pISSN:2349-5707 Volume:12, Issue 10, October-2025 eISSN:2349-5715

INTEGRATED ADAPTIVE DEVSECOPS FOR FINANCIAL AND CRITICAL INFRASTRUCTURE: LEVERAGING EXPLAINABLE AI, THREAT INTELLIGENCE, AND MODEL RISK MANAGEMENT FOR ENHANCED RISK AWARENESS

Vikram C. Reddy
Department of Cyber-Physical Systems and Explainable AI, Delft University of Technology, Netherlands

ABSTRACT: This article proposes a unified, adaptive framework for integrating model risk management, explainable machine learning, and threat intelligence into DevSecOps pipelines for financial services and critical-infrastructure domains. The framework is motivated by the rapid adoption of AI/ML in financial services, the unique model governance challenges that arise from automated decisioning, and the operational pressures of continuous delivery (Financial Stability Board, 2017; Bennett, 2017). We synthesize multicriteria decision methods, model-risk governance principles, and contemporary DevSecOps practices to develop a practical, theoretically grounded process that prioritizes model interpretability, data privacy, and automated pre-deployment risk mitigation (Alimohammadlou & Bonyani, 2017; Holzinger et al., 2018; Díaz et al., 2019). The methodology emphasizes (1) continuous model validation and governance controls that sit within CI/CD and IaC workflows (Bennett, 2017; Crespo et al., 2017); (2) combining MCDM approaches with explainable AI to make trade-offs transparent for stakeholders (De Almeida et al., 2017; Alimohammadlou & Bonyani, 2017); and (3) enriched pipeline automation incorporating threat intelligence to stop high-risk artifacts prior to production deployment (Díaz et al., 2019; Malik, 2025). We present a descriptive results section that outlines expected outcomes, detection and mitigation pathways, and organizational enablers, and we discuss limitations, governance implications, and future research directions. The proposed approach is intended to be actionable for risk and engineering leads in banks, energy markets, and other AI-driven operation environments while also contributing to academic discourse on aligning model risk, cybersecurity, and rapid software delivery lifecycles.

Keywords: Dev SecOps, model risk management, explainable AI, threat intelligence, multicriteria decision making, financial services, continuous validation.

INTRODUCTION

The intersection of accelerating software delivery and the proliferation of machine-learning-driven decisioning creates a set of urgent governance and operational challenges. Organizations increasingly deploy AI/ML models within production systems that directly affect financial outcomes, customer experience, and critical-infrastructure stability (Financial Stability Board, 2017). Simultaneously, modern software development has embraced continuous delivery and deployment practices encapsulated by DevOps and its security-aware evolution, DevSecOps (Forsgren, Humble & Kim, 2018; Myrbakken & Colomo-Palacios, 2017). This convergence exposes two interdependent problem spaces: (1) the management of model risk — the possibility that a model causes material harm due to error, mis-specification, or adversarial manipulation — and (2) the embedding of robust security and privacy practices into the velocity-driven pipelines that deliver model artifacts into production (Bennett, 2017; Crespo et al., 2017).

Model risk management (MRM) in regulated industries such as banking has matured as a discipline focused on governance, validation, and documentation to ensure models behave as intended and that decision makers understand limitations (Bennett, 2017). However, traditional MRM approaches were developed in

slower change environments and often assume lengthy validation cycles that conflict with continuous integration and continuous deployment (CI/CD) rhythms (Crespo et al., 2017). Meanwhile, DevSecOps encourages shift-left security and automated controls, yet many implementations stop short of tightly integrating model governance or advanced AI concerns such as explainability and data privacy (Díaz et al., 2019; Myrbakken & Colomo-Palacios, 2017). The literature therefore demonstrates a gap: governance frameworks for model risk and privacy need practical mechanisms to operate at DevSecOps speed while preserving rigorous validation and interpretability guarantees (Bennett, 2017; Holzinger et al., 2018).

