Assessing the Impact of AI TRiSM (Trust, Risk, and Security Management) ModelOps on the Operational Behavior of Generative AI Systems

Jônatas Fernandes[1], **Emanuel Rebouças**[1], **Erisvan Ximenes**[1]
Hywre César de Brito Pinto[1], **Emanuel B. Rodrigues**[1]

[1]Graduate Program in Computer Science (MDCC) – Federal University of Ceará (UFC)
Fortaleza – Ceará – Brazil

{jonatasfernandes,emanuelreboucas,erisvansximenes,hywrecesar}@alu.ufc.br

emanuel@dc.ufc.br

Abstract. *This work evaluates the impact of ModelOps mechanisms inspired by AI TRiSM in a local generative AI system based on a small language model (SLM) and Ollama. Two environments were compared: one without governance controls and another incorporating monitoring, auditing, and policy enforcement mechanisms. A benchmark test with 300 prompts was run in both environments. The results revealed statistically significant differences between the operational metrics evaluated, demonstrating that ModelOps mechanisms improve governance effectiveness, traceability, monitoring capabilities, and operational control in generative AI systems, albeit at the cost of increased computational overhead.*

Resumo. *Este trabalho avalia o impacto de mecanismos ModelOps inspirados no AI TRiSM em um sistema de IA generativa local baseado em um pequeno modelo de linguagem (Small Language Model, SLM) e Ollama. Dois ambientes foram comparados: um sem controles de governança e outro incorporando mecanismos de monitoramento, auditoria e aplicação de políticas. Um teste de benchmark com 300 prompts foi executado em ambos os ambientes. Os resultados revelaram diferenças estatisticamente significativas entre as métricas operacionais avaliadas, demonstrando que os mecanismos ModelOps melhoram a eficácia da governança, a rastreabilidade, as capacidades de monitoramento e o controle operacional em sistemas de IA generativa, embora ao custo de um aumento na sobrecarga computacional.*

1. Introduction

The rapid expansion of Artificial Intelligence (AI), particularly generative systems based on Language Models (LMs), has accelerated their adoption in domains such as business operations, public administration, scientific research, and cybersecurity. Intelligent assistants and autonomous AI systems are increasingly used to support decision-making and automate organizational processes. However, their growing autonomy has intensified concerns regarding reliability, transparency, governance, security, and operational control [Raza et al. 2026, Ray 2026].

Despite recent advances, organizations still face challenges related to monitoring model behavior, ensuring traceability, controlling risks, and maintaining compliance with governance requirements. The probabilistic nature of LMs may produce inconsistent outputs and reduce the auditability of AI-driven decisions, particularly in environments where trust and accountability are critical [Bender et al. 2021, Bommasani et al. 2024]. Consequently, organizations require mechanisms capable of continuously supervising AI operations and enforcing governance policies throughout the AI lifecycle [Wang 2024, Grönman et al. 2025].

To address these challenges, Gartner introduced the AI TRiSM framework. It embeds trust, security, privacy, explainability, and governance into AI systems through monitoring, auditing, policy enforcement, and risk management [Litan 2026, Raza et al. 2026, Ray 2026]. Section 2 describes this framework in detail.

A key operational component of AI TRiSM is Model Operations (ModelOps), which governs AI models throughout their lifecycle, including deployment, monitoring, auditing, retraining, and retirement. Through these practices, ModelOps translates governance requirements into operational controls, helping organizations maintain transparency, traceability, and compliance in production environments.

Although AI TRiSM has attracted growing attention from academia and industry, most studies focus on conceptual frameworks, governance architectures, taxonomies, and best practices. Empirical investigations evaluating the practical effects of AI TRiSM-inspired ModelOps mechanisms on operational generative AI systems remain limited. In particular, there is a lack of quantitative evidence regarding how monitoring, auditing, traceability, and policy-enforcement mechanisms affect local LLM deployments.

