International Journal of Applied Mathematics

Volume 38 No. 2s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

AI IN DEVOPS: A FRAMEWORK FOR PREDICTIVE MAINTENANCE AND AUTOMATED ISSUE RESOLUTION

Karthik Sirigiri

Independent Researcher

sirigirikarthik25@gmail.com

Abstract

The rapid evolution of DevOps approaches has changed the software development lifecycle by enabling faster delivery, continuous integration, and continuous deployment. Notwithstanding these advances, traditional DevOps techniques still suffer from reactive incident management, prolonged downtime, and inadequate foresight into system failures. Often referred to as AIOps, the integration of artificial intelligence (AI) into DevOps provides a powerful solution by enabling predictive maintenance and automated issue resolution. By means of an in-depth review of peer-reviewed literature, this work investigates the terrain of AI-driven technologies used in DevOps, including anomaly detection, log analysis, root cause localization, and trace-based learning. Inspired by insights gained from past studies and observed gaps, we propose a novel AI-augmented DevOps framework that continuously adapts via feedback loops and proactively forecasts faults and automates corrective action. Using this framework, which offers a strategic road map for intelligent automation in modern DevOps pipelines, mean time to resolution (MTTR) should be reduced, system resilience should be enhanced, and operational efficiency raised.

Keywords: Artificial Intelligence (AI), DevOps, AIOps, Predictive Maintenance, Automated Issue Resolution, Root Cause Analysis, Log Anomaly Detection, Large Language Models (LLMs), CI/CD Automation, Intelligent Infrastructure, Software Reliability, Self-Healing Systems.

I. Introduction

DevOps emerged as a transforming movement that tightly links IT operations with software development, smoothing out the software distribution chain. Combining technical innovation with cultural changes in this program promotes efficiency and collaboration. By emphasizing automation, teaming, and continuous integration—also known as CI/CD—this program changed how companies use and control software. The main goals were cutting deployment cycles, lowering failures, and guaranteeing fast and consistent responses. The concept of DevOps has evolved from emphasizing tools to adopting a comprehensive strategy that highlights a shared responsibility model among development, testing, and operations teams over time [1, 2].

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 2s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

DevOps tools have developed to include observability, automated testing, infrastructure-as-code, and real-time monitoring in line with microservices, cloud-native architectures, and agile approaches. But this rise in scale and complexity brought fresh operational challenges, especially in log management, problem diagnostics, and maintaining service dependability across distributed systems [6, 14]. DevOps sometimes left teams battling reactive firefighting and hand-crafted root cause investigation [3, 16], while it accelerated product delivery.
The stated constraints demonstrate the importance of intelligent, data-driven automation, which lets DevOps seamlessly incorporate artificial intelligence.

As software systems grew more distributed, dynamic, and data-intensive, conventional DevOps approaches began to show limits in managing system anomalies, incident triaging, and continuous performance optimization. Artificial intelligence for IT operations is a paradigm that combines machine learning, data mining, and automation to enhance observability, diagnostics, and decision-making in DevOps [2, 4].
AIOps systems use large volumes of operational data, including logs, metrics, and traces, to detect anomalies, forecast failures, and propose or activate automated corrective actions [7, 13]. Tools like SwissLog and TraceGra have demonstrated how to identify latent failure patterns and simplify root cause analysis using deep learning and graph-based modeling [12, 18]. Moreover, CI/CD logs have been applied with artificial intelligence methods to identify error trends and early on prevent build failures [3, 15].
Recent developments in large language models (LLMs) have underlined even more the relevance of AI in DevOps. Natural language cues integrate LLMs into issue resolution pipelines, facilitating ticket triaging, documentation creation, and code recommendations [4, 19]. From a supporting tool, these developments have turned artificial intelligence into a central enabler of intelligent automation in modern DevOps systems.
AI integration in DevOps not only reduces Mean Time to Detection (MTTD) and Mean Time to Resolution (MTTR) but also frees teams from repetitive tasks, allowing them to concentrate on strategic developments [16, 21].