This article addresses that gap by presenting a comprehensive, adaptive framework that fuses multicriteria decision-making (MCDM) techniques, explainable AI (XAI), and operationalized threat intelligence into DevSecOps. We synthesize insights from MCDM research applied to risk contexts (Alimohammadlou & Bonyani, 2017; De Almeida et al., 2017), literature on explainability and interpretability in AI (Holzinger et al., 2018), and operational cybersecurity practices for agile pipelines (Díaz et al., 2019). Additionally, we incorporate contemporary thinking on model risk governance (Bennett, 2017) and the financial-regulatory perspective on AI/ML adoption (Financial Stability Board, 2017). The intended contribution is both practical — offering a staged, instrumented approach for engineering organizations — and conceptual — articulating the trade-offs, governance requirements, and decision criteria that must be reconciled when embedding advanced analytics within rapid-delivery software ecosystems.

The remainder of this document elaborates the theoretical foundations and operational design of the framework, outlines a detailed methodology for instrumenting DevSecOps pipelines with MRM/XAI/threat intelligence controls, presents descriptive results on expected detection and mitigation capabilities, and discusses limitations, governance implications, and future research directions.

METHODOLOGY

This work adopts a synthesis methodology: rather than empirical experimentation within a single organization, we integrate extant theoretical and practitioner literature to design a cohesive, actionable process. The goal is to produce a prescriptive yet adaptable methodology that practitioners can map into their CI/CD pipelines and governance structures. The methodological steps are:

Literature mapping and cross-domain synthesis. We began by mapping themes across three domains: model risk governance and validation (Bennett, 2017; Crespo et al., 2017), multicriteria decision and risk assessment (Alimohammadlou & Bonyani, 2017; De Almeida et al., 2017; Dagoumas, Koltsaklis & Panapakidis, 2017), and DevSecOps and pipeline-integrated security (Díaz et al., 2019; Myrbakken & Colomo-Palacios, 2017; Forsgren, Humble & Kim, 2018). This mapping identified complementary mechanisms (e.g., MCDM for operational trade-offs; XAI for governance explainability) and conflicts (e.g., MRM's lengthy validation cycles vs. CI/CD velocity).

Design objectives formulation. From the mapping we distilled five operational objectives necessary for the integrated framework: 1) Continuous validation of models without all-or-nothing gatekeeping; 2) Embedded interpretability and explainability artifacts packaged with model releases; 3) Risk-aware automated gating informed by threat intelligence and model governance indicators; 4) Traceable data lineage and privacy-preserving practices; and 5) Stakeholder-visible decision dashboards for risk trade-offs (Bennett, 2017; Holzinger et al., 2018; Díaz et al., 2019).

Framework architecture specification. We specified a layered architecture that places MRM and XAI artifacts as first-class citizens in artifacts repositories and CI pipelines. Key components include: model artifact packaging with metadata, automated validation harnesses (unit, integration, and statistical

validation), MCDM scoring engines for multi-stakeholder trade-offs, explainability extractors that produce governance-ready summaries, and an intelligence ingestion layer that feeds threat signals to gating logic (Alimohammadlou & Bonyani, 2017; De Almeida et al., 2017; Díaz et al., 2019).

Operationalization patterns. For each architectural component we propose concrete operational patterns that map to standard DevSecOps practices: pre-commit checks for data quality, CI jobs that incorporate reproducible validation notebooks and XAI reporters, automated pull-request reviews augmented by model-risk scoring, canary deployment patterns tied to monitoring and drift detectors, and automated rollback or quarantine actions triggered by combined model-validation and threat-intel thresholds (Forsgren et al., 2018; Bennett, 2017; Díaz et al., 2019).