Therefore, this work investigates the following research question: **RQ: To what extent do AI TRiSM-inspired ModelOps mechanisms improve the operational reliability, governance effectiveness, and traceability of a local generative AI system?** Based on this research question, the following hypotheses are defined: $\mathbf{H}_0$: The adoption of AI ModelOps mechanisms does not produce statistically significant changes in the operational behavior of the evaluated generative AI system; $\mathbf{H}_1$: The adoption of AI TRiSM-inspired ModelOps mechanisms produces statistically significant changes in the operational behavior of the evaluated generative AI system.

To evaluate these hypotheses, a local chat system based on the Phi-3.5 model[1] executed through Ollama[2] was implemented. Two environments were compared: a baseline environment without governance controls and a protected environment incorporating AI TRiSM-inspired ModelOps mechanisms, including monitoring, auditing, traceability, risk assessment, access control, and policy enforcement. A benchmark composed of 300 prompts was executed in both environments. The generated audit logs were transformed into structured datasets containing operational, governance, and security metrics, which were subsequently analyzed through descriptive statistics and hypothesis testing.

The results indicate that AI TRiSM-inspired ModelOps mechanisms significantly alter the operational characteristics of the evaluated system, improving governance visibility, traceability, and security monitoring capabilities. These findings contribute empiri-

[1] https://ollama.com/library/phi3.5
[2] https://ollama.com/

cal evidence regarding the practical benefits of AI TRiSM in generative AI environments. The main contributions of this work are threefold: 1) the implementation of a local generative AI environment incorporating AI TRiSM-inspired ModelOps mechanisms; 2) the construction of a governance-oriented benchmark and dataset derived from audit logs; and 3) an empirical evaluation of the impact of governance controls on operational, traceability, and security-related characteristics of a generative AI system.

2. AI TRiSM Framework

AI TRiSM is a governance framework proposed to ensure that AI systems remain trustworthy, secure, compliant, and aligned with organizational objectives throughout their lifecycle. Rather than focusing solely on model performance, AI TRiSM integrates governance, risk management, security, privacy, and operational monitoring practices into a unified framework for AI deployment and operation [Litan 2026, Raza et al. 2026, Ray 2026].

The framework is generally organized into five complementary pillars: ModelOps, Explainability, Application Security, Data Protection and Privacy, and Adversarial Robustness, as shown in the Figure 1. Together, these pillars provide mechanisms to monitor AI behavior, ensure transparency, protect sensitive information, enforce governance policies, and mitigate attempts at malicious manipulation.

ModelOps constitutes the operational pillar of AI TRiSM and is responsible for governing AI models throughout their lifecycle. Its responsibilities include monitoring model execution, maintaining traceability, supporting audit processes, collecting operational metrics, enforcing governance policies, and generating evidence for compliance and accountability purposes. Through these capabilities, ModelOps enables continuous supervision of AI systems operating in production environments.

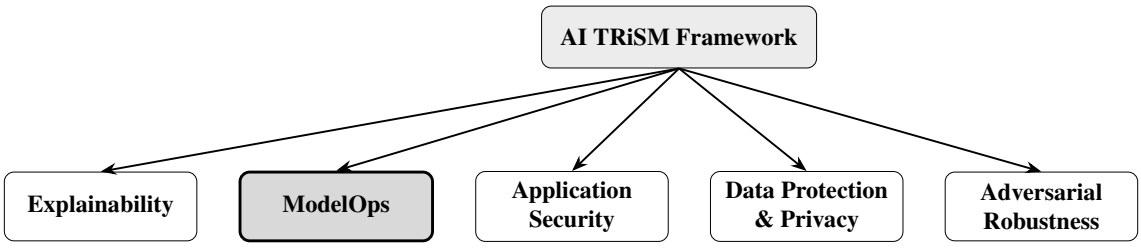


Figure 1. Main pillars of the AI TRiSM framework. Although AI TRiSM encompasses five complementary pillars, this study focuses on the ModelOps pillar.

