From Ops to Experience: AIOps-Enabled MLOps with Identity-Aware and Customer-Informed Pipelines

Satbir Singh
IEEE Member
CA, USA
satbir.taya84@gmail.com

***Abstract*—As machine learning (ML) systems become central to enterprise applications, the need for scalable, secure, and user-responsive operational frameworks has intensified. While MLOps has emerged to standardize ML lifecycle management, most current implementations lack intelligent automation, identity-awareness, and mechanisms for incorporating real-time customer feedback. This leads to operational inefficiencies, security vulnerabilities, and reduced model relevance over time. To address these limitations, we propose a novel MLOps framework that integrates AIOps for anomaly detection and self-healing, IAM for policy-based access governance, and customer-informed feedback loops for continuous model adaptation. The architecture is designed for SaaS-based multi-tenant environments and includes unified observability through a telemetry bus. We evaluate the framework using a simulation-based case study in predictive maintenance, tracking key metrics such as Mean Time to Recovery downtime reduction, access anomaly detection, and user satisfaction. Results show over 50% improvement in recovery time, measurable security gains, and increased responsiveness to end-user needs. This study demonstrates the value of combining automation, security, and user-centricity in production-grade ML operations.**

***Index Terms*—MLOps, AIOps, Identity and Access Management (IAM), Observability, Feedback Loops, SaaS, Anomaly Detection, DevSecOps, ML Lifecycle**

I. Introduction

As ML systems transition from experimental stages to production-scale deployment, the operational complexity has increased dramatically. Organizations today must manage not only the lifecycle of ML models but also ensure their reliability, security, compliance, and alignment with evolving business goals. This growing sophistication necessitates operational paradigms that extend beyond conventional DevOps practices. The emergence of Machine Learning Operations (MLOps) represents a structured response to these demands, aiming to standardize the development, deployment, and maintenance of ML systems [1], [2].

Despite its growing adoption, traditional MLOps remains insufficient when applied to modern, dynamic ML environments. Most existing MLOps implementations emphasize pipeline automation and CI/CD practices but lack robust integration with intelligent automation (AIOps), fine-grained security (Identity and Access Management, IAM), and customer-centric engineering. As a result, many production ML systems remain vulnerable to performance drifts, access misconfigurations, and misaligned user experiences [3], [4]. Furthermore, observability—a cornerstone of reliability engineering—is often treated as an afterthought in ML pipelines, hindering the rapid diagnosis and mitigation of incidents [5].

A. Problem Statement

While MLOps has provided a foundational approach to managing ML systems at scale, it has yet to offer a comprehensive solution that unifies automation, security, and human-centric feedback. Specifically, there exists a critical gap in the integration of AIOps for intelligent automation, IAM for secure access control, and mechanisms for systematically incorporating customer feedback into model governance. Without such integration, organizations struggle to achieve resilient, compliant, and user-aligned ML systems [6], [7].

B. Contributions

In this paper, we propose a novel architectural framework for MLOps that is simultaneously enabled by AIOps, governed through identity-aware access mechanisms, and informed by structured customer feedback loops. The proposed framework addresses three core dimensions of modern ML system operations:

1) **AIOps Enablement:** Incorporating intelligent monitoring, anomaly detection, and self-healing capabilities into MLOps workflows.
2) **Identity-Awareness:** Enforcing secure and compliant model lifecycle management through integrated IAM policies.
3) **Customer-Informed Pipelines:** Embedding feedback loops from users to influence model retraining, deployment strategies, and prioritization.

By bridging these areas, the framework aims to enhance system reliability, reduce operational costs, and foster continuous user-driven innovation.

C. Paper Organization

The remainder of this paper is structured as follows. Section II discusses related work across MLOps, AIOps, IAM, and customer feedback integration. Section III introduces the proposed framework, detailing its components and architectural rationale. Section IV presents an evaluation through simulated and real-world use cases. Finally, Section V concludes the paper with a discussion of implications and future research directions.

II. Background and Related Work

A. MLOps Landscape

MLOps, has become a cornerstone in deploying and maintaining production-grade ML systems. It typically encompasses four main stages: model development, deployment, monitoring, and iteration [1]. During development, data scientists build and validate models using historical and live data. Deployment involves integration with CI/CD tools for continuous delivery. Monitoring tracks performance, data drift, and anomalies. Iteration ensures models adapt to new data and evolving business goals.