Decision and scoring model. We propose adopting an MCDM-inspired scoring approach to synthesize diverse signals into a single gating decision. The scoring takes as input: model performance metrics (accuracy, calibration), explainability metrics (feature importance stability, counterfactual plausibility), model risk indicators (complexity, data sensitivity), operational risk signals (infrastructure vulnerabilities, patch status), and threat intelligence indicators (observed exploit campaigns relevant to model dependencies). We recommend weighting that is configurable by stakeholder role (risk, engineering, compliance) and is updated through governance cycles to reflect changing tolerances (Alimohammadlou & Bonyani, 2017; De Almeida et al., 2017; Bennett, 2017).

Privacy-preserving and regulatory mapping. Data privacy and regulatory compliance must be encoded as non-functional constraints in validation pipelines. We suggest automated checks for dataset provenance, anonymization sufficiency, and policy compliance, complemented by documentation artifacts that feed model inventories for auditability (Cobb, 2016; Bennett, 2017).

Explainability artifact templates. Building on XAI research, we provide templates for the explainability artifacts to be produced during CI: local explanations for high-impact decisions, global model summaries, and sensitivity analyses that explore how model outputs shift with plausible data perturbations. These artifacts must be human-readable and stored alongside model artifacts for auditors (Holzinger et al., 2018).

Threat intelligence integration. We specify an ingestion and mapping layer where external threat feeds (open-source and vendor-intelligence) are mapped to model-relevant indicators (e.g., adversarial model-exploitation campaigns, supply-chain vulnerabilities in ML libraries). These are translated into pipeline-level signals that can raise or lower gating thresholds and trigger specific mitigation playbooks (Díaz et al., 2019; Malik, 2025).

Stakeholder governance and escalation. Finally, we propose a governance loop in which automated pipeline outcomes feed a Model Risk Committee or equivalent decision body. This body periodically reviews weighting schemes, high-severity incidents, and emergent risks to tune the system (Bennett, 2017; Crespo et al., 2017).

Each of these methodological elements is elaborated below with operational detail and theoretical justification.

RESULTS

This section provides a descriptive account of the expected outcomes, capabilities, and behavioral changes when an organization adopts the proposed framework. Because this paper is a synthesis and design contribution, results are framed as predictive and evaluative expectations grounded in the literature rather than as empirical measurements from a single field trial.

Improved alignment between model governance and CI/CD velocity. Traditional MRM practices can cause significant delays when each model change triggers protracted validation cycles. By contrast, embedding automated, reproducible validation and XAI artifact generation within CI allows organizations to preserve governance standards while maintaining deployment velocity. Bennett (2017) emphasized the need for governance structures that adapt to tooling; our framework operationalizes this by transforming manual validation artifacts into machine-verifiable artifacts that can be assessed rapidly during CI jobs.

Reduction in production 'surprises' through continuous validation and monitoring. Continuous validation — unit tests for models, calibration checks, and statistical monitoring for drift — creates early detection pathways for model failures that historically manifest post-deployment. The literature on energy-trade risk modeling and integrated risk systems demonstrates that early detection of anomalies reduces downstream systemic risk (Dagoumas, Koltsaklis & Panapakidis, 2017). In a DevSecOps context, tying drift detectors to canary deployments and automated rollback reduces the blast radius of model regressions (Forsgren et al., 2018).

Transparent trade-offs through MCDM. Multicriteria decision frameworks provide a language for expressing stakeholder trade-offs: for example, a more interpretable but slightly less accurate model may be preferred for high-stakes customer decisions (Alimohammadlou & Bonyani, 2017; De Almeida et al., 2017). By surfacing a weighted score that aggregates performance, interpretability, and risk indicators, teams can make deployment decisions that are defensible to regulators and audit bodies (Bennett, 2017; Holzinger et al., 2018).

Pre-deployment risk mitigation via threat intelligence. Incorporating threat intelligence into gating decisions enables proactive mitigation of supply-chain and model-specific threats before code hits production. Díaz et al. (2019) argue that self-service cybersecurity monitoring empowers teams to act quickly; our approach operationalizes that argument within DevSecOps by mapping threat signals to model artifact risk tiers. For instance, an active exploit targeting a dependency used by a candidate model could automatically alter gating thresholds or require specific mitigation steps.

