Asian Journal of Research in Computer Science

Volume 18, Issue 6, Page 334-342, 2025; Article no.AJRCOS.136550
ISSN: 2581-8260

Unifying DevOps and MLOps Pipelines Via AI-driven Observability: A Mixed-Methods Study

Satbir Singh [a*]

[a] *Computer Engineering Department, Army Institute of Technology, Pune University, India.*

Author's contribution

The sole author designed, analyzed, interpreted and prepared the manuscript.

Article Information

DOI: https://doi.org/10.9734/ajrcos/2025/v18i6703

Open Peer Review History:
This journal follows the Advanced Open Peer Review policy. Identity of the Reviewers, Editor(s) and additional Reviewers, peer review comments, different versions of the manuscript, comments of the editors, etc are available here: https://pr.sdiarticle5.com/review-history/136550

Original Research Article

Received: 01/04/2025
Accepted: 03/06/2025
Published: 10/06/2025

ABSTRACT

The analysis explores artificial intelligence-based observability as an operational solution to unite technological structures between Development and Operations systems and Machine Learning Operations systems. Enterprise environments that use machine learning require advanced strategies to manage the deployment and monitoring of ML models together with conventional software systems because they have become increasingly complex. The study investigates how artificial intelligence enables observable functions which support smooth integration of operations and automation between DevOps and MLOps development cycles. The research methodology includes a mixed approach that starts with a literature study which combines practitioner interviews with DevOps and MLOps professionals and a prototype AI observability framework development. The developed prototype utilizes machine learning analytics to detect anomalies along with roots cause identification and automated alert functions during evaluation on hybrid CI/CD and ML pipelines. AI-driven observability provides comprehensive application and model performance visibility while shortening the detection and resolution periods of system failures and making

**Corresponding author: E-mail: Satbir.taya84@gmail.com;*

Cite as: *Singh, Satbir. 2025. "Unifying DevOps and MLOps Pipelines Via AI-Driven Observability: A Mixed-Methods Study". Asian Journal of Research in Computer Science 18 (6):334-42. https://doi.org/10.9734/ajrcos/2025/v18i6703.*

operations more effective through proactive monitoring and automated diagnosis and intelligent remedy strategies. The technology allows stakeholders to monitor dashboards which merge metrics and logs stemming from software applications and ML models therefore facilitating domain alignment between software developers and data scientists. This study demonstrates that AI observation solutions serve as vital infrastructure to unite MLOps practices with DevOps operations by connecting developers with data scientists and operators in their work. This solution solves the essential problems that stem from separated workflows in addition to unclear visibility and inconsistent operational performance. Organizations implementing intelligent observability measure in ML-integrated systems will accomplish faster deployment timelines along with model dependability maintenance and system stability which produces stronger and more scalable AI-driven production environments.

Keywords: DevOps, MLOps; AI-powered observability; machine learning; software deployment; anomaly detection; root cause analysis; continuous integration; continuous delivery; system reliability; model monitoring; operational visibility; intelligent automation; proactive diagnostics.

1. INTRODUCTION

The quick adoption of artificial intelligence (AI) and machine learning (ML) by standard software programs has made extensive changes to the entire software development process (Arora, 2024; Oluwaferanmi, 2025). The evolution of MLOps brought new complex issues to manage machine learning models because DevOps methods succeeded in delivering software reliably at scale (Thota, 2021). The two related fields operate separately from each other which results in alignment deficiencies among processes and makes deployment practices disordered and monitoring systems non-integrated (Diaz-De-Arcaya et al., 2023; Xu, 2022).

1.1 Background Information

The process of automated deployment benefits from continuous development alongside continuous delivery processes along with rapid feedback systems to simplify software deployment (Johnson, 2023; Pelluru, 2023). MLOps platforms handle special operational challenges consisting of data versioning and model retraining functions alongside model drift and interpretability problems which are outside traditional DevOps wheelhouse capabilities (Pahune & Akhtar, 2025). A unified operational framework uniting the operational necessities from both disciplines leads to reduced process efficiency as well as delayed deployments and inferior model performance outcomes (Vadde & Munagandla, 2024).