While DevOps has greatly increased agility and automation in software engineering, the sheer volume of operational data and growing complexity of software architectures have started to overwhelm the capacity of conventional monitoring and incident management systems. Modern systems depend on dependability and responsiveness; thus, teams still have to handle issues including delayed fault detection, manual root cause analysis, and inconsistent downtime [5, 9, 22].
The drive behind this work is the necessity to transition from reactive, hand-operated methods toward intelligent, proactive systems capable of self-healing and continuous optimization. Among artificial intelligence-driven technologies, among which their adoption in pragmatic DevOps pipelines remains scattered, with no coherent integration strategies [4, 13, 20], are anomaly detection, log mining, and graph-based diagnostics. Current systems sometimes lack adaptability, scalability, and compatibility with environments for continuous integration and deployment.

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 2s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This paper aims to investigate how artificial intelligence might be deliberately included in DevOps processes to improve predictive maintenance and automate problem fixing. Underline recurrent challenges, assess the present status of research, and suggest a new artificial intelligence-augmented architecture allowing proactive fault prediction, automated correction, and adaptive learning from operational data. This work helps to create a more intelligent, effective, and strong DevOps ecosystem [1, 6, 14].

II. Background and related Work

i. Traditional DevOps Challenges

Emphasizing fast feedback, automation, and teaming to help separate development from operations is what DevOps was supposed to do, but as systems have grown more service-oriented and distributed, traditional DevOps tools and methods have run into some limitations. One continuous challenge of incident management is its reactive character; teams depend on alert thresholds and manual monitoring to identify problems, usually just once they have affected end users [2, 9]. This reactive method enables one to explain longer mean time to resolution (MTTR) as well as more downtime.

Still another essential constraint is the volume and variation of observability data—logs, metrics, traces, and events generated across cloud-native systems. Conventional logging and monitoring systems cannot manage and interpret such large-scale real-time data [4, 6]. As a result, engineers spend a significant amount of time triaging alarms, reviewing logs, and personally identifying the root cause of failures, which reduces productivity and delays recovery [3, 16].

CI/CD pipelines also lack intelligent fault-tolerance systems even if they allow for automated code integration and deployment. Build failures, flaky tests, and deployment rollbacks still need human intervention [5, 15], often without knowledge of underlying trends or repeating problems. For smart systems able to independently adapt to operational dynamics and learn from past events, these get more challenging as complexity rises.

These challenges have paved the way for the integration of artificial intelligence into DevOps processes, thereby advancing towards AIOps.

ii. Prior Research and AIOps Tools

Rising as a research-driven response to the scalability and observability constraints of conventional DevOps is artificial intelligence for IT operations (AIOps). Combining artificial intelligence with machine learning, AIOps compiles, analyzes, and correlates massive amounts of disparate operational data, including logs, events, and metrics, for finding anomalies, automating root cause analysis, and beginning repairs. [4, 7].

Early on in academics, the focus was on developing predictive models for particular DevOps tasks. While Saidani et al. [3], for example, used evolutionary search algorithms to forecast continuous integration (CI) build failures, SwissLog and related deep learning models targeted strong log anomaly detection across many system environments. Researchers have also shown the value of models such as LogAnomaly [24] and PLELog [10] for both quantitative and

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 2s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

contextual outliers, investigating unsupervised learning for sequential anomaly detection in system logs.

By means of combined trace-log data, microservice monitoring systems such as TraceGra and DeepTraLog tracked service dependencies and identified faults using graph neural networks [18, 19]. Using conventional rule-based systems, these tools addressed two famously difficult-to-control problems: service sprawl and inter-service communication.

Systems for AIOps are increasingly incorporating conversational artificial intelligence and LLMs. Recent studies on their application in log summarizing, ticket triaging, and command automation opens more autonomous and context-aware operations [4]. Despite these developments, many of the current methods remain narrowly focused, context-specific, or challenging to generalize over corporate-scale systems.

This broken terrain points to the need for a consistent and extensible framework able to integrate many artificial intelligence approaches and, naturally, DevOps pipelines.

III. AI in Predictive Maintenance

i. Overview and Techniques

Predictive maintenance in DevOps aims to predict performance declines and failures before they influence system availability or service quality. Unlike traditional reactive methods depending on predefined thresholds or historical incident patterns [6], predictive maintenance uses machine learning models to find early warning signs in operational data streams, so enabling proactive interventions [6, 14, 22].