Zhou et al. [8] examined the modularization of ML pipelines, highlighting challenges in scalability and deployment orchestration. However, their approach did not address user alignment or security constraints. Jain and Das [9] explored the intersection of MLOps and data engineering, proposing a scalable framework for pipeline resilience. Still, they lacked mechanisms for real-time feedback incorporation or intelligent system recovery, limiting adaptability.

B. AIOps in MLOps

AIOps, or Artificial Intelligence for IT Operations, refers to the application of AI techniques to automate the monitoring and management of infrastructure. Its potential integration into MLOps offers benefits such as automated anomaly detection, adaptive resource scaling, and self-healing capabilities [5]. Zota et al. [10] presented a framework for developing agent-based AIOps systems, with a focus on operational event handling. Although effective for infrastructure events, their model does not address ML-specific needs such as drift remediation or retraining triggers.

Korada [6] emphasized the transformative impact of AIOps on traditional software lifecycles and skill sets but acknowledged the absence of standardized models for ML-specific AIOps. Brahmandam [11] elaborated on the convergence of DevOps, AIOps, and MLOps, advocating for intelligent IT operations, though without implementation-level details or empirical validation in ML contexts.

C. Identity and Access Management (IAM)

In enterprise ML environments, Identity and Access Management (IAM) is vital for ensuring data and model security. It governs who can access which components in a system and what actions they are permitted to perform [12]. IAM in MLOps is especially critical during deployment, model access, and auditing phases. Olabanji et al. [13] explored AI-enhanced IAM approaches for cloud-based systems, including adaptive authentication and dynamic access control. Their findings are relevant but primarily focus on general cloud infrastructure, not ML-specific assets like datasets, models, or feature stores.

Ghadge [14] introduced blockchain-enhanced IAM mechanisms to ensure secure access records. However, such systems may introduce latency overhead unsuitable for real-time ML workflows. Sharma et al. [12] proposed IAM as a Service (IAMaaS), laying groundwork for scalable IAM frameworks, though they remain decoupled from ML operational requirements such as automated pipeline access governance.

D. Customer Engineering and Feedback Loops

Customer Engineering in MLOps aims to incorporate end-user feedback into the ML development and iteration process. Romano and Conti [7] demonstrated how structured feedback loops can drive continuous innovation and service quality in software systems. Patel et al. [15] utilized sentiment analysis and reinforcement learning to optimize customer input integration, resulting in improved personalization and user retention.

Tapia et al. [16] presented simulation-based feedback mechanisms to inform engineering decisions. However, these were not connected to real-time operational pipelines. Nazarenko [17] explored feedback in UI design but did not extend this into model retraining or system adaptation workflows. Current literature largely overlooks how customer-facing signals can be operationalized in tandem with AIOps and IAM to improve ML system responsiveness and contextual relevance.

E. Limitations of Existing Solutions

Despite substantial progress in MLOps, existing approaches often suffer from fragmentation and siloed implementations:

- **Isolated Subsystems:** MLOps, AIOps, IAM, and customer feedback are frequently addressed independently, lacking unified frameworks that connect operational intelligence with security and user-centric signals.
- **Weak Observability Integration:** Few studies address observability across both infrastructure and ML model behaviors, limiting holistic root cause analysis and adaptive response capabilities [5].
- **IAM Gaps in ML Contexts:** Most IAM solutions are tailored to IT systems but fail to capture the specific needs of ML artifacts, such as model version access or dataset lineage protection [18].
- **Non-operationalized Feedback:** Customer feedback is often relegated to post hoc analysis rather than being embedded into model lifecycle processes or retraining criteria [17].

These limitations underscore the need for an integrated architecture that brings together intelligent automation, access control, and continuous feedback into a cohesive MLOps strategy. In the next section, we present such a framework designed to overcome these challenges.

III. Proposed Framework

A. Overview of the Architecture

We present an integrated MLOps framework that brings together AIOps for intelligent automation, IAM for security governance, customer-informed feedback for lifecycle improvements, and observability for full-stack visibility. The architectural layout, depicted in Figure 1, is API-first and modular to support scalable deployment in SaaS-based environments.