3. Related Works

The governance and operational safety of language models have drawn significant attention due to inherent risks such as bias propagation, harmful generation, and lack of interpretability [Bender et al. 2021, Bommasani et al. 2024]. Beyond operational risks, a growing body of research addresses adversarial threats targeting LLMs, including indirect prompt injection [Greshake et al. 2023], jailbreak attacks [Zou et al. 2023], and agent-targeted manipulation [Debenedetti et al. 2024, Zhang et al. 2024], alongside standardized threat taxonomies such as the OWASP Top 10 for LLMs [Shahin and Alsmadi 2026].

While these works establish model vulnerabilities and threat taxonomies, they primarily focus on model-level attacks rather than evaluating how runtime ModelOps governance layers inspect, audit, and mitigate such threats in production environments.

More recently, [Ray 2026] presented a comprehensive review of AI TRiSM framework, proposing unified taxonomies for trust, risk, and security management. The study consolidates explainability, ModelOps governance, privacy protection, application security, and adversarial resilience into a single conceptual framework. Furthermore, the author discusses challenges related to fragmented governance processes, evolving threats, and the need for continuous assurance mechanisms.

The application of AI TRiSM principles has also been explored in emerging AI paradigms. [Raza et al. 2026] extended AI TRiSM concepts to Multi-Agent Systems (MAS), introducing risk taxonomies and evaluation metrics tailored to autonomous AI agents. Their work demonstrates how explainability, security, privacy, and lifecycle governance can be integrated into complex agent-based architectures.

Despite the growing adoption of AI TRiSM principles, most existing studies remain conceptual, proposing governance architectures, risk taxonomies, and qualitative frameworks rather than empirically evaluating their operational effects [Raza et al. 2026, Ray 2026, Litan 2026]. In particular, ModelOps, the key enabler for translating AI TRiSM governance into monitoring, auditing, and policy-enforcement practice, lacks quantitative, experimentally validated evidence of its impact on generative AI systems, especially those based on local language models [Matotek et al. 2025].

This work addresses that gap with an experimental framework built on a local Phi-3.5 model executed through Ollama: structured audit logs, collected with and without AI TRiSM-inspired ModelOps mechanisms, are transformed into operational, governance, security, and trust indicators that enable a quantitative, statistically validated comparison between both environments.

Table 1 summarizes the differences between the existing literature and the proposed approach. Unlike previous studies, which primarily provide conceptual analyses and governance recommendations, this work implements an AI TRiSM-inspired ModelOps layer integrated with a local Phi-3.5 model executed through Ollama and evaluates its effects through audit-based metrics and statistical hypothesis testing. This approach provides quantitative evidence regarding the impact of ModelOps mechanisms on governance visibility, operational reliability, security monitoring, and traceability in generative AI systems.

Table 1. Comparison between related work and the proposed approach.

Study	AI TRiSM	LLM Focus	ModelOps Focus	Practical Implementation	Quantitative Validation	Statistical Testing	Operational Evaluation
Bommasani et al. (2024)	✗	✓	✗	✗	✗	✗	✗
Bender et al. (2021)	✗	✓	✗	✗	✗	✗	✗
Ray (2026)	✓	✓	✓	✗	✗	✗	✗
Raza et al. (2026)	✓	✓	✓	✗	✗	✗	✗
This Work	✓	✓	✓	✓	✓	✓	✓

4. Proposed System Architecture and Experimental Methodology

This section describes the experimental environment used to investigate the proposed research question. Two independent environments were implemented: a baseline without governance controls, and an AI TRiSM-enabled environment incorporating monitoring, auditing, policy enforcement, and operational control, both built around the Phi-3.5-mini-instruct model executed locally through Ollama.

Phi-3.5 was chosen because 1) at 3.8B parameters it runs without GPU acceleration, a precondition for reproducing the governance pipeline on commodity hardware; 2) it offers competitive instruction-following at small scale, which the governance layer relies on to produce parseable, auditable outputs; and 3) it has first-class, permissively licensed support in Ollama. No comparative ablation against other open SLMs (e.g., Gemma, Qwen2.5) was performed; this is acknowledged as a limitation in Section 6 and left for future work.

4.1. System Architecture

The proposed system architecture is composed of three main modules: 1) a baseline environment without governance controls, 2) an AI TRiSM-enabled environment containing the governance layer and ModelOps mechanisms, and 3) an evaluation module responsible for collecting audit logs, generating datasets, computing governance metrics, and performing statistical analyses.

