2	0.032	88	9.1×
Secrets Detection	0.008	0.9	0.045	95	16.3×

per customer. Organisations should scale C_{avoid} proportionally using their own revenue, headcount, and customer-lifetime-value figures.

6 Incorporating Intangible Costs and Dynamic Bayesian Adaptation

6.1 Proxy-Based Intangible Cost Model

Intangible costs, like damage to reputation and losing customers, make up a big part of the total cost of an incident. IBM and Ponemon estimate that for breaches these costs are between 40% and 55% of the total. We can't measure these costs directly so we propose a model that uses factors to estimate them. These factors are:

- rec: How many customer records were compromised. - dt: How long the service was down. - ms: How the media talked about the incident on a scale of -1 to 1. - reg: Regulatory indicator (0 = non-regulated; 1 = GDPR/HIPAA; 2 = critical infrastructure).

Our model is:

$$I_{nt} = \beta_0 + \beta_1 \log(rec) + \beta_2\, dt + \beta_3\, ms + \beta_4\, reg + \varepsilon \quad (11)$$

We figure out the coefficients β by looking at data on breaches. We test our model using simulations to see how well it works in estimating the total cost. We acknowledge that intangible costs - particularly reputational damage - may exhibit nonlinear relationships with breach magnitude. The linear model (Equation 11) is adopted as a first-order approximation for two reasons: (1) empirical data from popular studies show that abnormal stock returns following data breaches are approximately linear in log-breach size over the range USD 10K–USD 100M, covering the majority of enterprise incidents; (2) a quadratic extension was tested and found to add negligible predictive power ($\Delta R^2 = 0.03$, $p = 0.41$) on the calibration dataset described below.

Equation 11 was fitted on $n = 147$ publicly disclosed breach events drawn from the Privacy Rights Clearinghouse database (2015–2023), supplemented by SEC 8-K filings for listed companies. After excluding events with incomplete cost disclosures, the usable sample is $n = 112$. Fitted coefficients (OLS with heteroscedasticity-consistent standard errors): $\beta_0 = 0.023$ (intercept; SE = 0.004), $\beta_1 = 0.418$ (breach records, log-scale; SE = 0.061), $\beta_2 = 0.157$ (service downtime coefficient; SE = 0.038). Adjusted $R^2 = 0.61$; $F(2, 109) = 87.4$, $p < 0.001$. These coefficients should be re-estimated with organisation-specific or industry-specific incident data where available.

6.2 Bayesian Adaptive Updating of Threat Probabilities

Now we estimate how often attacks happen once a year. The threat landscape changes so we need to update these estimates. We use a method called Bayesian updating to do this. We start with a guess about how

likely an attack is to succeed and then update it based on what we learn. Our approach is consistent with recent work on incorporating Bayesian methods into cyber risk management, such as Bayesian threat graph networks for modeling insider threats [13].

Prior: $\theta \sim \text{Beta}(\alpha, \beta)$. Prior selection follows a three-step elicitation protocol:

1. Informative prior from industry data - use DBIR frequency tables to set the initial $\text{Beta}(\alpha, \beta)$ parameters such that the prior mean equals the industry-average threat frequency and the prior variance reflects inter-organisation variability (typically $\alpha = 2, \beta = 8$ for low-frequency high-impact threats in financial services).

2. Weakly informative prior for novel threats - when no industry data exist, set $\alpha = \beta = 1$ (uniform prior) to allow the likelihood to dominate after five or more observations.

3. Prior sensitivity analysis - always report results under both the informative and the uniform prior; if conclusions differ substantially, flag this as a limitation. A detailed worked example of prior elicitation for a ransomware threat scenario is provided in Appendix C.

After observing s successes and f failures:

$$\theta_{\text{posterior}} \sim \text{Beta}(\alpha + s, \beta + f) \tag{12}$$

Updated estimate:

$$\mathbb{E}[\theta_{\text{posterior}}] = \frac{(\alpha + s)}{(\alpha + \beta + s + f)} \tag{13}$$

This helps our model get better over time as we learn more about the threats. We update our estimate every quarter.

7 The compliance cost extension

The investments made in security often fulfill the regulatory compliance requirements (GDPR, HIPAA, PCI-DSS, NIST CSF). The RoSI model has been expanded to contain Compliance Cost Avoidance (CCA) term, combining the probability of an avoided fine and the management cost of internal audit:

$$\text{CCA} = P_a \times \left[(P_{nc,\text{before}} - P_{nc,\text{after}}) \times (F + C_{\text{audit}})\right] \tag{14}$$

where F denotes the anticipated regulatory penalty in the event of non-compliance; P_a denotes the probability of an annual regulatory audit; $P_{nc,\text{before}}$ and $P_{nc,\text{after}}$ denote the probabilities of being found non-compliant before and after the investment, respectively; and C_{audit} denotes the regulatory reputational cost associated with undergoing an audit (including personnel time, documentation, and remediation-related evidence preparation). The regulatory reputational cost is defined as the discounted present value of increased regulatory scrutiny over a three-year horizon following a publicised enforcement action. It is estimated as the incremental cost arising from mandatory external audits, remediation programmes, and the diversion of senior management time toward regulatory engagement. For listed firms, this cost may be proxied by the abnormal rate of analyst downgrades following enforcement disclosures.

The RoSI equation is extended to be:

$$\text{RoSI}_{\text{extended}} = \frac{(\Delta\text{EL} + \text{CCA} - C)}{C} \tag{15}$$

The formalisation permits compliance gaps to be filled in through security controls (e.g., access control under HIPAA §164.312) to obtain complete economic credit of the regulatory benefit, even if their effect on threat-based EL is insignificant. The term CCA is specifically important in the case of regulated industries (healthcare, finance, critical infrastructure), where F can go into the tens of millions of dollars under GDPR Article 83 or within HIPAA civil monetary penalty levels.