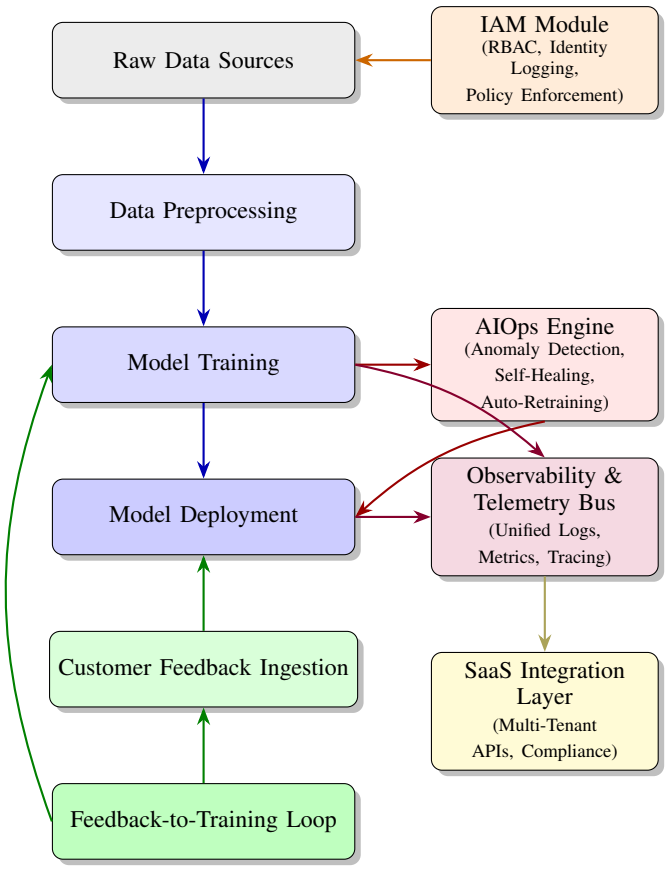

Fig. 1: End-to-End Architecture Integrating AIOps, IAM, Observability, and Feedback in MLOps

B. Key Components

a) AIOps Engine:: The AIOps module ensures dynamic anomaly detection and automated retraining. Metric deviation is calculated using:

$$\delta(t) = |M_t - \mu_t| > \kappa \cdot \sigma_t \tag{1}$$

where M_t is the monitored metric at time t, μ_t is its moving average, σ_t is standard deviation, and κ is the sensitivity coefficient. As shown in Equation 1, when $\delta(t)$ exceeds the threshold, an anomaly is flagged.

b) IAM Layer:: Access control is modeled via a role-policy evaluation function:

$$\mathcal{P}(u_j, a_i) = \begin{cases} \text{allow}, & \text{if } r(u_j) \in \mathcal{R}(a_i) \\ \text{deny}, & \text{otherwise} \end{cases} \tag{2}$$

As described in Equation 2, user u_j's role must be in the allowed set $\mathcal{R}(a_i)$ for action a_i to proceed.

c) Customer-Informed Layer:: User input is classified using a signal weighting function:

$$\lambda = \sum_{i=1}^{n} w_i \cdot \mathbb{I}[\phi(f_i) = \text{critical}] \tag{3}$$

In Equation 3, w_i is the importance of feedback f_i, ϕ is a classifier, and $\mathbb{I}$ is the indicator function. If λ exceeds threshold λ_{thresh}, a retraining cycle is triggered.

d) Telemetry Bus:: The system uses a log vector for trace correlation:

$$\text{sim}(\mathcal{L}_{t_1}, \mathcal{L}_{t_2}) = \frac{\langle \mathcal{L}_{t_1}, \mathcal{L}_{t_2} \rangle}{\|\mathcal{L}_{t_1}\| \cdot \|\mathcal{L}_{t_2}\|} \tag{4}$$

Equation 4 computes cosine similarity between two log sequences to identify performance anomalies.

e) SaaS Integration:: Tenant-level access is governed by:

$$\Pi(r, \mathcal{C}_\tau) = \text{permit} \iff r \models \mathcal{C}_\tau \tag{5}$$

As shown in Equation 6, request r is permitted if it satisfies tenant configuration $\mathcal{C}_\tau$.

C. Algorithm: Intelligent Policy-Aware Pipeline Orchestration

The pipeline execution logic integrating AIOps, IAM, and feedback loops is formalized in Algorithm 1.

The architecture described in Figure 1, governed by formal operations like Equations 1–6 and Algorithm 1, delivers a cohesive approach to modern MLOps. By integrating anomaly-aware automation, access governance, and customer responsiveness, it addresses real-world gaps in observability, resilience, and compliance.