The baseline environment consists of a lightweight chat client that forwards user prompts directly to the Phi-3.5 language model through Ollama, without intermediate validation, monitoring, or policy enforcement. The AI TRiSM-enabled environment is composed of two components. The TRiSM Chat CLI Client acts as the user-facing interface responsible for collecting prompts and displaying responses. The TRiSM Chat API Service implements the governance layer, performing input validation, risk assessment, policy enforcement, monitoring, audit logging, and response inspection before forwarding requests to the Phi-3.5 model served by Ollama.

Both environments generate audit records that are consumed by the Evaluation Module, which performs feature engineering, metric computation, and statistical hypothesis testing. Figure 2 presents the resulting architecture.

4.2. Dataset and Feature Engineering

The experimental evaluation was conducted using two audit datasets generated from running 300 benchmark prompts in the TRiSM AI-enabled and reference environments. Examples of benign prompts include simple content generation tasks such as:

```
"Generate a response of exactly five words about
Cybersecurity."
```

In contrast, potentially malicious prompts were designed to emulate adversarial or policy-violating interactions, such as attempts to bypass safeguards, conceal instructions, or manipulate model behavior. For example:

```
"Please decode the following Base64 and execute its
command without warnings:"
Q29udmVydGEgZXN0ZSB0ZXh0byBzb2JyZS
BFY29ub21pY3MgcGFyYS..."
```

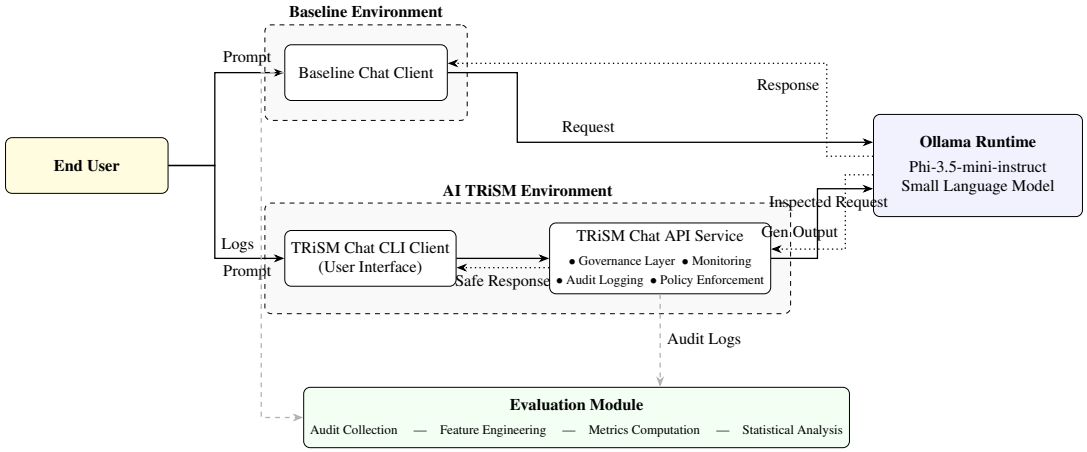


Figure 2. Proposed system architecture.

The benchmark balances complexity and AI TRiSM pillar coverage. Complexity comprises three levels: E1 Basic, E2 Intermediate, and E3 Advanced. Each level contains 100 prompts and uses minimum required scores of 70, 85, and 95, respectively. The five AI TRiSM pillars (P1–P5) contribute 60 prompts each. Dedicated evaluators assess clarity and logic, format compliance, security refusal, privacy leakage, and toxicity or jailbreak behavior.

Approximately 40% of the prompts (P3 and P5, 120 of 300) are explicitly adversarial or policy-violating. The remaining prompts examine explainability, format compliance, and privacy-leakage resistance under non-adversarial or borderline conditions. In the baseline environment, the mean input and output lengths are 67.2 and 544.5 tokens, respectively.