Improved auditability and regulatory readiness. Automatic generation of explainability artifacts, data lineage records, and model inventories supports regulatory requirements and audit processes. Bennett (2017) underscored the importance of documentation and governance — packaging these artifacts with models makes evidence available on-demand and reduces friction during reviews.

Operationalized privacy controls. By embedding automated checks for dataset provenance and anonymization metrics within CI, organizations can flag potential privacy lapses early. Cobb (2016) describes the complexity of US data privacy regulations; automated enforcement within pipelines reduces human error and the probability of non-compliant releases.

Organizational and cultural shifts. Beyond technical outcomes, implementing the framework tends to produce cultural changes: security and risk teams engage earlier because gate decisions are now transparent and integrated into the same tools developers use (Forsgren et al., 2018; Díaz et al., 2019). This shift-left orientation reduces friction and creates a shared language around risk, performance, and interpretability.

Failure modes and mitigations. Anticipated failure modes include over-reliance on automated scoring (leading to false positives or negatives), gaming of interpretability metrics by modelers, and incomplete threat mapping. The governance loop, featuring periodic recalibration of weights and manual review for high-impact decisions, mitigates these threats (Bennett, 2017; De Almeida et al., 2017).

Case illustration (descriptive). Imagine a retail bank deploying a credit-risk model. The candidate model enters CI where performance tests, calibration checks, and XAI summaries are generated. Threat intelligence flags a vulnerability in a widely used ML library; the scoring engine downgrades the model's risk tier. The pipeline requires a security patch before gating; an engineer patches and re-runs the pipeline, which now passes the combined score. The model is deployed canary-mode for seven days under strict monitoring, and explainability artifacts and lineage data are stored in the model registry for auditors. This sequence demonstrates how the framework reduces exposure while preserving delivery speed (Financial Stability Board, 2017; Díaz et al., 2019).

DISCUSSION

Interpretation of contributions. The core contribution of this paper is a pragmatic architecture and operational methodology that joins model risk, explainability, and threat intelligence within DevSecOps — an alignment that prior literature has identified as necessary but has not fully specified operationally (Bennett, 2017; Myrbakken & Colomo-Palacios, 2017; Díaz et al., 2019). By combining MCDM techniques with automated artifact generation and intelligence-driven gating, the framework balances competing demands: speed, safety, interpretability, and regulatory compliance.

Theoretical implications. The synthesis suggests a reframing of model risk from a purely ex-post governance activity to a continuous engineering property that can be measured and acted upon within CI/CD. This reframing aligns with the broader movement in software engineering that treats non-functional requirements (security, privacy, reliability) as testable attributes in CI (Forsgren et al., 2018). From a decision-science perspective, embedding MCDM into pipelines makes trade-offs computationally tractable and auditable — transforming subjective governance choices into documented, reproducible processes (Alimohammadlou & Bonyani, 2017; De Almeida et al., 2017).

Practical implications for practitioners. Practitioners must consider several concrete implications when adopting the framework. First, tooling and automation investments are required: model registries, CI jobs that execute validation notebooks, explainability tooling, and threat-intel ingestion. Second, governance processes must evolve: Model Risk Committees should accept automated artifacts as partial evidence and define thresholds for manual escalation (Bennett, 2017). Third, organizational roles blurred by DevSecOps (developers, security engineers, data scientists) must be coordinated via shared language and SLAs; self-service monitoring reduces bottlenecks but does not remove the need for expert oversight (Díaz et al., 2019).

Limitations. This work is conceptually rich but empirically limited: it does not present field experiment data from deployment at scale. The proposed MCDM weighting and scoring approach will require context-specific calibration and may be sensitive to stakeholder preferences; mis-weighting can produce undesirable decisions (De Almeida et al., 2017). Threat intelligence mapping is also imperfect — intelligence feeds may contain false positives or lack relevance for specific models; therefore, pipelines must be tuned to avoid unnecessary blocking (Malik, 2025). Explainability techniques vary in maturity and applicability; some model classes (e.g., complex ensembles or deep networks) pose interpretability challenges that require more sophisticated XAI methods and human-in-the-loop review (Holzinger et al., 2018).