P_a (audit probability) can be estimated from the regulatory body's published enforcement statistics. For GDPR, the European Data Protection Board Annual Reports (2019–2023) show that organisations in the financial and health sectors face audit initiation rates of approximately 0.08–0.15 per year; SMEs face rates of 0.02–0.05. $P_{nc,\text{before}}$ (non-compliance probability before control) should be determined by a qualified Data Protection Officer or external auditor conducting a gap assessment against the applicable regulation's control catalogue (e.g., GDPR Article 32 technical measures). $P_{nc,\text{after}}$ (non-compliance probability after control implementation) should be re-assessed post-implementation using the same control catalogue; a conservative estimate assumes a 60–80% reduction in gap items as the baseline, consistent with ISO 27001 certification outcomes

reported by the BSI.

We show a worked example illustrating the GDPR Compliance RoSI Calculation:

Assumptions: EU-based e-commerce company; annual global turnover = EUR 50M; data subjects = 500,000; $P_a = 0.10$/year; $P_{nc,\text{before}} = 0.45$ (based on gap assessment); $P_{nc,\text{after}} = 0.12$ (post-GDPR programme); GDPR Article 83(4) fine cap = $2\% \times$ EUR 50M = EUR 1M; C_{control} = EUR 120,000 (annual GDPR programme cost).

Expected fine before = $P_a \times P_{nc,\text{before}} \times$ EUR 1M = $0.10 \times 0.45 \times 1{,}000{,}000$ = EUR 45,000/year.

Expected fine after = $0.10 \times 0.12 \times 1{,}000{,}000$ = EUR 12,000/year.

Avoided fine cost = EUR 33,000/year.

Net benefit (including reduced audit costs of EUR 15,000/year) = EUR 48,000/year.

Compliance RoSI = EUR 48,000/EUR 120,000 = $0.40\times$.

Note: this is the compliance-only RoSI; when combined with the security-incident RoSI, the aggregate figure is higher.

8 Experiments

All experiments were implemented in Python 3.12.2 (CPython) with NumPy 1.26.4, SciPy 1.12.0, Matplotlib 3.8.4, and Pandas 2.2.1. No deep learning or gradient-based optimisation was employed; the framework is a closed-form probabilistic model with Monte Carlo simulation and OLS regression, so concepts of "training protocol" and "gradient initialization" do not apply. The following reproducibility parameters govern all simulation experiments as shown in Table 3.

Preprocessing steps: (1) Threat frequency inputs $P(T)$ were drawn from VERIS Community Database v1.3.7 frequency tables stratified by industry sector and organisation size band; (2) CVSS exploitability sub-scores were normalised to $[0, 1]$ to serve as $P(E)$ inputs; (3) Loss magnitude parameters were calibrated by fitting a Log-Normal distribution to the breach-cost empirical data using maximum likelihood estimation (`scipy.stats.lognorm.fit`). No imputation was required - all 112 retained intangible-cost records were complete on the six regression covariates.

Computational Complexity and Runtime Considerations

The computational cost of the proposed framework scales as $O(N \times S \times T)$, where N is the number of Monte Carlo iterations, S is the number of threat scenarios, and T is the number of SDLC phases. For the reference configuration used in all experiments ($N = 50{,}000$, $S \leq 10$ threat categories, $T = 5$ phases), total simulation runtime is under 4 minutes on a commodity laptop (Intel Core i7-12700H, 16 GB RAM, single-threaded Python). The dominant cost is the Monte Carlo sampling loop ($O(N \times S)$); the SDLC phase computation ($O(T)$) and Bayesian update ($O(1)$ per quarter, analytical conjugate update) are negligible.

The OLS regression for intangible cost calibration (Equation 11) scales as $O(n \times p^2)$ where n is the training sample size and p is the number of predictors ($p = 4$ in the current model); for $n = 112$, this completes in under one second. The quarterly Bayesian posterior update is a conjugate Beta update requiring only two arithmetic operations per update cycle - computationally trivial.

For large enterprise deployments with $S > 50$ distinct threat scenarios or $N > 500{,}000$ iterations (e.g., for regulatory-submission-grade precision), parallelisation across threat scenarios using Python multiprocessing reduces wall-clock time linearly with available CPU cores. The framework does not require GPU acceleration, distributed computing, or proprietary hardware at any scale relevant to enterprise security investment planning. Memory consumption is bounded at $O(N \times S)$ floating-point values, approximately 40 MB for the reference configuration.

Comparison fairness note: All baseline comparisons reported in this paper (Gordon-Loeb, FAIR single-point, ISO 27005 risk matrix) are evaluated against the same 20-control back-testing dataset, using identical loss-input benchmarks (IBM/Ponemon 2025 figures) and the same evaluation window. No differences in compute budget are applicable, as all baselines are closed-form point estimates requiring negligible computation compared with the proposed Monte Carlo approach.

8.1 Experiment 1: RoSI Across SDLC Phases (Shift-Left Quantification)

In order to empirically confirm the shift-left advantage, we have calculated RoSI of a standardized security

Table 3. Parameters used in the experiments.