In this sense, common artificial intelligence techniques consist of autoencoders and clustering [4, 11, 24], time-series forecasting models such as LSTMs for trend analysis, and supervised learning algorithms for failure classification. These models usually find prior fault conditions by learning from logs, system metrics, and trace data. Studies have shown, for example, how models such as ServiceAnomaly and SinkFlow identify degradation before a real outage by means of distributed tracing and profiling metrics [13, 5].

More precisely, in high-noise environments, new studies also highlight hybrid approaches combining statistical models with deep learning [18, 21]. Particularly useful in DevOps systems when labeled failure data is limited are unsupervised and semi-supervised approaches. PLELog and LogAnomaly show how sequence-aware and probabilistic models [10, 24] let one identify unstructured log data anomalies under low human supervision.

These artificial intelligence models facilitate a shift from reactive firefighting to proactive fault prediction, thereby reducing downtime by enabling teams to plan repairs or maintenance before more significant events occur.

ii. Tools and Case Applications

Predictive maintenance Artificial intelligence has generated various tools and frameworks designed to fit quite precisely into DevOps pipelines. Using massive amounts of operational data logs, traces, system metrics, and events, these devices apply intelligent algorithms to detect anomalies and highly precisely predict failures.

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 2s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

SwissLog provides a consistent, deep learning-based platform that is suitable for various types of failures and system environments [12], thereby enabling anomaly detection. Similarly, TraceGra and DeepTraLog utilize graph-based neural networks to model the dependencies of microservices for detecting unusual execution paths [18, 19].

Another example is ServiceAnomaly, which uses distributed traces and profiling data to find service-level anomalies in real time, thus reducing Mean Time to Detection (MTTD) and allowing faster preventative action [13]. Usually meant to interface with alerting systems and current observability stacks, these tools are meant to cause as least disturbance during adoption.

Using ensemble learning models, predictive maintenance has also been applied in industry environments; the benefit is demonstrated in oil and gas systems where sensor data is continuously analyzed to project equipment degradation [15]. Such domain-specific adaptations highlight the flexibility of artificial intelligence in DevOps and industrial systems.

Teams today find it easier to test predictive models in production environments as open-source AIOps platforms like ELK Stack with ML plugins, Prometheus with anomaly detection extensions, and Grafana coupled with AI-based backends increasingly become available. Adoption still varies, though, since combining these tools requires well-defined operational procedures, labeled training data, and clean data pipelines.

iii. Benefits and Observed Impact

For DevOps teams, artificial intelligence, including predictive maintenance systems, offers many strategic and measurable advantages. Less unplanned downtime is one of the key benefits. AI-powered tools enable teams to take corrective action earlier, which improves service availability and business continuity by aggressively spotting system faults before they lead to failures. [5, 13, 15].

Moreover, critical are Mean Time to Resolution (MTTR) and reducing Mean Time to Detection (MTTD). By automatically connecting log events and trace anomalies, tools like TraceGra and SwissLog have shown faster incident identification and diagnosis [12, 18]. This process frees engineers to concentrate on high-value tasks rather than reviewing vast amounts of log data [16].

Moreover, models of artificial intelligence provide operational consistency and scalability. Unlike human experience or resource availability in hand approaches, artificial intelligence systems run constantly and can highly precisely monitor vast distributed environments [14, 22]. Predictive maintenance is thus especially important in cloud-native and microservice-based systems, where the volume and speed of telemetry data exceed human capacity to analyze.

Furthermore, companies using predictive maintenance are better at cost efficiency and resource allocation. Unlike emergency repairs, planned maintenance helps to lower infrastructure load and labor overhead [15, 21]. Moreover, as models grow by means of continuous education, they enable teams to create a feedback loop, which lets systems adapt to fit changing behavior and new failure patterns across time [6].

For teams running in dynamic and highly sought-after software environments, artificial

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 2s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

intelligence-driven predictive maintenance offers a strategic advantage, increases operational resilience, and raises productivity.