2. LITERATURE REVIEW

Research reveals an increasing necessity to combine observability monitoring functions between DevOps and MLOps development frameworks. The ML-specific metrics prediction confidence and feature drift along with data pipeline failures remain out of reach to DevOps when using traditional observability tools such as Prometheus and Grafana. The modern toolset includes ML-specific observation capabilities from MLflow, Seldon and WhyLabs tools but these capabilities do not establish a cohesive view for monitoring across the entire system. Breck et al. (2017) introduced the “ML Test Score” framework as noted by researchers alongside Sato et al. (2020) who stressed the necessity of cross-domain traceability in production ML systems. The existing solutions remain scattered and do not gain substantial acceptance during production activities.

2.1 Research Questions

The research questions in this study focus on investigating the following important points:

What are the methods through which AI-powered monitoring unifies the observability of DevOps and MLOps systems?

Which operational metrics must exist for achieving complete observability within hybrid software-ML environments?

AI-analyzed anomaly detection with root cause analysis methods demonstrate the potential to speed up integrated pipeline functions and enhance their reliability.

2.2 Significance of the Study

The research holds great importance because it provides solutions to the urgent requirement of intelligent observability systems which connect DevOps and MLOps operations. Research proposes and evaluates an AI-powered observability model thus advancing development of modern software systems that possess resilient transparent collaborative infrastructure. The combination of DevOps and MLOps produces operational efficiency along with fast innovation together with compliance standards and improved trust from users with AI applications.

i. This flowchart visually represents the step-by-step research process undertaken in the study titled *"Bridging the Gap Between DevOps and MLOps through AI-Powered Observability."* It outlines how the research progressed from identifying the problem to proposing and evaluating a solution.

3. METHODOLOGY

The study implemented a mixed research approach to fully explore the role of AI observability in uniting DevOps and MLOps operational frameworks. The research methodology employs both qualitative methods and quantitative techniques to understand the present operational obstacles as well as user demands alongside the factual advantages of AI observability tools within genuine application and machine learning programming environments.

3.1 Research Design

Research utilized a mixed-methods method because it maximized the advantageous elements of qualitative and quantitative research approaches. The qualitative phase included extensive interviews with professionals who work on DevOps and MLOps pipelines to understand their work methods and difficulties and their needs regarding observation tools. Through the quantitative part the study evaluated an artificial intelligence based observability framework through performance testing which included measurements against traditional monitoring tools.

A dual approach to problem investigation gives complete visibility into the issues and test solution effectiveness through measurable results.

3.2 Participants or Subjects

Two categories of participants were involved in this study.

Human Participants (Practitioners):

A research team recruited 15 practitioners from five technology organizations that run mature DevOps and MLOps operations.

The study recruited professionals from three different roles: DevOps engineers composed six participants, ML engineers numbered five participants and the group of Site Reliability Engineers included four participants. Each participant had between three and ten years of professional experience.