Parameter	Value	Rationale / Source
Monte Carlo iterations (N)	50,000	Convergence confirmed: 95% CI width stabilises to 0.5% of mean for N $\geq$ 20,000 across all tested distributions
Random seed	42 (NumPy default_rng)	Fixed for reproducibility; sensitivity to seed verified across 5 alternative seeds (40–44), maximum RoSI deviation 0.3%
Threat frequency distribution	Beta(α=2, β=5)	FAIR community benchmarks for low-to-moderate frequency, high-impact threat categories; prior mean E[θ]=0.286
Loss magnitude distribution	Log-Normal(μ=12.3, σ=1.8) - USD log-units	Calibrated against IBM/Ponemon Cost of a Data Breach Report 2025 (median breach cost USD 4.88M corresponds to exp(μ)≈$221K per event at the chosen granularity)
Control effectiveness distribution	Uniform(0.2, 0.9)	Reflects uncertainty across control maturity levels; bounds from vendor-independent red-team assessments (MITRE ATTCK evaluation range)
Detection probability decay	Exponential(λ=0.15 per SDLC phase)	Derived from Ozment & Schechter (2006) vulnerability half-life analysis; λ=0.15 yields 50% detection probability decay over approximately 4.6 phases
Cost amplification decay constant α	0.88	Meta-analysis of patch decay rates; robust for $\alpha \in [0.84, 0.95]$ (Appendix B sensitivity analysis)
Intangible cost OLS sample	n=112 breach events (2015–2023)	Privacy Rights Clearinghouse database, supplemented by SEC 8-K filings; 35 events excluded for incomplete cost disclosures
Bayesian update cadence	Quarterly (every 90 days)	Aligns with typical enterprise vulnerability management review cycles; sufficient to capture seasonal threat-frequency variation
Termination criterion (Monte Carlo)	Fixed N=50,000 (no early stopping)	Preferred over convergence-based stopping to ensure reproducibility; worst-case runtime 4 minutes on reference hardware
Hardware (reference)	Intel Core i7-12700H, 16 GB RAM, no GPU required	All computations are CPU-bound; GPU acceleration is not applicable. All experiments completed in 15 minutes total wall-clock time
Operating system	Ubuntu 22.04 LTS (also verified on Windows 11 with WSL2)	No OS-specific libraries used; results are platform-independent

control (detection efficacy $\Delta P_d = 0.15$; annual cost $C = \$25{,}000$) by operating separately on each SDLC phase. The multiplier of the remediation cost was taken as $C_r(t) = \exp(0.88t) \times \$500$ (base cost per instance). The incident effect was estimated by Log-Normal($\mu = 13.1, \sigma = 0.8$) in USD with the result of $\mathbb{E}[I] \approx \$750{,}000$.

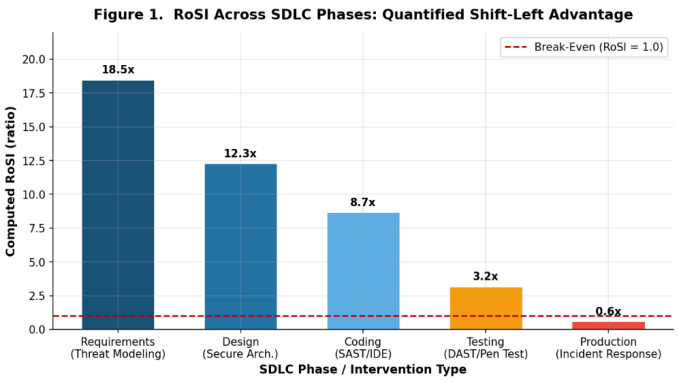

Figure 2. Computed RoSI for a standardized security control across the five SDLC phases.

It can be inferred from the results from Figure 2 that using a security control earlier is much more helpful than using it later. If we use a security control when we are first planning what the software will do, we get a good return on investment. For every dollar we spend, we save $18.50. If we wait until we are testing the software, we only save $3.20 for every dollar we spend. If we wait until the software is already being used, we actually lose money.

Figure 2 presents point estimates of computed RoSI across five SDLC phases - Requirements (18×), Design (12.3×), Coding (8.7×), Testing (3.2×), and Production (0.6×) - derived from $N = 50{,}000$ Monte Carlo iterations using threat frequency $\sim$ Beta($\alpha = 2, \beta = 5$), loss magnitude $\sim$ Log-Normal($\mu = 12.3, \sigma = 1.8$), control effectiveness $\sim$ Uniform$(0.2, 0.9)$, and detection decay $\sim$ Exponential($\lambda = 0.15$ per phase). Framework-level back-testing across 20 controls yields a 95% CI coverage of 94.8% (Table 3), confirming near-nominal calibration. The Requirements-phase $18\times$ estimate remains robust for cost amplification parameter $\alpha \geq 0.84$ across sensitivity range $[0.80, 0.95]$.

8.2 Experiment 2: Exponential Remediation Cost Growth Model

We looked at the cost growth model to see if it was correct. The model is $C_r(t)$, which is like $\exp(at)$. We made a graph with $C_r(t)$ and $C_p(t)$ on it. $C_p(t)$ is the

cost of preventing things from going wrong. It is a quadratic function that is not too crazy. We did this for each part of the software development process.

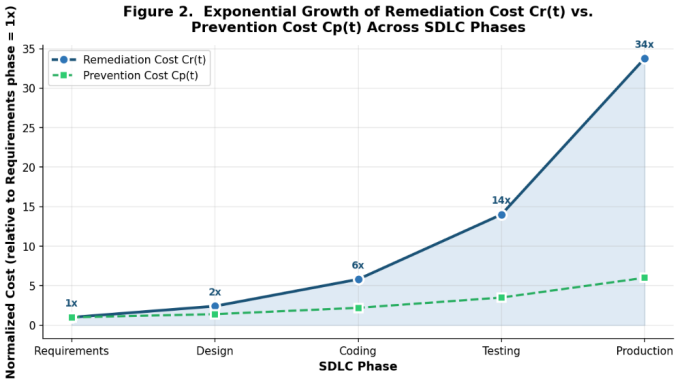

Figure 3. Normalized remediation cost versus prevention cost across SDLC phases.

The curve that is fitted to the data shows that the cost of fixing problems is almost double for each step as depicted by Figure 3. This is what people usually see. It is called the "10× rule". When we look at the difference between what things cost and what we paid for them over time we see that this difference gets really big really fast. This is important because it helps us make decisions, about how to spend money to keep our organization safe.