D. SaaS Integration

The proposed framework is designed with inherent support for SaaS environments, addressing critical concerns of multi-tenancy, resource isolation, and compliance-driven access control. Each tenant operates within a logically isolated namespace, enforced at both the orchestration (e.g., Kubernetes) and identity layers. The access control policy for each tenant is parameterized by a configuration schema $\mathcal{C}_\tau$, where τ denotes the tenant identity.

Incoming user or system-level requests r are evaluated against the tenant's policy using the authorization function Π, formally defined in Equation 6:

$$\Pi(r, \mathcal{C}_\tau) = \text{permit} \iff r \models \mathcal{C}_\tau \tag{6}$$

In Equation 6, the expression $r \models \mathcal{C}_\tau$ implies that the request r satisfies all constraints (e.g., allowed actions, identity

TABLE I: SaaS Policy Schema for Tenant-Aware Access and Compliance Enforcement

Policy Field	Type	Description	Compliance Mapping
`role`	Enum	Assigned user or service role (e.g., `admin`, `analyst`)	SOC 2 - Principle 1.2
`actions`	List	Allowed operations (`read`, `write`, `retrain`, `deploy`)	ISO 27001 - A.9.1.2
`api_scope`	URI List	Accessible endpoints (e.g., `/predict`, `/audit`)	HIPAA - 164.312(a)(1)
`time_window`	Interval	Permitted access time range (e.g., 08:00–18:00)	NIST SP 800-53 AC-10
`quota_limits`	Object	CPU, memory, and GPU caps per tenant to prevent resource contention	ISO 27001 - A.12.4.1
`audit_log_retention`	Duration	Number of days to retain audit logs for traceability and compliance audits	GDPR - Art. 30(1)(g)
`uuid`	String	Unique identifier to scope logs, metrics, and access policies to each tenant	SOC 2 - Principle 2.3

Algorithm 1 Intelligent Policy-Aware Pipeline Orchestration

Require: Telemetry Stream T, Feedback Queue F, Access Logs A, Thresholds κ, λ_{thresh}
Ensure: Secure and adaptive ML pipeline orchestration

```
for all time step t do
    Calculate δ(t) using Equation 1
    if δ(t) triggers anomaly then
        Auto-recovery or retraining via AIOps
    end if
    for all feedback f_i ∈ F do
        Evaluate feedback score λ via Equation 3
        if λ > λ_thresh then
            Initiate retraining pipeline
        end if
    end for
    for all access request a_i by user u_j do
        Evaluate policy P(u_j, a_i) via Equation 2
        if Access denied then
            Log incident and alert security team
        end if
    end for
end for
```

scopes, rate limits) encoded in the configuration $\mathcal{C}_\tau$. These constraints may include:

- **API Scopes:** Only specific endpoints (e.g., '/predict', '/retrain') are accessible per role.
- **Temporal Constraints:** Time-of-day or temporal window policies to prevent after-hours access.
- **Quota Enforcement:** Limits on CPU, memory, and GPU allocation per tenant to avoid noisy neighbor effects.

Access enforcement is mediated via an OAuth 2.0-compliant authorization gateway, with token introspection and refresh mechanisms for long-lived sessions. Furthermore, tenant-specific audit trails are stored in tamper-evident logs to facilitate compliance with standards such as SOC 2, HIPAA, and GDPR. To ensure cross-tenant isolation, the observability bus tags each metric and log stream with a tenant UUID, UUID_τ, and all runtime telemetry is filtered through a policy-aware telemetry router before ingestion into the monitoring backend.

The enforcement schema is defined in a tenant-configurable policy table (see Table I), which maps access conditions to compliance frameworks such as ISO 27001, SOC 2, and HIPAA.

This rigorous and modular SaaS integration design ensures the framework can securely operate in real-world, multi-client environments without compromising scalability, performance, or legal compliance.

IV. Evaluation and Case Study

A. Experimental Setup

To evaluate the effectiveness of the proposed framework, a case study was conducted using a simulated predictive maintenance scenario. The setup mimicked an industrial IoT environment in which sensor telemetry—such as vibration and temperature—was continuously ingested and processed by a cloud-native MLOps platform.