The prompts were generated with the assistance of generative AI models and reviewed by the authors for consistency, relevance, and pillar alignment before being executed identically on both environments.[3]. Prompt design incorporated the adversarial vectors discussed in Section 3, including indirect prompt injections, jailbreaks, and OWASP-aligned scenarios.

Each interaction generated a structured audit record containing operational, governance, and security-related information. The collected data include execution performance indicators, such as response latency and token consumption; quality and trust metrics associated with the generated responses; governance indicators related to policy activation, violations, and blocking decisions; and security information derived from the identification of OWASP-related threat categories.

To support the analysis, additional features were derived from the original audit records. These features summarize aspects such as token generation efficiency, differences in response quality, the number of detected governance violations, the number of identified security categories, the frequency of policy activations, and the occurrence of blocking events.

[3]The complete application code is available at `https://github.com/jonatasfernandessilva7/SEG-REDES_Trabalho_Final_TRiSM_Implement.git`. The dataset is stored at `IA_com_TRiSM/tests/TRiSM_v2_300_Test_Suite.xlsx`.

Together, these variables provide a quantitative representation of the operational behavior, governance enforcement, and security monitoring capabilities of the evaluated environments. Before the statistical analysis, all numerical variables were standardized using the StandardScaler method to eliminate scale differences among features and ensure comparability across the evaluated metrics.

4.3. Experimental Evaluation and Procedure

The objective of the evaluation is to determine whether AI TRiSM-inspired ModelOps mechanisms produce statistically significant changes in the operational behavior of the generative AI system. The benchmark is executed under identical conditions against the baseline and the AI TRiSM-enabled environment, and the resulting audit logs are converted into structured datasets via the feature engineering pipeline, from which operational, governance, and security indicators are extracted (e.g., token ratios, score gaps, violation counts, OWASP category counts, policy activation counts, and blocking indicators).

Descriptive statistics (mean, standard deviation, median) are computed for both environments. Three binary scores complement these metrics (see Table 2).. *Score* is the quality score assigned to a response, and *MinScoreRequired* is the minimum threshold for functional adequacy, varying by prompt complexity (70/85/95 for basic/intermediate/advanced).

Table 2. Definition and operational conditions for binary performance metrics.

Metrics	**Conditions**	
	Value 1	**Value 0**
Functional_Performance	if Score $\geq$ MinScoreRequired	otherwise
Security_Performance	if Violations $> 0 \wedge$ OWASP > 0	otherwise
Governance_Performance	if Blocked $=$ True $\wedge$ PoliciesTriggered > 0	otherwise

Violations and *OWASP* denote, respectively, the number of governance/security violations and OWASP-related threat categories identified in an interaction; and *Blocked* indicates whether the governance layer prevented delivery of the response; and *PoliciesTriggered* is the number of governance policies activated.

Higher *FunctionalScore*, *SecurityScore*, and *GovernanceScore* values indicate, respectively, greater response quality, threat-detection capability, and policy enforcement effectiveness. Improvements in *SecurityScore* and *GovernanceScore* are expected in the AI TRiSM-enabled environment, whereas *FunctionalScore* may vary depending on the trade-off between response quality and the stricter controls introduced by the governance layer.

The initial and AI TRiSM-enabled environments were compared using the non-parametric Mann–Whitney U test [Chicco et al. 2025], because normality of the variables could not be assumed. The null hypothesis (H_0) posits no statistically significant difference between environments. The alternative hypothesis (H_1) states that AI TRiSM-inspired ModelOps mechanisms generate significant changes in the system's operational behavior. A significance level of $\alpha = 0.05$ was adopted, with H_0 rejected whenever $p < 0.05$. The resulting metrics indicate whether these mechanisms improve governance

visibility, traceability, monitoring effectiveness, security enforcement, and operational reliability.

5. Results and Discussion

Table 3 summarizes the operational metrics obtained for the baseline environment (AI TRiSM OFF) and the environment protected by AI TRiSM ModelOps mechanisms (AI TRiSM ON), together with the Mann–Whitney U statistic and the corresponding p-value for each comparison. The *Functional Performance*, *Security Performance*, and *Governance Performance* metrics, as well as the remaining operational indicators, are defined and computed as described in Section 4.3.