Regulatory and privacy considerations. Integrating privacy checks and artifact generation can significantly improve regulatory readiness, but privacy laws (Cobb, 2016) vary across jurisdictions, and automated checks must be aligned with legal interpretations. Moreover, explainability artifacts should be carefully designed to avoid exposing sensitive training data or mechanisms that could enable adversarial exploitation.

Future research directions. Empirical validation across multiple organizations and domains is a priority. Comparative studies can evaluate whether the integrated approach reduces incidents, speeds time-to-deployment, and improves audit outcomes. Additional research should refine MCDM weighting strategies, perhaps using reinforcement learning to adapt weights over time based on governance feedback. Further exploration is needed into adversarial-resilient explainability — ensuring that XAI outputs cannot be exploited to craft attacks — and into quantifying the marginal value of threat intelligence signals in pipeline gating (Holzinger et al., 2018; Malik, 2025).

Counter-arguments and alternative perspectives. Some practitioners may argue that heavy gating and automated risk scoring could slow innovation and introduce rigidities inconsistent with agile principles. This tension highlights an important design trade-off: the goal is not to restore slow, manual validation cycles but rather to create lightweight, targeted checks that are proportional to risk. Other critics may contend that explainability is insufficient for regulatory compliance; while explainability does not replace comprehensive validation, it complements validation by providing interpretable narratives for decisions and is particularly valuable for high-impact models (Bennett, 2017; Holzinger et al., 2018). Finally, skeptics of threat intelligence integration may note the operational overhead of ingesting and mapping signals; this can be mitigated through curated, relevance-filtered feeds and role-based thresholds for gating (Díaz et al., 2019; Malik, 2025).

Governance and organizational change management. Implementing the framework requires change management: aligning incentives across development, security, and risk functions; revising SLAs and incident response playbooks; and investing in training for explainability and model-ops tooling. Importantly, governance structures must commit to periodic review of scoring functions and thresholds so that automated decisions remain aligned with risk appetite (Bennett, 2017). The self-service monitoring approach advocated by Díaz et al. (2019) can reduce gatekeeper bottlenecks, but it requires governance artifacts to be transparent and accessible to a broad set of stakeholders.

Ethical considerations. The framework directly engages with ethical concerns around AI deployment: fairness, transparency, and accountability. Explainability and documented model inventories enable organizations to better evaluate fairness concerns and to respond to consumer inquiries or regulatory investigations. However, technical measures alone do not suffice. Ethical governance must be accompanied by policies that mandate human oversight for sensitive use-cases and that require remediation pathways when unfair or harmful outcomes are detected.

CONCLUSION

This article presented an adaptive, risk-aware DevSecOps framework that integrates model risk management, explainable AI, and threat intelligence to enable responsible, rapid deployment of AI/ML artifacts in financial and critical-infrastructure settings. Grounded in literature from model governance, multicriteria decision-making, and pipeline security, the proposed approach operationalizes governance through automated validation, explainability artifact generation, MCDM-based scoring, and intelligence-informed gating. The principal contribution is a practical architecture and operational patterns that reconcile the tension between regulatory-quality model governance and the velocity of modern software delivery.

While the framework requires investment in tooling and organizational change, its expected benefits include improved detection of model failures, better regulatory readiness, proactive mitigation of intelligence-identified threats, and clearer articulation of trade-offs between performance and interpretability. Limitations include the need for empirical validation, the challenge of calibrating MCDM weightings, and the imperfect nature of threat intelligence feeds. Future work should empirically evaluate

the framework across domains, explore automated adaptation of scoring weights, and further develop adversarially robust explainability techniques.