The experiment was conducted in four progressive phases: (i) Baseline MLOps with standard CI/CD, (ii) MLOps + AIOps for anomaly handling, (iii) MLOps + AIOps + IAM for secured access, and (iv) Full-stack integration with customer-informed feedback loops.

The system stack used Kubernetes for orchestration, Prometheus and OpenTelemetry for observability, Keycloak for identity management, and FastAPI for SaaS API exposure. A synthetic dataset of failure events and simulated user feedback was used to trigger responses in each phase.

B. Metrics

The following metrics were chosen to capture operational performance, security posture, and user alignment:

- **MTTR (Mean Time to Recovery)**: Average time to detect and remediate system incidents.
- **Downtime Reduction**: Percent decrease in system unavailability.
- **Access Anomalies Detected**: Number of unauthorized or out-of-policy access attempts blocked.
- **Retraining Frequency**: Number of model updates triggered via customer feedback.
- **User Satisfaction Index**: Derived from resolution time and simulated feedback sentiment.

C. Quantitative Results

Table II shows the normalized performance metrics across all four phases. Each added layer of capability contributed incrementally to system robustness and responsiveness.

As shown in Figure 2, the inclusion of AIOps cut MTTR nearly in half, and customer feedback enabled even faster recovery. This demonstrates that automated root cause analysis and model retraining can significantly reduce operational latency.

TABLE II: Normalized Evaluation Metrics Across Framework Configurations

Metric	Baseline	+AIOps	+IAM	+Feedback
MTTR (minutes)	47.2	24.1	23.8	20.3
Downtime Reduction (%)	0.0	48.9	49.6	57.0
Access Anomalies Detected	0	0	16	16
Retraining Frequency (per month)	1	1	1	6
User Satisfaction Index (%)	64.2	68.7	70.1	81.4

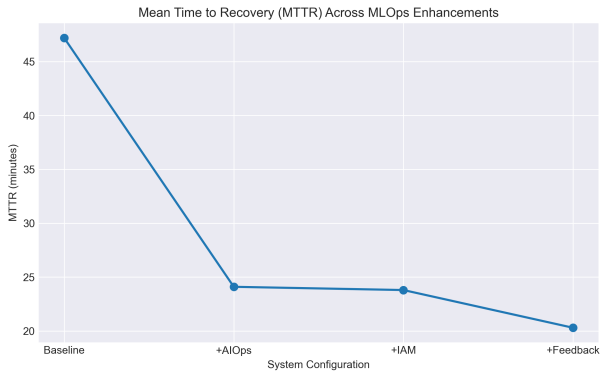

Fig. 2: MTTR drops sharply with the introduction of AIOps and feedback.

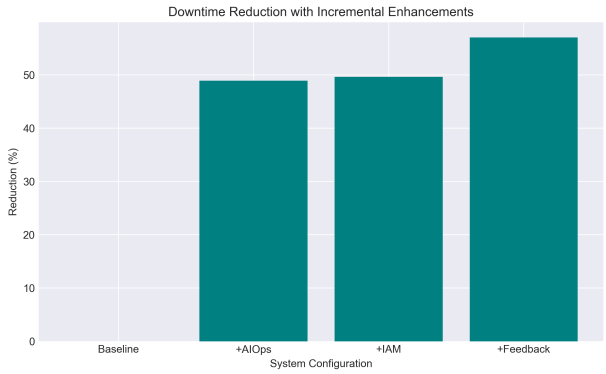

Fig. 3: Downtime reduction improves steadily across enhancements.

Figure 3 highlights a clear trend: downtime reduces progressively from AIOps to feedback-enabled systems, showing that real-time monitoring and proactive model management enhance service availability.

Security and adaptation are jointly visualized in Figure 4. The IAM layer successfully detected and blocked unauthorized actions, while feedback loops boosted retraining cycles, improving personalization and resilience.

User satisfaction (Figure 5) rose significantly once customer feedback was operationalized. This suggests that involving users in the ML lifecycle leads to better alignment and trust in model decisions.