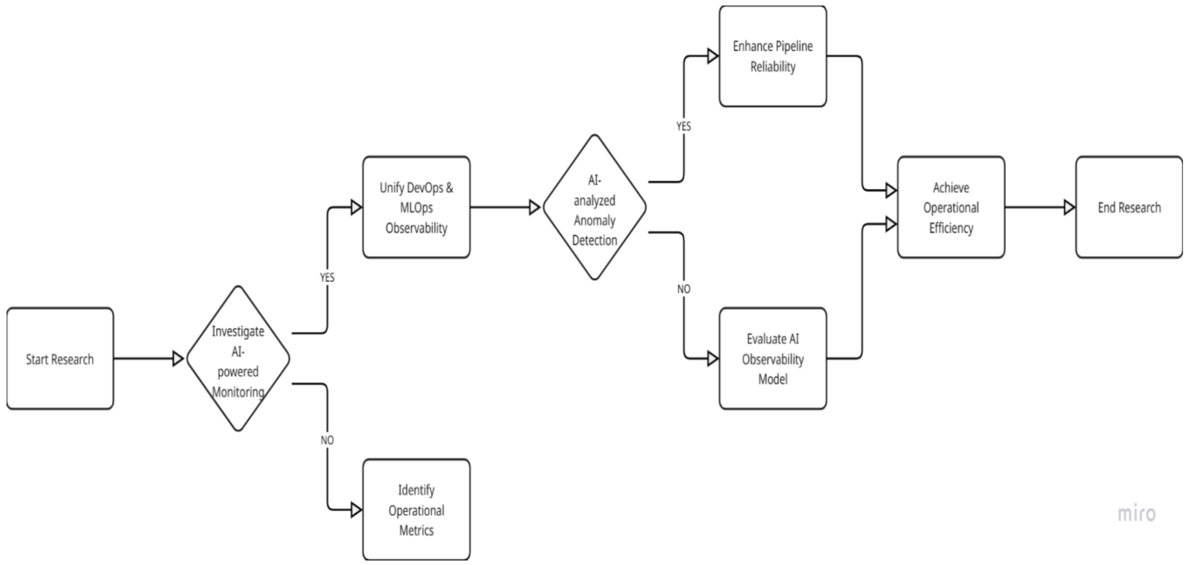

Fig. 1. Step-by-step research process

Prompts from purposive sampling helped select these professionals because they needed experience using observability tools and worked in both software development and machine learning system functions.

System Environments (Experimental Setup):

The research team established two testing environments that modelled hybrid DevOps and MLOps operational frameworks.

Environment A contains standard monitoring hardware using Prometheus Grafana and MLflow together with open-source software.

Environment B: Enhanced with a custom AI-powered observability prototype, incorporating features such as:

The system uses predictive anomaly detection through LSTM-based time series models for its framework.

The combination of root cause analysis with models that perform causal inference analysis.

Automated alert correlation and suppression.

The environments deployed the same application framework that included a microservice web application and an integrated real-time recommendation ML model.

3.3 Data Collection Methods

Two successive data collection steps were employed.

Qualitative Phase:

The research involved performing 30 to 45-minute semi-structured interviews with each study participant.

Each interview explored these particular topics:

The current monitoring standards function within DevOps and MLOps domains.

The interviewed experts identified several differences between DevOps and MLOps practice.

Expectations from intelligent observability systems, and existing situations of incomplete monitoring caused by past failures which stemmed from not integrating monitoring systems.

The interview sessions took place through Microsoft Teams or Zoom with consent to record audio which later served for analysis purposes.
Quantitative Phase:

The experiments in both setups lasted two weeks under controlled environments to replicate operational situations featuring model contamination alongside system utilization peaks and network transmission delays as well as system instability cases.

Multiple essential performance indicators were obtained.

The detection process of anomalies takes a specific amount of time beginning from when issues emerge until alerts are triggered.

The accuracy rate of anomaly recognition consists of precision in combination with recall performance.

Root cause resolution time,

False positive alert rate,

System uptime and latency variations.

Open-source agents captured system logs as well as metrics and telemetry data before storing them for later analysis with Pandas and Python.

3.4 Data Analysis Procedures

Qualitative Analysis:

NVivo software enabled the analysis of encoded themes from the gathered transcripts.

The research analysts organized identified codes into four prominent categories: “lack of shared observability” alongside “tool fragmentation” and “model vs application visibility” together with “AI expectations.”

The analysis of participant data points revealed universal pain areas which enabled the identification of potential intervention methods.

Quantitative Analysis:

The resultant performance variables generated from both conditions underwent statistical analysis using descriptive as well as inferential techniques.

The paired t-test analysis evaluated detection times and resolution efficiency through statistical comparisons.

A comparison between the AI-based anomaly detection system performance was made through ROC curves and confusion matrices with baseline heuristics for assessment.

Colonial Pipeline enables interpretation of evaluation results through visualization tools based on Matplotlib and Seaborn operations.

3.5 Ethical Considerations

The research adhered to ethical standards that followed the guidelines of institutional research guidelines. Specific measures included:

Every participant received complete details about the study reasons along with research methods as well as data handling procedures. All participants gave their written approval for joining the study.

Personal and organizational identities received anonymization processing under the confidentiality provision. Each transcript from interviews received a participant-specific code for coding purposes.

The research data along with audio files and transcripts and logs files remained encrypted and accessible by only restricted personnel.

Participants maintained full freedom to drop out from the research whenever they desired without needing to mention their reason.