Deploying AI responsibly in high-stakes domains demands that engineering, security, and risk governance converge on shared processes and artifacts. This paper outlines a roadmap for that convergence, offering operational mechanisms that make model risk visible, explainable, and actionable within the continuous delivery lifecycles that define modern software and systems engineering.

REFERENCES

1. Alimohammadlou, M., & Bonyani, A. (2017). A novel hybrid MCDM model for financial performance evaluation in Iran's food industry. Accounting and Financial Control, 1(2), 38–45. https://doi.org/10.21511/afc.01(2).2017.05

2. Bennett, D. E. (2017). Governance and organizational requirements for effective model risk management. Journal of Risk Model Validation, 11(4), 97–116. https://doi.org/10.21314/JRMV.2017.188

3. Cobb, S. (2016). Data privacy and data protection: US law and legislation. Eset, (April), 1–16. Retrieved from https://www.welivesecurity.com/wp-content/uploads/2018/01/US-data-privacy-legislation-whitepaper.pdf

4. Crespo, I., Kumar, P., & Noteboom, P. (2017). The evolution of model risk management. McKinsey Global Institute, 1–8.

5. Dagoumas, A. S., Koltsaklis, N. E., & Panapakidis, I. P. (2017). An integrated model for risk management in electricity trade. Energy, 124, 350–363. https://doi.org/10.1016/j.energy.2017.02.064

6. Dash, S. (2018). An Efficient AI Model for Financial Market Prediction Optimized by SVR. International Journal for Research in Applied Science and Engineering Technology, 6(5), 1884–1889. https://doi.org/10.22214/ijraset.2018.5307

7. De Almeida, A. T., Alencar, M. H., Garcez, T. V., & Ferreira, R. J. P. (2017, April 1). A systematic literature review of multicriteria and multi-objective models applied in risk management. IMA Journal of Management Mathematics. Oxford University Press. https://doi.org/10.1093/imaman/dpw021

8. Díaz, J., Pérez, J. E., Lopez-Peña, M. A., Mena, G. A., & Yagüe, A. (2019). Self-service cybersecurity monitoring as enabler for DevSecops. IEEE Access, 7, 100283–100295. https://doi.org/10.1109/ACCESS.2019.2930000

9. Financial Stability Board. (2017). Artificial Intelligence and Machine Learning in Financial Services - Market Developments and Financial Stability Implications. Financial Stability Board, (November), 45. Retrieved from http://www.fsb.org/2017/11/artificial-intelligence-and-machine-learning-in-financialservice/

10. Holzinger, A., Kieseberg, P., Weippl, E., & Tjoa, A. M. (2018). Current advances, trends and challenges of machine learning and knowledge extraction: From machine learning to explainable AI. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 11015 LNCS, pp. 1–8). Springer Verlag. https://doi.org/10.1007/978-3-319-99740-7_1

11. Myrbakken, G., & Colomo-Palacios, R. (2017). DevSecOps: A Multivocal Literature Review. Proceedings of the 18th International Conference on Product-Focused Software Process Improvement, 17–29.

12. Forsgren, N., Humble, J., & Kim, G. (2018). Accelerate: State of DevOps. DevOps Research and Assessment (DORA).

13. Debois, P. (2011). DevOps: A Software Revolution in the Making. Cutter IT Journal, 24(8).

14. Allspaw, J., & Hammond, P. (2009). 10+ Deploys per Day: Dev and Ops Cooperation at Flickr. Velocity Conference.

15. OWASP Foundation. OWASP Top Ten. https://owasp.org/www-project-top-ten/

16. Kim, G., Humble, J., Debois, P., & Willis, J. (2016). The DevOps Handbook. IT Revolution Press.

17. Zaydi, A., & Bouchaib, H. (2020). From DevOps to DevSecOps: The Role of Security and Compliance in ITSM. International Journal of Information Systems Engineering (IJISE), 8(2).

18. Malik, G. (2025). Integrating Threat Intelligence with DevSecOps: Automating Risk Mitigation before Code Hits Production. Utilitas Mathematica, 122(2), 309-340.