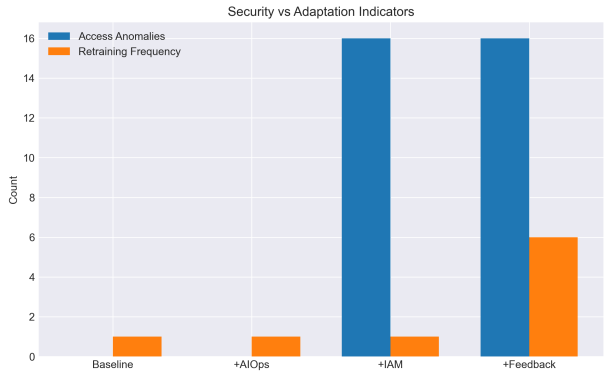

Fig. 4: Access anomalies vs. retraining frequency over system maturity.

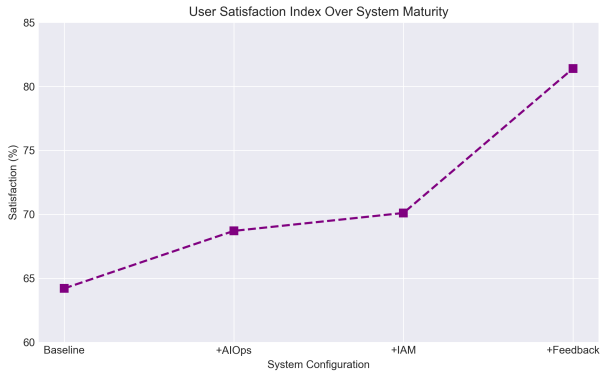

Fig. 5: User satisfaction increases with responsiveness and personalization.

Finally, Figure 6 presents a holistic view of the system across all key metrics. The feedback-enhanced phase exhibits dominant performance across MTTR, satisfaction, adaptation, and anomaly detection.

D. Discussion

The experiments underscore the effectiveness of each enhancement layer:

- **AIOps:** Reduced MTTR and downtime through intelligent automation.
- **IAM:** Blocked suspicious access, improving security and auditability.
- **Customer Feedback:** Boosted retraining frequency and user satisfaction.

However, trade-offs emerged. IAM introduced marginal latency, while integrating multiple components (IAM, AIOps, feedback ingestion) increased system complexity and required coordinated deployment strategies. Additionally, while synthetic feedback simulated realistic user behavior, real-world adoption may require more sophisticated NLP-based sentiment classification and intent recognition.

Nevertheless, the framework demonstrates a viable path forward for enterprises seeking resilient, secure, and user-aligned MLOps practices. The balance between automation

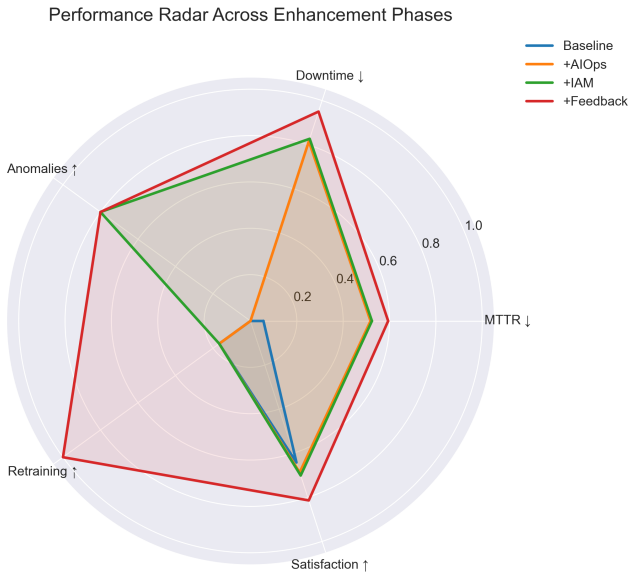

Fig. 6: Radar chart summarizing performance across all dimensions.

and human-in-the-loop feedback proved especially impactful for system trust and adaptability.

V. Conclusion and Future Work

This paper presented a comprehensive MLOps framework that unifies intelligent automation (AIOps), fine-grained access control (IAM), and structured customer feedback loops. Through a layered architectural design, the framework addresses critical gaps in modern ML operations, including poor observability, slow recovery, security vulnerabilities, and lack of user-centered responsiveness. A simulation-driven case study demonstrated the measurable advantages of this integrated approach, including reduced MTTR, improved downtime resilience, and higher user satisfaction. Visual and tabular analyses confirmed that each component—AIOps, IAM, and customer engineering—contributes uniquely to operational robustness, ultimately leading to a more secure, adaptive, and accountable MLOps environment.