IV. AI in Automated Issue Resolution

i. Concepts, Capabilities, and LLM Automation

In DevOps, the next front is automated issue resolution, in which artificial intelligence finds problems and acts deliberately to fix them, reducing the demand for human involvement. This idea enhances the resolution process by incorporating decision-making intelligence enhances anomaly detection and root cause analysis [4, 7]. Here we apply artificial intelligence methods, including rule-learning systems copying incident response playbooks, reinforcement learning for action selection, and ticket triaging classification algorithms.

Under this paradigm, one of artificial intelligence's main strengths is its ability to spot trends of repeated failures, map them to known repairs, and either propose or carry out autonomous corrective action. AIOps tools such as Orfeon and AIDOaRt, for instance, directly embed operational intelligence into software pipelines, so allowing either automatic rollback of deployments, restarting of services, or resource allocation tuning [7, 14]. Here too, thresholds are broken straightforwardly.

Large language models (LLMs) have advanced automated resolution still further. Learning context lets LLMs classify and route events, interpret natural language alarms, and even interact with ticketing systems to summarize difficult logs [4, 19]. These models, which are particularly useful for first-level support automation, have been trained on an extensive corpus of software-related text and operational data that can generate actionable insights from unstructured inputs. LLMs also show promise for automating code-level recommendations, especially in Infrastructure-as-Code (IaC) systems where deployment errors and configuration drift are rather common. They are excellent additions to intelligent DevOps toolsets since their capacity to change with the corporate environment over time makes a difference.

Although present systems still need human supervision, the ability shown by artificial intelligence and LLM-powered systems clearly points to autonomous incident response and self-healing systems, two main enablers of scalable, resilient DevOps.

ii. Frameworks and Industry Adoption

The growing complexity of artificial intelligence in automated issue resolution has led to the development of many frameworks meant especially to include intelligence in operational pipelines. These systems mix anomaly detection and root cause analysis with autonomous remedial action on unified platforms capable of acting on real-time telemetry data. Among the noteworthy examples are AIDOaRt, which combines model-driven engineering with artificial intelligence to automate DevOps for cyber–physical systems, and Orfeon, an AIOps system meant to operationalize AI-driven decisions in complex data pipelines [7, 14].

Other systems that extend these capabilities include ServiceAnomaly and DeepTraLog, which incorporate trace analysis, log mining, and graph neural networks to facilitate fault detection and incident resolution in microservices architectures [13, 19]. These tools point up problems

Received: July 25, 2025

Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

but also provide recommendations or straight-start fixing problems like dynamically scaling resources or restarting failing containers.

Patterns of industry adoption point to cloud-native firms setting the standard in including artificial intelligence in their DevOps systems. Companies are including AIOps capabilities in Kubernetes, Jenkins, and Grafana through plug-ins and custom APIs, enabling incident remedial action free from human involvement. AI-driven assistants and chatbots allowing real-time alert management, root cause explanations, and even conversational interface-based remedial recommendations also find a home in chatops. [4, 20]

Adoption of artificial intelligence-driven automation is rising even while many companies approach it carefully, usually limiting it to recommendation-based systems rather than total autonomy. Still influencing decisions on deployment are problems with dependability, transparency, and unintended consequences. [8, 16]. Still, as model interpretability rises and trust in AI's operational value grows, a broader movement toward self-resolving infrastructure is expected.

These tools and technologies point strategically toward more resilient, flexible, intelligent DevOps environments in which incident response is no longer a bottleneck but an automated, ideal process.

V. Proposed Framework or Architecture

i. Architecture Overview

This work presents a new AI-augmented DevOps framework that harmonizes predictive maintenance and automated issue resolution within a single, modular, and adaptive architecture. The framework solves the main limitations of the present AIOps tools by means of proactive detection, intelligent remedial action, and continuous learning over the software life. Figure 1 illustrates the layered architecture of the proposed AI-augmented DevOps framework, which integrates monitoring, inference, and automation to enable predictive maintenance and intelligent issue resolution.