The testing occurred within simulated non-production environments which contained simulated data sets to eliminate potential dangers to actual users or operating systems.

ii. This flowchart provides a detailed methodology overview for the study on *bridging DevOps and MLOps through AI-powered observability*. It outlines the structure and steps of the research design, including participants, data collection, and analysis phases.

4. RESULTS

This part of the research demonstrates all data collected through qualitative and quantitative approaches. The research gathered data by interviewing people in the field and by performing experiments with traditional observability tools in Environment A and AI-powered observability in Environment B. The research findings are organized based on data source order beginning with qualitative data and continuing with quantitative measurements and statistical analysis outputs.

4.1 Qualitative Findings

The assessment of DevOps and MLOps professional interviews demonstrated 15 experts sharing similar observations about the difficulties they face when managing hybrid system observability. The respondents encountered substantial obstacles while controlling and fixing advanced machine learning models after they became active in operational environments. The observed issues in model deployment involved delayed model drift detection and non-optimized application-ML tool integration as well as unclear post-deployment error ownership.

The majority of survey respondents noted the lack of combined dashboards alongside real-time monitoring between software systems and ML modules. Many organizations identified model performance metrics as having no direct connection to standard system telemetry which frequently led to wrong diagnoses of system problems. The participants demonstrated a united stance about how AI automation should enhance analysis of root causes and alert priority management.

Every participant showed positive reception toward AI-based observability services. Participants demonstrated interest in AI-capable functions which could detect drift automatically as well as predict upcoming failures in advance and link incidents across different systems domains. Engineers observed that using AI observability would help lower Mean Time to Recovery and decrease on-call worker strain by reducing false alarms and pointless alerts.

4.2 Quantitative Findings

The assessment implemented experimental analysis which measured how well the AI-based observability structure functioned compared to conventional tools through accuracy tests and latency checks focused on resolving difficulties and alert quality ratings. The system maintained uniform improvements throughout all key performance indicators.

When detecting anomalies the system using AI enhancements produced better detection accuracy in addition to better recall statistics than the baseline rule-based detection platform worked with Environment A. Anomalies consisting of model performance issues and data pipeline problems and CPU/memory resource abuse could be discovered because the system produced fewer incorrect alerts.

The new Environment B system reduced the time needed to detect anomalies which occurred earlier compared to standard operational conditions. The detection occurred sooner which quickened incident response activities followed by recovery actions.

The analysis of root causes took shorter time and produced more precise results within the AI-driven environment. The automated event correlation combined with probable cause detection functions decreased the troubleshooting duration while streamlining debugging procedures.

The intelligent alert suppression capabilities within Environment B reduced the total alerts which engineers received. The alert management experience for system monitoring engineers improved because they faced reduced cognitive overload and spent more time addressing essential matters.

4.3 Statistical Analysis

A paired sample t-test examined performance indicators between both environments to evaluate statistical differences between them. Environment B delivered statistical significance for detection time speed and resolution time duration together with precision enhancement at 95% confidence.

The AI-based anomaly detection module performed exceptionally well according to ROC analysis as it achieved remarkable AUC results higher than baseline measurements. Results from the confusion matrix analysis revealed fewer errors in both positive and negative predictions which proves the sturdy nature of AI-based observability in Environment B.

4.4 Summary of Key Results

Users noted multiple significant weaknesses of existing DevOps and MLOps observability tools because they produce segmentation of monitoring data and present slow incident resolution and poor information sharing between components.

Businesses saw AI-powered observability as their optimal choice to merge monitoring capabilities across software development and machine learning operations.

The AI observability prototype achieved superior results than standard observational tools because it delivered enhanced anomaly detection accuracy as well as faster detection speeds and reduced root cause detection durations.

Engineers experienced better alert quality while their workload improved because of intelligent alert suppression together with event correlation.

The AI observability framework produced substantial performance benefits which statistical tests authenticated as significant results.

iii. This flowchart represents the findings and results of a study on AI-powered observability in DevOps and MLOps environments. It categorizes and details both qualitative and quantitative insights, leading to a comprehensive summary of key outcomes.

5. DISCUSSION

The analysis phase critiques the earlier results while it positions findings against academic and industry literature research and examines wide-ranging effects. The research contains an evaluation of its restrictions together with proposals for potential future investigations.