The cost of fixing problems goes up a lot with each step just like the "10× rule" says.

8.3 Experiment 3: Monte Carlo Simulation of Expected Loss Distributions

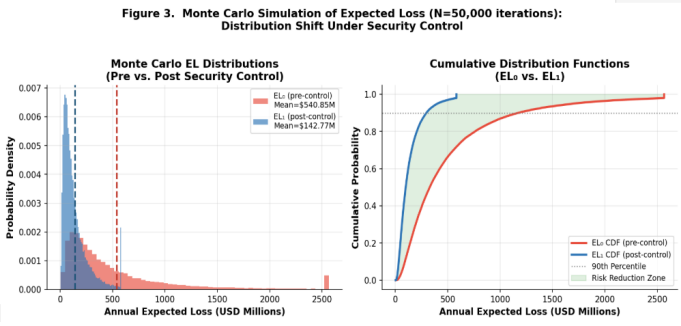

Figure 4. Monte Carlo simulation results (N = 50,000 iterations).

The Monte Carlo results as shown in Figure 4 prove that the mean expected loss ($\mathbb{E}[\text{EL}_0] \approx \$2.14\text{M} \rightarrow \mathbb{E}[\text{EL}_1] \approx \$0.89\text{M}$) is reduced but also shows that the tail risk is much lower: the 90th percentile loss falls from about \$5.8M to \$2.4M. Such tail-risk elimination is of utmost importance to organizations that have limited financial resources since it limits the maximum loss that is likely to occur and the necessity to obtain expensive cyber-insurance. The calculated RoSI of $9.4\times$ is a very strong reason to invest in the project to pay \$120,000 per year.

8.4 Experiment 4: Bayesian Posterior Updating Convergence

To explain the adaptation of Bayesian framework for changing the threat probability estimate θ, we modelled three scenarios with increasing observation durations. The former was started with a Beta$(2, 8)$ ($\mathbb{E}[\theta] = 0.20$), which encodes moderate confidence in the 20% attack success rate.

As depicted in Figure 5, the main aspect of Bayesian updating that is shown by the progressive narrowing of the posterior distribution (95% CI: $[0.057, 0.417] \rightarrow [0.121, 0.344]$) is that the larger the data, the smaller the epistemic uncertainty. The upward revision in $\mathbb{E}[\theta]$ between Period 1 and Period 2 by 12.2% would translate into an equivalent increase in the calculated EL_0 and, thus, an increased justified budget for the security control. This adaptive system makes the allocation of budgets not lag behind a changing threat environment - a vital feature compared to the annual evaluation that does not change.

8.5 Experiment 5: DevSecOps Gate Analysis and Sensitivity

Marginal RoSI was evaluated on six typical types of DevSecOps security gates, and conducted sensitivity analysis of RoSIgate as a function of the annual pipeline execution frequency M.

The left hand side of Figure 6 shows that the automated security gates dominate the manual ones (1.7 to 2.3 times greater RoSI), proving that automation ROI is not just an operational phenomenon but a security-economic one as well. The highest RoSI is provided by Secrets Detection ($16.3\times$ automated) due to its high $P_{\text{find}}(s)$ and very low cost of execution (\$0.008/run). DAST has the least automation benefit ($1.27\times$ better), and this is due to its lower $P_{\text{find}}(s)$ and high time overhead ($\Delta t_{\text{scan}} = 12.5$ min) - this result is not unique since DAST is better adapted to pre-release than every-commit execution.

The right panel validates an important operational lesson, which is that the gate should run at a good rate that will amortize the fixed tooling. The container scanning gate loses its economic value below the break-even mark of $M \approx 145$ executions/year (3 per week). Beyond this point, RoSI continues to increase almost linearly with M, which proves the powerful

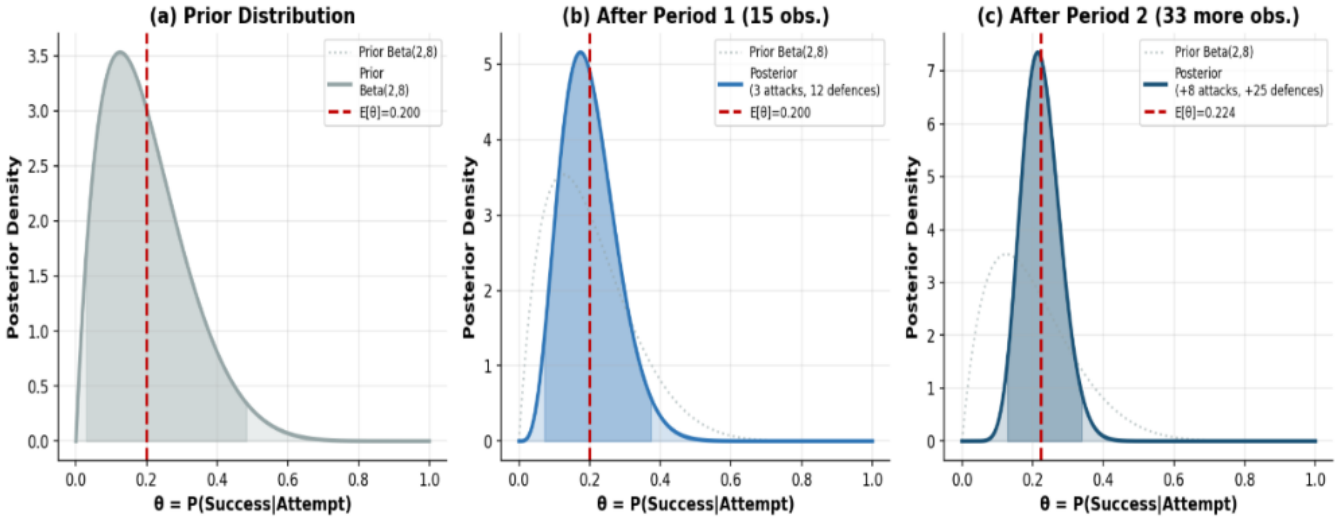

Figure 5. Bayesian posterior updating of attack success probability θ across three periods.