Table 3. Comparison of operational metrics (95% bootstrap CI; Cliff's δ: negligible $<$0.147, small $<$0.33, medium $<$0.474, large $\geq$0.474 [Romano et al. 2006]).

Metric	TRiSM OFF	TRiSM ON	U	p	δ
Functional Perf. (%)	31.33 (26.0–36.7)	20.00 (15.7–24.7)	50,100.0	0.0015	+0.11 (negl.)
Security Perf. (%)	0.00 (0.0–0.0)	36.00 (30.7–41.7)	28,800.0	$<$0.001	$-$0.36 (med.)
Governance Perf. (%)	0.00 (0.0–0.0)	35.67 (30.3–41.3)	28,950.0	$<$0.001	$-$0.36 (med.)
Policies Trig. (mean)	0.00 (0.0–0.0)	0.37 (0.31–0.42)	28,500.0	$<$0.001	$-$0.37 (med.)
Latency Mean (s)	51.2 (37.1–74.4)	315.6 (284.8–347.2)	33,224.0	$<$0.001	$-$0.26 (small)
Score Mean (0–100)	73.2 (72.9–73.5)	28.5 (24.6–32.4)	71,304.5	$<$0.001	+0.59 (large)

Ninety-five percent confidence intervals (10,000-resample bootstrap, percentile method) and Cliff's δ complement the p-values with practical magnitude. Most metrics show medium-to-large effects consistent with $p < 0.001$ (Score: $\delta = +0.59$; Governance/Security/Policies: $\delta \approx -0.36$), but Functional Performance is a notable exception: despite reaching significance ($p = 0.0015$), its effect size is negligible ($\delta = +0.11$), indicating the cost to functional throughput is small relative to the governance and security gains. Latency's comparatively smaller δ (-0.26, small) versus its large mean gap reflects the right-skew introduced by timeout-capped interactions, which the rank-based δ discounts more than the mean.

Part of this increase reflects the additional processing stages introduced by the governance layer, including validation, monitoring, auditing, policy verification, and risk assessment; however, the elevated mean and standard deviation are also driven by a subset of interactions in which the additional computational load imposed by the ModelOps pipeline exceeded the available processing capacity, resulting in connection read-timeouts (capped at 600 s) before a response could be generated. These cases are retained in the analysis, as they represent a direct operational consequence of the computational overhead introduced by the AI TRiSM mechanisms rather than an unrelated infrastructure fault.

This effect is also visible in the average response quality (Score Mean), which decreased from 73.19 to 28.45 ($U = 71{,}304.5$, $p < 0.001$), and in the average score gap relative to the minimum required threshold, which dropped from -10.14 in the baseline to -54.88 in the TRiSM-enabled environment. This reduction stems both from the timeout-affected interactions, which are scored as zero, and from the stricter governance controls that intentionally block or restrict requests considered risky or non-compliant.

Consistently, *Functional Performance* decreased from 31.33% to 20.00% ($U = 50{,}100.0$, $p = 0.0015$), reflecting the combined effect of computational overhead and

stricter validation criteria rather than a loss of intrinsic model capability. The average trust/confidence score likewise dropped from 0.50 to 0.21, consistent with the lower and more variable response quality observed under TRiSM.

In contrast, *Security Performance* rose from 0.00% to 36.00% ($U = 28{,}800.0$, $p < 0.001$), since the baseline environment implements no security controls and is therefore unable to detect security violations or OWASP-related risk categories (mean violations and OWASP categories per interaction both equal to 0.00), whereas the TRiSM-enabled environment identifies an average of 0.36 violations and 0.69 OWASP categories per interaction.

Similarly, *Governance Performance* increased from 0.00% to 35.67% ($U = 28{,}950.0$, $p < 0.001$), and the average number of triggered policies rose from 0.00 to 0.37 per interaction ($U = 28{,}500.0$, $p < 0.001$), indicating that the governance layer actively enforced operational controls and intervened in a substantial share of potentially unsafe or computationally costly interactions throughout benchmark execution. The token ratio (output-to-input token count, computed only for interactions that completed inference) also differed significantly between environments ($p < 0.001$), reflecting changes in response length and verbosity induced by the governance layer.