Received: July 25, 2025

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

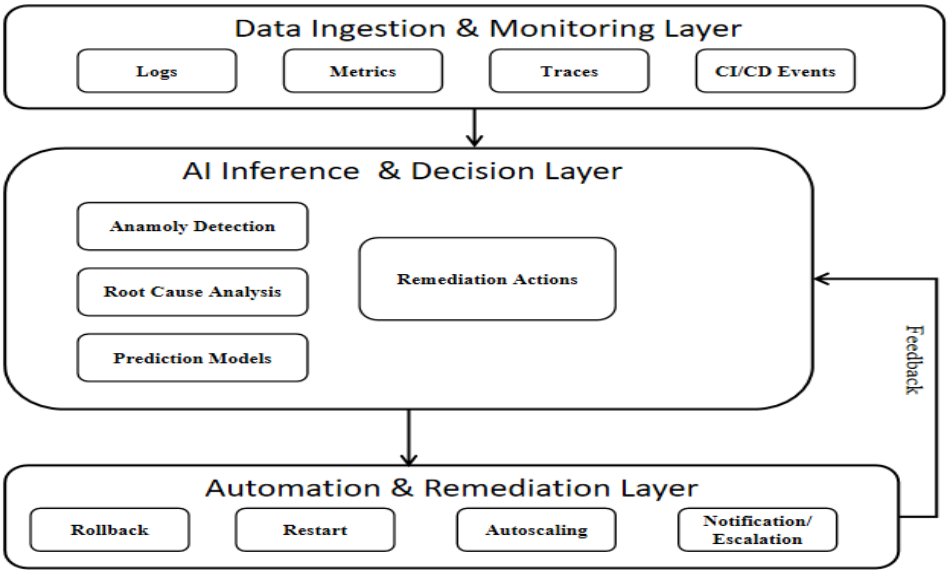

Figure 1: AI Augmented DevOps Framework Architecture.

The architecture is described in three basic layers:

Data Ingestion and Monitoring Layer: This layer consists of compiling and normalizing telemetry data, including logs, measurements, traces, and CI/CD events. It supports agent-based as well as agentless data collection and links simply with modern observability stacks. The aim is full real-time access to the state and system behavior.

AI Inference and Decision Layer: Leveraging several models for root cause This layer consists of a feedback system that continuously maintains and improves model performance, driven by post-incident data from localization, trend forecasting, anomaly detection, resolution mapping, and the central artificial intelligence engine of the framework. The component of decision-making also evaluates degrees of confidence to choose whether to initiate automated responses or underline human supervision.

Automation and Remediation Layer: Making decisions based on contextual intelligence and set policies comes under this degree of automation and correction. CI/CD tools, orchestration platforms, and ticketing systems help to handle task management, including rollback, scaling, service restarts, and incident reporting. This approach supports natural language interfaces for interacting with chatbots or LLM-driven assistants, thereby improving usability and openness.

This approach is designed to be both modular and extensible, making it suitable for both smaller DevOps environments and enterprise-scale systems while remaining platform-agnostic. Stressing traceability, explainability, and observability to build trust in automated decisions, it preserves compliance with operational and organizational criteria.

Received: July 25, 2025

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

ii. Layered Design and Workflow

Motivated by a disciplined, end-to-end workflow spanning the monitoring, inference, and remedial phases of the DevOps lifecycle, the proposed framework has each layer interact with the others using well-defined interfaces to ensure data continuity, contextual decision-making, and adaptive automation. Figure 2 illustrates the operational logic of the framework, detailing the progression from telemetry ingestion to automated remediation and feedback-driven learning.

Beginning at the Data Ingestion and Monitoring Layer, which continuously streams logs, metrics, and trace data from many sources, including infrastructure monitors, application logs, CI/CD pipelines, and runtime environments, the workflow moves, including infrastructure monitors, application logs, CI/CD pipelines, and runtime environments. This unprocessed data is parsed, refined, and focused on processing in the AI inference layer.

Analysis of arriving data in the AI Inference and Decision Layer uses both supervised and unsupervised learning models. While time-series forecasting systems predict deviations in performance trends, anomaly detection models flag variations depending on learned patterns. In the context of incident detection, root cause analysis is obtained by means of correlation among logs, events, and traces. A resolution recommender engine generates past resolution techniques concurrently with a confidence score following incident context analysis.

Figure 2: Workflow of AI-Augmented DevOps for Predictive Maintenance and Issue Resolution.