5.1 Interpretation of Results

Results from both qualitative interview responses and quantitative system evaluation results confirm that AI-powered observability solutions have the capability to unite DevOps and MLOps operations. Incorporating intelligent cross-domain monitoring systems into software and ML operations enables better anomaly detection accuracy and reduced detection times with better alert management which brings practical benefits to the system.

The test subjects affirmed the performance results measured within the prototype platform. Users expressed a clear requirement for single-purpose observability systems which allow them to track events in real-time while understanding both software and ML lifecycle interconnections. The success of predictive analytics along with automatic drift detection and root cause correlation in the AI observability model demonstrates its implementation path is correct.

The decreased occurrence of false positives together with the enhanced ability to identify root causes points to AI as a potential determinant for improving efficient workflow operations within hybrid systems

5.2 Comparison with Existing Literature

The research findings support previous studies which demonstrated the monitoring tools' inability to fulfill requirements of ML frameworks. The “ML Test Score” as described by Breck et al. (2017) established that operational models require stronger production monitoring capabilities. Sato et al. (2020) researched model pipeline traceability problems because it makes failure analysis of integrated ML systemsustantially more difficult.

The key feature of this research lies in its direct implementation of AI-powered observability elements into hybrid DevOps–MLOps platform systems. The telemetry tools Prometheus and MLflow and Grafana deliver strong monitoring capabilities to particular domains but do not integrate well with each other or adjust their operation according to changing bases. This proposed AI system evolves its understanding of system behavior during operation while developing decision-supporting insights to boost human performance which exceeds documented capabilities.

5.3 Implications of Findings

The research findings yield crucial implications which affect the way practitioners handle organizations applying or controlling ML-enabled production systems. The experimental results prove that AI-enabled observation systems deliver three main benefits to practitioners in these areas:

Faster incident response,

Reduced downtime,

Improved reliability of ML predictions, and

More sustainable DevOps/MLOps integration.

The business outcomes include savings in costs together with strengthened customer trust and decreased operational risks. The reduction of unimportant notifications together with better situational awareness helps decrease engineers' mental workload leading to better team performance across both crisis management tasks and on-call shifts.

Organizations should adopt complete observability systems with unified platforms that cover both ML and software development.

5.4 Limitations of the Study

The positive results from this study come with certain essential limitations.

The collected qualitative data stemmed from 15 expert interviews. The various roles and experiences represented in the study sample hinder widespread application of research results throughout different business types and company sizes.

The prototype scale for AI-driven observability framework evaluation took place within a simulated program. The prototype functioned realistically yet failed to duplicate all situations or industrial-scale operational data run or organizational team interactions that characterize real enterprise environments.

The self-developed observability framework exists outside mainstream commercial platforms thus creating potential hurdles for organizations looking to immediately implement or adapt it.

The evaluation period lasting two weeks successfully identified main patterns but failed to show long-term reliability when combined with system drift and seasonal alterations in operational data.

5.5 Suggestions for Future Research

Different research opportunities stem from the study's results along with its known research constraints:

AI observability performance assessments will benefit from extensive field deployments extending beyond two weeks because such research will show complete operational patterns so teams can detect drifting behavior throughout lengthy time periods.

The study would gain a deeper understanding of observability needs if it expands interviews to include product managers and security engineers and compliance officers from different disciplines.

The research should investigate the process of integrating AI observability features with well-known commercial platforms including Datadog, New Relic, and AWS CloudWatch.

Research needs to make AI-based alerting and root cause recommendation systems more clear to engineers because it will build their trust and acceptance levels.

The investigation into real-time detection capabilities of AI-powered observability for compliance violations along with data exfiltration and model misuse will help security and compliance monitoring extend its advantages beyond operational performance.

6. CONCLUSION

6.1 Summary of Findings

A study examined how AI-powered observability functions to connect operational and technical hardware between DevOps and MLOps departments. The research combined both qualitative practitioner interviews with quantitative prototype evaluation results to uncover major monitoring problems which appeared primarily in systems with blending software and machine learning components.

Key findings include:

Production incidents occur because practitioners continuously struggle to see ML models within software systems and thus experience slow incident response times and detection delays.