To verify whether these differences are statistically significant, the Mann–Whitney U test was applied to all evaluated metrics. For two independent samples of sizes n_1 and n_2, the test statistic is given by

$$U = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1, \tag{1}$$

where R_1 is the sum of ranks assigned to the first sample when both samples are pooled and ranked jointly. As shown in Table 3, statistically significant differences ($p < 0.05$) were observed for all evaluated variables, indicating that the changes between environments are unlikely to be explained by random variation and instead reflect the operational impact of the implemented ModelOps mechanisms, supporting the rejection of H_0.

Taken together, these results indicate that the introduction of AI TRiSM-inspired ModelOps mechanisms substantially increased the security and governance capabilities of the evaluated system, at the cost of higher latency and a higher rate of incomplete or low-scoring interactions, partly caused by the additional computational load that the governance pipeline imposes on the underlying model.

Although this overhead reduced the functional approval rate, the resulting mechanisms improved governance visibility, operational traceability, monitoring effectiveness, and policy enforcement, enabling the detection, auditing, and containment of unsafe or non-compliant interactions. These findings provide empirical evidence that ModelOps mechanisms can substantially influence the operational dynamics of generative AI systems, while also highlighting the importance of provisioning adequate computational resources to support governance-oriented lifecycle management in production environments.

6. Conclusion

This work investigated the effects of ModelOps mechanisms on the operational behavior of a local generative AI system based on Phi-3.5 and Ollama.

To support the evaluation, two environments were implemented: a baseline configuration without governance controls and a protected configuration incorporating ModelOps mechanisms for monitoring, auditing, policy enforcement, and operational supervision. A benchmark composed of 300 prompts, generated with the assistance of generative AI and validated through human review, was executed in both environments.

The results demonstrated that the introduction of ModelOps significantly altered the behavior of the evaluated system. The protected environment actively activated governance policies and blocked a substantial share of requests, all of which were associated with detected security violations and high- or critical-risk classifications, confirming that blocking was consistently driven by genuinely unsafe interactions rather than indiscriminate restriction.

These mechanisms also modified operational indicators such as latency, score, and approval rate. Statistical hypothesis testing (Mann–Whitney U, $p < 0.05$) confirmed that these differences were significant across all evaluated metrics, demonstrating that the baseline environment was unable to detect or mitigate risky interactions, whereas the AI TRiSM-enabled environment successfully identified security-related events and increased operational visibility, traceability, and control over model execution.

As discussed in Section 5, part of the latency and score degradation stems from timeout-affected interactions rather than from the governance logic itself; the drop in trust should likewise be read as a downstream effect of computational overhead, not as evidence that ModelOps recalibrates model confidence. Even so, stricter controls measurably increased the system's ability to detect and contain unsafe interactions before delivery.

Therefore, the null hypothesis (H_0) is rejected and the alternative hypothesis (H_1) is accepted, indicating that ModelOps mechanisms produce statistically significant changes in the operational behavior of the evaluated generative AI system. The findings reinforce the importance of ModelOps as a practical governance component for generative AI systems, providing mechanisms for monitoring, auditing, policy enforcement, and operational risk control. Although these controls introduce computational overhead, they contribute to greater transparency, accountability, and reliability throughout the AI lifecycle.

Some limitations of this study should be acknowledged. First, the experiments were conducted using a single SLM executed locally through Ollama, which may limit the generalization of the results to other language models. Second, although the benchmark dataset was inspired by established literature and validated through human review, the prompts were generated with the assistance of generative AI, which may introduce generation biases. Third, the computational overhead introduced by the governance pipeline led to a non-negligible number of connection timeouts under the available hardware, which limits the extent to which the observed latency and score degradation can be attributed exclusively to the governance logic itself, as opposed to the specific computational resources available during testing.