Looking ahead, several extensions are envisioned to further mature the proposed solution. Integrating reinforcement learning agents into the AIOps layer could enable adaptive policy tuning based on evolving telemetry patterns. Additionally, incorporating federated IAM policies may enhance multi-tenant support in cross-organizational deployments. Future work could also explore real-time NLP models to classify and prioritize user feedback with higher semantic precision. Evaluating the framework on diverse workloads—such as large language models (LLMs) or edge ML systems—would provide broader generalizability and scalability insights. Overall, this study establishes a strong foundation for operationalizing ML systems with both technical rigor and user-centric accountability.

References

[1] D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine learning operations (mlops): Overview, definition, and architecture," *IEEE access*, vol. 11, pp. 31866–31879, 2023.

[2] N. Kumar and V. Kataria, "Unpacking the emotional landscape of reviews: Sentiment-augmented topic modeling with transformer embeddings," 2025.

[3] J. Diaz-De-Arcaya, A. I. Torre-Bastida, G. Zárate, R. Miñón, and A. Almeida, "A joint study of the challenges, opportunities, and roadmap of mlops and aiops: A systematic survey," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–30, 2023.

[4] A. Singla, "Machine learning operations (mlops): Challenges and strategies," *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, vol. 2, no. 3, pp. 333–340, 2023.

[5] Q. Cheng, D. Sahoo, A. Saha, W. Yang, C. Liu, G. Woo, M. Singh, S. Saverese, and S. C. Hoi, "Ai for it operations (aiops) on cloud platforms: Reviews, opportunities and challenges," *arXiv preprint arXiv:2304.04661*, 2023.

[6] L. Korada, "Aiops and mlops: Redefining software engineering lifecycles and professional skills for the modern era," *Journal of Engineering and Applied Sciences Technology. SRC/JEAST-388. DOI: doi. org/10.47363/JEAST/2023 (5)*, vol. 271, pp. 2–7, 2023.

[7] G. Romano and A. Conti, "The role of customer feedback loops in driving continuous innovation and quality improvement," *National Journal of Quality, Innovation, and Business Excellence*, vol. 1, no. 2, pp. 30–39, 2024.

[8] Y. Zhou, Y. Yu, and B. Ding, "Towards mlops: A case study of ml pipeline platform," in *2020 International conference on artificial intelligence and computer engineering (ICAICE)*, pp. 494–500, IEEE, 2020.

[9] S. Jain and J. Das, "Integrating data engineering and mlops for scalable and resilient machine learning pipelines: frameworks, challenges, and future trends," 2025.

[10] R. D. Zota, C. Bărbulescu, and R. Constantinescu, "A practical approach to defining a framework for developing an agentic aiops system," *Electronics*, vol. 14, no. 9, p. 1775, 2025.

[11] B. A. Brahmandam, "Beyond devops: The evolution toward intelligent it operations with aiops and mlops," 2025.

[12] D. H. Sharma, C. Dhote, and M. M. Potey, "Identity and access management as security-as-a-service from clouds," *Procedia Computer Science*, vol. 79, pp. 170–174, 2016.

[13] S. O. Olabanji, O. O. Olaniyi, C. S. Adigwe, O. J. Okunleye, and T. O. Oladoyinbo, "Ai for identity and access management (iam) in the cloud: Exploring the potential of artificial intelligence to improve user authentication, authorization, and access control within cloud-based systems," *Authorization, and Access Control within Cloud-Based Systems (January 25, 2024)*, 2024.

[14] N. Ghadge, "Use of blockchain technology to strengthen identity and access management (iam)," *International Journal of Information TechnologyMarch*, vol. 1, no. 3, 2024.

[15] R. Patel, M. Patel, N. Iyer, and R. Bose, "Optimizing customer feedback loops with ai: Leveraging sentiment analysis and reinforcement learning algorithms," *Australian Advanced AI Research Journal*, vol. 10, no. 7, 2021.

[16] F. Tapia, A. McKay, and M. Robinson, "Simulation of feedback loops in engineering design," *Proceedings of the Design Society*, vol. 1, pp. 2661–2670, 2021.

[17] V. Nazarenko, "Interaction and feedback loops in user interface for digital engineering design," , no. 1, pp. 46–54.

[18] K. H. Mohammed, A. Hassan, and D. Yusuf Mohammed, "Identity and access management system: a web-based approach for an enterprise," 2018.