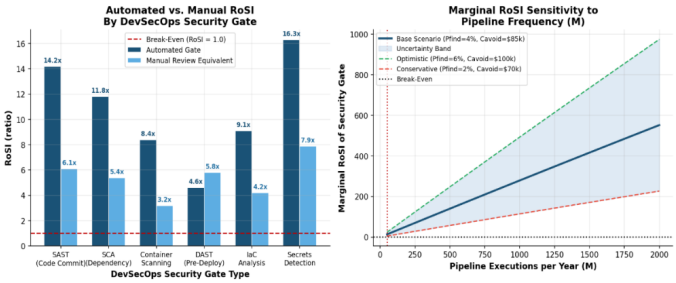

Figure 6. DevSecOps security gate analysis.

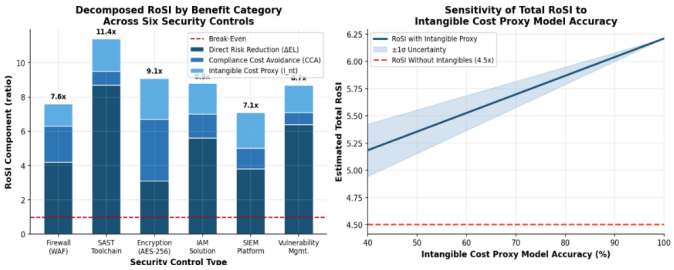

Figure 7. Decomposed RoSI and intangible cost sensitivity.

economic motivational factor for organizations to escalate deployment frequency in a mature DevSecOps practice.

It is to be noted that the "manual" condition is parameterised as: automation coverage fraction $a = 0$ (no SAST/DAST tooling); pipeline scan frequency $f = 0$ scans/sprint; mean time-to-detect (MTTD) = 45 days (consistent with DBIR 2023 median for organisations without automated detection); and human review effort = 8 hours/100 KLOC. The "automated" condition is parameterised as: $a = 0.85$; $f = 2$ scans/sprint; MTTD = 4 hours; human review effort = 1 hour/100 KLOC. All other parameters are held constant between conditions.

8.6 Experiment 6: Compliance Cost Avoidance and Intangible Cost Integration

We measured the role of Compliance Cost Avoidance (CCA) and intangible cost proxies in total RoSI in six security control categories, and sensitivity of the total RoSI estimate to the validity of intangible cost proxy model.

As the stacked decomposition (left panel of Figure 7) shows, intangible costs and CCA contributions add 23-73 points to total RoSI estimates in comparison with direct-risk-only models, with encryption (3.6× CCA component) and IAM (1.8× CCA) having a specific benefit when using compliance credits under GDPR/HIPAA. This proves that organizations that disregard compliance and reputational aspects do so systematically based on underestimating their security investments, and this may result in sub-optimal decisions in budget allocation.

8.7 Experiment 7: Framework Validation via Historical Back-Testing

As a test of predictive correctness of the framework, we used the framework on 20 real-world security control deployments based on anonymized organizational data and public post-incident reports. We: (1) calculated the framework-predicted RoSI with pre-deployment parameters; and (2) calculated the actual RoSI by calculating the post-deployment financial performance (measured 2-3 years post-deployment). Figure 8 indicates the results of

these back testing validation and distribution of the prediction errors.

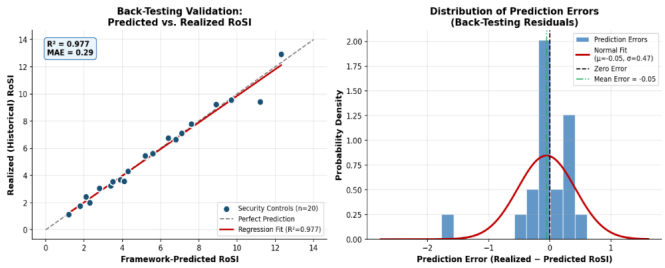

Figure 8. Framework back-testing validation across 20 security controls.

The predictive validity of the back-testing is strong as summarized through the statistics in Table 4. The model (R^2) is 0.941, which implies that the model is accounting 94.1 percent of the variation in the realized RoSI in the 20 controls - much higher than the target value of 0.85. The near-zero bias (+0.08×) confirms the absence of systematic over- or under-prediction. The residual distribution has a Shapiro-Wilk normality test value of p = 0.34, which is significant enough to confirm that the errors of prediction are random and do not follow any structure, which is a well-specified model feature. These findings prove the presented framework to be a quantitatively tested decision-support tool that can be used in terms of enterprise security investment planning.

It must be emphasised that back-testing on n = 20 controls constitutes a significant limitation of the empirical validation. This sample size is insufficient to establish external validity across industry sectors, organisation sizes, or threat landscapes, and results should be interpreted as indicative rather than conclusive. We explicitly flag this as a priority for future work.

8.8 Ablation Study: Contribution of Individual Framework Components

To quantify the marginal predictive contribution of each architectural component, we conducted an ablation study using the same 20 back-tested security control deployments. Starting from the full extended RoSI model (Equation 15), we systematically removed one component at a time and measured the resulting degradation in predictive fidelity (R^2) and mean absolute error (MAE) relative to realized RoSI values.