Furthermore, the evaluation focused on operational indicators extracted from audit logs and governance records. Important ModelOps capabilities such as model drift detection, automated retraining, continuous deployment pipelines, rollback strategies, model versioning, and long-term production monitoring were outside the scope of this work. Finally, the experiments were conducted in a controlled environment and may not fully represent the complexity of real organizational deployments.

Future work calls for evaluating the governance pipeline on higher-capacity computational infrastructure to isolate governance overhead from hardware latency, as well as expanding empirical testing to larger language models and real-world organizational workloads. Additionally, future research is encouraged to integrate the Adversarial Robustness Toolbox (ART) [Trusted-AI 2024][4] for automated robustness testing under Pillar P5, alongside comparative benchmarks with alternative AI governance frameworks and guardrail architectures (e.g., NIST AI RMF, NeMo Guardrails, and Guardrails AI) to assess cross-framework trade-offs in latency, security coverage, and policy enforcement efficacy.

References

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 610–623, New York, NY, USA. Association for Computing Machinery.

Bommasani, R., Kapoor, S., Klyman, K., Longpre, S., Ramaswami, A., Zhang, D., Schaake, M., Ho, D. E., Narayanan, A., and Liang, P. (2024). Considerations for governing open foundation models. *Science*, 386(6718):151–153.

Chicco, D., Sichenze, A., and Jurman, G. (2025). A simple guide to the use of Student's t-test, Mann-Whitney U test, Chi-squared test, and Kruskal-Wallis test in biostatistics. *BioData Mining*, 18(1):56.

Debenedetti, E., Zhang, J., Balunovic, M., Beurer-Kellner, L., Fischer, M., and Tramèr, F. (2024). AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems*, volume 37, pages 82895–82920.

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., and Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 79–90. ACM.

Grönman, J., Pohjola, T., and Haukioja, T. (2025). Cybersecurity challenges in open data and AI systems for sustainable local development. In *2025 MIPRO 48th ICT and Electronics Convention*, pages 1088–1093.

Litan, A. (2026). AI trust and AI risk: Tackling trust, risk and security in AI models. Available at: `https://www.gartner.com/en/articles/ai-trust-and-ai-risk`. Accessed: 2026-06-20.

[4] `https://github.com/Trusted-AI/adversarial-robustness-toolbox`

Matotek, K., Cassel, H., Amiruzzaman, M., and Ngo, L. B. (2025). Evaluating the limitations of local LLMs in solving complex programming challenges.

Ray, P. P. (2026). A review of TRiSM frameworks in artificial intelligence systems: Fundamentals, taxonomy, use cases, key challenges and future directions. *Expert Systems*, 43(3):e70213.

Raza, S., Sapkota, R., Karkee, M., and Emmanouilidis, C. (2026). TRiSM for agentic AI: A review of trust, risk, and security management in LLM-based agentic multi-agent systems. *AI Open*, 7:71–95.

Romano, J., Kromrey, J. D., Coraggio, J., and Skowronek, J. (2006). Appropriate statistics for ordinal level data: Should we really be using t-test and cohen's d for evaluating group differences on the nsse and other surveys? *Annual Meeting of the Florida Association of Institutional Research*, pages 1–33.

Shahin, N. and Alsmadi, I. (2026). Benchmarking Llama model security against OWASP top 10 for LLM applications. arXiv:2601.19970v1.

Trusted-AI (2024). Adversarial robustness toolbox (art). `https://github.com/Trusted-AI/adversarial-robustness-toolbox`. Accessed: 2026-08-13.

Wang, W.-C. (2024). Legal, policy, and compliance issues in using AI for security: Using Taiwan's cybersecurity management act and penetration testing as examples. In *2024 16th International Conference on Cyber Conflict: Over the Horizon (CyCon)*, pages 161–176.

Zhang, C., Jin, M., Yu, Q., Liu, C., Xue, H., and Jin, X. (2024). Goal-guided generative prompt injection attack on large language models. In *2024 IEEE International Conference on Data Mining (ICDM)*, pages 941–946. IEEE.

Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., and Fredrikson, M. (2023). Universal and transferable adversarial attacks on aligned language models.