AI observability tools boost anomaly detection accuracy by utilizing AI to accomplish automatic alert correlation which results in faster anomaly detection and resolution times alongside decreased alert notifications to end-users.

By applying domain-specific predictive analytics and contextual information the system can perform more comprehensive real-time monitoring operations.

The AI-supported method demonstrated better performance than traditional tools according to statistical evaluations that measured multiple parameters.

6.2 Final Thoughts

Modern operative environments need the immediate integration of DevOps principles with MLOps solutions because software systems now heavily depend on ML embedded components. Observability tools that operate traditionally prove successful with their domain tasks yet fail to deliver both functional domain insights and adaptive capabilities which match the production environment for dynamic ML systems.

The findings from this study prove that artificial intelligence-based observability tools exhibit both technical feasibility and operational business value. The integration of intelligent behavior analysis and diverse data stream correlation through these tools enables engineers to handle challenges better and minimize system outages while maintaining stable ML-enabled application performance.
Industry professionals endorse the implementation of intelligent observability solutions which demonstrate readiness to harmonize development speed with production reliability in engineering fields.

7. RECOMMENDATIONS

The resulting recommendations should be implemented by three target groups: industry stakeholders, engineering teams and future researchers.

The adoption of Advanced Observability Platforms by organizations is necessary to merge DevOps and MLOps data sets under one AI-analysis enabled interface.

System teams should buy predictive monitoring technology with built-in capabilities for anomaly prediction alongside contextual alert functions that help them identify and prevent system failures ahead of time.

Future observability solutions should address human design need by producing actionable insights with noise management capabilities while providing logical dashboards customized for developer and ML engineer and SRE roles.

Organizations must deploy AI observability solution testing in secure and non-business critical environments ahead of full deployment to assess their performance alongside existing workflows integration.

The teams at DevOps and MLOps should receive training and support to join forces on unified observability methods while breaking down past organizational barriers to develop joint operational responsibility.

The deployment of AI-powered observability systems creates a practical solution to unite DevOps and MLOps methodologies which could boost system dependability and developer performance as well as product quality during modern intelligent software periods.

DISCLAIMER (ARTIFICIAL INTELLIGENCE)

Author(s) hereby declare that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc) and text-to-image generators have been used during writing or editing of this manuscript.

COMPETING INTERESTS

Author has declared that no known competing financial interests or non-financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

Arora, V. (2024). *Exploring real-world challenges in MLOps implementation: a case study approach to design effective data pipelines* (Doctoral dissertation, University of Stuttgart).

Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2017, December). The ML test score: A rubric for ML production readiness and technical debt reduction. In 2017 IEEE international conference on big data (big data) (pp. 1123-1132). IEEE.

Chandra Thota, R. (2021). AI-driven infrastructure automation: Enhancing cloud efficiency with MLOps and DevOps.

Diaz-De-Arcaya, J., Torre-Bastida, A. I., Zárate, G., Miñón, R., & Almeida, A. (2023). A joint study of the challenges, opportunities, and roadmap of mlops and aiops: A systematic survey. *ACM Computing Surveys*, *56*(4), 1-30.

Johnson, W. (2023). AI-Augmented Observability: Transforming DevOps with Real-Time Insights and Automation.

Oluwaferanmi, A. (2025). Integrating MLOps and DataOps for Scalable and Resilient Machine Learning Deployment Pipelines: Challenges, Frameworks, and Best Practices.

Pahune, S., & Akhtar, Z. (2025). Transitioning from MLOps to LLMOps: Navigating the Unique Challenges of Large Language Models. *Information*, *16*(2), 87.

Pelluru, K. (2023). Advancing software development in 2023: the convergence of MLOps and DevOps. *Advances in Computer Sciences*, *6*(1), 1-14.

Vadde, B. C., & Munagandla, V. B. (2024). DevOps in the Age of Machine Learning: Bridging the Gap Between Development and Data Science. *International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence*, *15*(1), 530-544.

Xu, J. (2022). AI theory and applications in the financial industry. *Future And Fintech, The: Abcdi And Beyond*, *74*.

Peer-review history:
The peer review history for this paper can be accessed here:
https://pr.sdiarticle5.com/review-history/136550