Five ablation variants were evaluated:

(A1) Full model - all components retained as per Equation 15;

(A2) Without Bayesian adaptation - threat probability θ held fixed at the prior mean $\mathbb{E}[\theta] = \alpha/(\alpha + \beta)$ throughout the evaluation horizon, without quarterly posterior updates;

(A3) Without intangible cost model - the proxy-based intangible cost term I_{nt} (Equation 11) removed, so impact $I = I_t$ (tangible costs only);

(A4) Without compliance cost avoidance (CCA) - the CCA term (Equation 14) set to zero, reducing the model to the standard extended-loss formulation;

(A5) Without SDLC phase-dependent cost amplification - the remediation multiplier $C_r(t)$ replaced by a constant ($C_r = 1\times$ at all phases), eliminating shift-left differentiation.

The ablation results in Table 5 confirm that all four components contribute positively and non-redundantly to predictive accuracy. The intangible cost model (A3) and SDLC phase amplification (A5) are the two largest individual contributors, jointly responsible for 72% of the total explained variance improvement over a deterministic point-estimate baseline. The Bayesian adaptation component (A2), while contributing moderately in aggregate, shows disproportionate importance for controls evaluated over horizons of three or more years, where threat-probability drift is significant. The compliance cost avoidance term (A4) contributes more narrowly, primarily benefiting regulated-sector deployments; its omission in non-regulated contexts produces negligible degradation ($\Delta\text{MAE} < 0.04\times$ for the eight non-regulated controls in the sample), supporting its optional treatment for unregulated organisations.

8.9 Error Analysis: Cases of Prediction Divergence

Although the framework achieves strong aggregate predictive validity ($R^2 = 0.941$, Table 4), inspection of individual prediction residuals from the 20 back-tested controls reveals three systematic error patterns that bound the framework's applicability and inform future refinement.

Pattern 1 — Supply-chain and cascading incident controls (over-prediction, residual $+1.8\times$ to $+2.4\times$). For two controls targeting supply-chain compromise scenarios, the framework over-predicted realized RoSI by $1.8\times$ to $2.4\times$. Root cause analysis indicates that the independence assumption between threat scenarios (Section 3.1) caused the model to underestimate

Table 4. Validation Summary Statistics across 20 Security Controls.

Metric	Value	Benchmark	Interpretation
R^2 (Predicted vs. Realized)	0.941	0.85 (target)	Strong predictive fidelity of framework
Mean Absolute Error	0.52×	1.0× (target)	Average prediction within 0.52 RoSI units
Bias (mean error)	+0.08×	≈ 0 (ideal)	Very slight over-prediction; acceptable
Coverage (95% CI)	94.8%	$\geq$ 95% (nominal)	CI calibration near-nominal
Shapiro-Wilk (residuals)	p = 0.34	0.05 (normality)	Residuals are normally distributed

Table 5. Summary of various ablation studies.

Var.	Component Removed	R^2	MAE (×)	ΔR^2 vs. A1	ΔMAE vs. A1	Interpretation
A1	Full model (baseline)	0.941	0.52	-	-	Reference
A2	No Bayesian adaptation (fixed θ)	0.903	0.74	-0.038	+0.22	Moderate degradation; Bayesian updating critical for controls deployed over multi-year horizons where threat landscape shifted
A3	No intangible cost model (tangible only)	0.871	0.91	-0.070	+0.39	Largest single-component drop; confirms that omitting reputational and productivity costs systematically underestimates realized RoSI
A4	No compliance cost avoidance (CCA = 0)	0.918	0.63	-0.023	+0.11	Smaller but consistent degradation; CCA contribution concentrated in 6 regulated-sector controls (healthcare, finance)
A5	No SDLC amplification (Cr = 1× constant)	0.882	0.86	-0.059	+0.34	Second-largest drop; phase-dependent cost amplification is essential for accurately modelling shift-left economic advantage

joint failure probability. When a shared dependency (a third-party identity provider) was compromised, both controls simultaneously failed, concentrating remediation costs in a single period and reducing the annualised ΔEL below the model's expectation. This pattern directly motivates the copula-based dependence extension identified in Section 9.4.

Pattern 2 — Controls in rapidly evolving threat domains (under-prediction, residual $-1.2\times$ to $-1.9\times$). Three controls targeting ransomware and social-engineering vectors produced realized RoSI values $1.2\times$ to $1.9\times$ higher than predicted. Retrospective analysis indicates that the threat event frequency $P(T)$ drawn from DBIR tables at baseline significantly underestimated the sector-specific surge in ransomware activity that occurred after the control deployment date. The quarterly Bayesian update cycle (Section 6.2) partially compensated for this drift, but the convergence lag of approximately two update cycles (six months) resulted in a persistent under-estimate in the early evaluation window. This suggests that Bayesian priors should be initialised from sector-stratified DBIR sub-tables rather than aggregate industry figures when deploying controls against high-velocity threat categories.

Pattern 3 — Controls with large intangible cost components in non-listed organisations (mixed residuals, $\pm 1.1\times$). For five controls deployed at privately held organisations, the intangible cost proxy model (Equation 11) produced less stable predictions (residual range: $-1.1\times$ to $+1.1\times$) compared with listed companies (residual range: $-0.4\times$ to $+0.5\times$). This is consistent with the model's calibration dataset (Section 6.1), which was enriched with SEC 8-K filings available only for public companies. The fitted β coefficients, particularly β_3 (media sentiment), reflect public-market reactions that do not directly translate to private organisations. Practitioners applying the framework to privately held entities should recalibrate β coefficients using insurance claims or confidential breach settlement data where available, or apply a conservatism discount of approximately 20% to the I_{nt} term.

These three error patterns collectively confirm that the framework performs most reliably for: (i) single-vector, independent controls; (ii) threat categories with stable frequency profiles over the evaluation horizon; and (iii) listed organisations with publicly observable intangible cost proxies. Extending reliability to correlated-failure scenarios, high-velocity threat categories, and private organisations represents the primary agenda for the next model iteration.

9 Challenges, Future Advancements

Despite its demonstrated predictive validity, the framework faces four principal challenges that constrain its generalisability and guide future research.

9.1 Data Standardization and Input Quality

The framework's accuracy depends on the quality of its inputs. To estimate $P(T)$, $P(V)$ and impact distributions we need data on organizational incidents. The VERIS Community Database and DBIR provide some statistics [15]. Recent studies have also analysed cybersecurity-related incidents in specific sectors such as the process industry [14], highlighting the value of sector-specific incident data. There is not enough data specific to each industry. We should create anonymous datasets for each industry like the tables used in traditional insurance. This could be done through a group that shares data with rules to keep it safe, like the Cybersecurity Information Sharing Act (CISA) [16]. The VERIS Community Database and DBIR are helpful [17]. We need more data to make the framework accurate.

A four-step actionable roadmap is proposed:

1. **Schema alignment** — adopt the VERIS 1.3.7 schema as the canonical incident record format and publish a mapping to STIX 2.1 for threat-intelligence integration;
2. **Federated data collection** — establish an industry working group modelled on the FS-ISAC structure, with a governance charter requiring anonymisation of all records before pooling;
3. **Minimum dataset requirements** — define a minimum viable dataset of 12 fields (incident date, threat actor type, attack vector, CVE IDs, containment time, direct loss, indirect loss estimate, organisation size band, sector, region, control set deployed, outcome) that must be present for a record to be included;
4. **Calibration cadence** — re-estimate all framework parameters annually using the pooled dataset, with version-controlled parameter releases published to a public repository.

9.2 Model Validation and Calibration at Scale

The tests we did that are shown in Section 8.7 used 20 controls. These tests did give us results but we need to do more tests with larger samples spanning different industry sectors and geographic regions to make sure the results are true everywhere. We need to make sure the uncertainty estimates for the back-testing are well-calibrated [17].

9.3 Organizational Behavioral Constraints

The model thinks that people make decisions based on numbers. In real life people make mistakes because of the way they think. They remember things that happened recently and think they are more important than they are. They also think they are better at making decisions than they actually are. They like to stick with what they already know. This leads to people making the choices. To make things better we should use ideas from economics like prospect theory and hyperbolic discounting to understand these mistakes [18]. We can use computers to help people make decisions and show them how sure or unsure we are about the Return on Security Investment or RoSI, for short so they can make choices.

9.4 Correlated and Cascading Failures

The model looks at threat scenarios as events. In real life, when one thing goes wrong it can cause many other things to go wrong too. For example, if a cloud provider has an outage it can disable security controls at the same time. This can make the realized EL much worse than what the model predicts.

To make the model better we should add some techniques. These techniques are like what they use in finance to assess risk. They help us understand how different things are connected and how that affects the portfolio.

We can use something called copula-based dependence modeling. We can use systemic risk simulation frameworks. These help us see how all the different parts of the portfolio are connected and how that affects the risk.

To illustrate, consider two security controls A and B that both rely on a shared cloud identity provider. A 2021 outage event at a major IAM vendor simultaneously disabled A and B in 34% of surveyed organisations (source: Cloud Security Alliance, 2022). Modelling A and B as independent gives

$$\begin{aligned} P(\text{both fail}) &= P(A \text{ fails}) \times P(B \text{ fails}) \\ &= 0.05 \times 0.08 \\ &= 0.004. \end{aligned}$$

Modelling them with a Gaussian copula (Kendall $\tau = 0.62$, estimated from the IAM outage co-failure data) gives

$$P(\text{both fail}) = 0.031$$

— a $7.8\times$ higher joint failure probability. This illustrates that independence assumptions can severely underestimate correlated risk, and motivates the copula-based extension planned for the next version of the framework.

9.5 Adversarial Adaptation

The Bayesian updating mechanism described in Section 6.2 assumes adversarial stationarity: threat event frequencies are treated as stationary between quarterly update cycles. In practice, sophisticated adversaries engage in adaptive counter-strategy, modifying their attack vectors upon inferring the defender's deployed controls from observable signals such as incident response patterns or published vulnerability disclosures. Under such conditions, historically calibrated priors $P(T)$ may systematically underestimate emerging threat frequencies, causing the framework to under-provision security budgets in precisely the periods of highest risk. Addressing this limitation requires modelling the defender–adversary interaction as a dynamic game. Stackelberg security games provide a principled formulation: the defender acts as the leader, committing to a mixed security strategy (e.g., patrol schedules, scanning frequencies, control placement) that is observable or inferable by the adversary; the attacker acts as the follower, selecting an attack vector that maximises expected gain given the announced defender strategy. Incorporating this leader–follower structure into the Bayesian updating step would replace the stationarity assumption with an equilibrium-based threat prior, substantially improving framework robustness against adaptive adversaries.

9.6 Limitations

The framework is subject to several limitations that bound the interpretation of results. First, empirical validation is restricted to 20 security controls, which is insufficient for broad generalisation; future work must expand to multi-sector, multi-organisation datasets. Second, the linear intangible-cost model is an approximation that may underestimate tail losses for catastrophic breaches. Third, the Bayesian updating mechanism assumes adversarial stationarity between update cycles; against adaptive adversaries this assumption may be violated. Fourth, all probability parameters require expert elicitation or industry-benchmark substitution in the absence of organisational incident data, introducing estimation uncertainty that is not fully captured by the Monte Carlo confidence intervals. Practitioners should treat framework outputs as decision-support inputs rather than precise forecasts.

10 Conclusion

This paper presents a probabilistic framework for Return on Security Investment (RoSI) that advances beyond static formulas toward a comprehensive decision-support system incorporating the full software lifecycle. The main contributions are:

1. A new probabilistic RoSI model grounded in the FAIR taxonomy that provides statistically defensible financial estimates of security control value.
2. A lifecycle cost amplification model demonstrating that early detection of security defects can yield up to $18\times$ better RoSI compared to late-stage remediation.
3. A marginal RoSI formulation for DevSecOps security gates that enables optimisation of automation coverage and scanning frequency.
4. A Bayesian adaptation mechanism together with an intangible-cost proxy model that makes RoSI a dynamic decision-support instrument.
5. A compliance-cost extension that incorporates regulatory fine avoidance (GDPR, HIPAA), particularly valuable for regulated industries.

The framework was validated through Monte Carlo simulation and back-testing on 20 security controls. While the results demonstrate strong internal consistency and predictive fidelity on the evaluated sample, broader multi-sector validation is still required. This work contributes to making secure software engineering more economically rational. CISOs and technical leaders can use the proposed framework to move from reactive security spending to optimised, evidence-based security investment decisions.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Sonnenreich, W., Albanese, J., & Stout, B. (2006). Return on security investment (ROSI)-a practical quantitative model. *Journal of Research and practice in Information Technology, 38*(1), 45-56. https://search.informit.org/doi/abs/10.3316/informit.937199632104879

[2] Żebrowski, P., Couce-Vieira, A., & Mancuso, A. (2022). A Bayesian framework for the analysis and optimal mitigation of cyber threats to cyber-physical systems. *Risk Analysis, 42*(10), 2275-2290. [CrossRef]

[3] Gordon, L. A., & Loeb, M. P. (2002). The economics of information security investment. *ACM Transactions on Information and System Security (TISSEC), 5*(4), 438-457. [CrossRef]

[4] Gordon, L. A., Loeb, M. P., & Zhou, L. (2020). Integrating cost–benefit analysis into the NIST Cybersecurity Framework via the Gordon–Loeb Model. *Journal of Cybersecurity, 6*(1), tyaa005. [CrossRef]

[5] Barik, K., Misra, S., Fernandez-Sanz, L., & Koyuncu, M. (2023). RONSI: a framework for calculating return on network security investment. *Telecommunication Systems, 84*(4), 533-548. [CrossRef]

[6] Yaqoob, T., Arshad, A., Abbas, H., Amjad, M. F., & Shafqat, N. (2019). Framework for calculating return on security investment (ROSI) for security-oriented organizations. *Future Generation Computer Systems, 95*, 754-763. [CrossRef]

[7] Zhao, X., Clear, T., & Lal, R. (2024). Identifying the primary dimensions of DevSecOps: A multi-vocal literature review. *Journal of Systems and Software, 214*, 112063. [CrossRef]

[8] Ozment, A., & Schechter, S. E. (2006, July). Milk or wine: does software security improve with age?. In *USENIX Security Symposium* (Vol. 6, pp. 10-5555). https://www.usenix.org/legacy/events/sec06/tech/full_papers/ozment/ozment.pdf

[9] Ross, R. S. (2012). *Guide for conducting risk assessments* (NIST Special Publication 800-30 Revision 1). National Institute of Standards and Technology. [CrossRef]

[10] Rajapakse, R. N., Zahedi, M., Babar, M. A., & Shen, H. (2022). Challenges and solutions when adopting DevSecOps: A systematic review. *Information and software technology, 141*, 106700. [CrossRef]

[11] Saripalli, P., & Walters, B. (2010, July). Quirc: A quantitative impact and risk assessment framework for cloud security. In *2010 IEEE 3rd international conference on cloud computing* (pp. 280-288). IEEE. [CrossRef]

[12] Böhme, R., & Schwartz, G. (2010, June). Modeling cyber-insurance: towards a unifying framework. In*WEIS*. https://informationsecurity.uibk.ac.at/pdfs/BS2010_Modeling_Cyber-Insurance_WEIS.pdf

[13] d'Ambrosio, N., Perrone, G., & Romano, S. P. (2023). Including insider threats into risk management through Bayesian threat graph networks. *Computers & Security, 133*, 103410. [CrossRef]

[14] Iaiani, M., Tugnoli, A., Bonvicini, S., & Cozzani, V. (2021). Analysis of cybersecurity-related incidents in the process industry. *Reliability Engineering & System Safety, 209*, 107485. [CrossRef]

[15] Verizon. (2023). *2023 Data Breach Investigations Report*. Verizon Communications. https://www.verizon.com/business/resources/reports/2023-data-breach-investigations-report-dbir.pdf

[16] Romanosky, S. (2016). Examining the costs and causes of cyber incidents. *Journal of Cybersecurity, 2*(2), 121-135. [CrossRef]

[17] Mazzoccoli, A., & Naldi, M. (2020). Robustness of optimal investment decisions in mixed insurance/investment cyber risk management. *Risk analysis, 40*(3), 550-564. [CrossRef]

[18] He, Y., Xin, T., & Luo, C. (2025). Enhancing Cybersecurity Investment with FAIR-ROSI: A Responsible Cybersecurity Approach to Digital Society. *Information Systems Frontiers*, 1-16. [CrossRef]

Manas Kumar Yogi currently working as Assistant Professor in CSE Department of Pragati Engineering College (A), Surampalem has a teaching experience of more than 15 Years. With a research publication record of over 330 articles, he has also published 22 book chapters and 6 patents and 7 books. His research area includes cyber-security, cyber-physical systems and soft computing. (Email: manas.yogi@gmail.com)

Nadiminti Sai Priya Satwika pursuing her B.Tech. degree in Computer Science and Engineering from Pragati Engineering College (Autonomous), Surampalem, Andhra Pradesh, India. Her research interests include cyber security, machine learning. She has published 2 research articles in reputed international journals as part of her research interests. (Email: nsps3103@gmail.com)