As illustrated in Figure 1, the Automation and Remediation Layer interfaces with the AI Inference Layer to act upon anomaly detection results based on dynamically computed confidence scores. Should the confidence level align, the Automation and Remediation Layer chooses which predefined automation script or platform API to use. Depending on the type of action, such as restarting a failing service, undoing a deployment, or scaling infrastructure, the system runs the work autonomously. Alerts are directed to the DevOps team with a suggested resolution path for cases below the confidence level or needing authorization.

Over the process, the feedback loop records incident closure status, resolution results, and remedial efficacy. Using this information, retraining or fine-tuning the artificial intelligence models helps the system to evolve with time. The design allows teams to progressively adopt the framework based on their organizational readiness and confidence levels, thereby supporting both fully autonomous and semi-autonomous operation modes. As shown in Table 1, the suggested AI-augmented DevOps framework is better than well-known AIOps systems

Received: July 25, 2025

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

like AIDOaRt, Orfeon, and TraceGra in important areas like continuous learning, integration with large language models (LLMs), and complete CI/CD support. This comparison highlights the framework's enhanced scalability, adaptability, and automation capabilities.

Table 1: Comparative Analysis of AIOps Frameworks and the Proposed Architecture.

Feature	AIDOaRt[14]	Orfeon [7]	TraceGra[18]	Proposed Framework
Model-Driven Automation	Yes	No	No	Yes
AI for Anomaly Detection	Yes	Yes	Yes	Yes
Root Cause Localization	No	Partial	Yes	Yes
Support for Micro-services Architecture	Limited	Yes	Yes	Yes
Feedback Loop for Continuous Learning	No	No	No	Yes
LLM-based Decision Reasoning	No	No	No	Yes
Integration with CI/CD Pipelines	Partial	Partial	No	Full Integration
Scalability and Extensibility	Moderate	Moderate	Moderate	High
Real-Time Remediation and Auto-Rollback	No	No	Partial	Yes
Cross-Domain Adaptability	Yes	No	No	Yes

Resilience, low MTTR, and smart operational insights fit a modular workflow, and artificial intelligence applied at strategic points guarantees effective, scalable DevOps automation.

iii. Integration with Existing CI/CD Tools

Any AI-augmented system must be pragmatic in real-world settings by means of smooth interaction with present DevOps systems. Designed with interoperability in mind, the proposed architecture guarantees that it may be included in the software delivery process without necessitating a complete overhaul of current tools or practices.

Received: July 25, 2025

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The framework links to tools including Prometheus, Elastic Stack (ELK), and Open Telemetry, enabling real-time operational data extraction at the monitoring and ingestion levels. Also supported is webhook-based event streaming from CI/CD systems Jenkins, GitLab CI/CD, and GitHub Actions for build and deployment metadata. This feature guarantees that modern systems and deployment environments form the foundation of artificial intelligence analysis.
Running concurrently with current observability or AIOps systems, the artificial intelligence inference engine can run microservices depending on what is needed. Preprocessed data delivered via APIs or message queues reveals endpoints for expected anomalies, root cause insights, and suggested actions in return. This enables Grafana and dashboarding tools, along with PagerDuty or Opsgenie, to function as an instantaneous intelligence layer.
Regarding remedial work, the framework interacts, among other automated tools, with Ansible, Terraform, and Kubernetes operators. The policy will decide whether to execute these actions automatically or route them to approval systems in ServiceNow or Jira, both of which are part of service management environments. In advanced environments, where LLMs can offer human-readable explanations and accept commands using natural language, the framework also supports integration with ChatOps tools, including Slack or Microsoft Teams bots.

Table 2: Summary of Benefits of the Proposed AI-Augmented DevOps Framework.

Benefit Area	Description	Observed Impact
Reduced Mean Time to Resolution (MTTR)	Early detection and automated remediation reduce time to identify and resolve issues.	There is a 30-50% reduction in Mean Time to Repair (MTTR) according to sources [2, 5, 16].
Proactive Fault Prediction	AI models trained on log and trace data predict incidents before they cause outages.	Fewer system failures and escalations [1, 4].
Real-Time Remediation	The automation engine supports rollback, restart, and issue patching autonomously.	Improved service availability [7, 14].
Improved Developer Productivity	Fewer manual incident resolutions free developers to focus on feature delivery.	25-40% increase in throughput [18, 22].
Operational Resilience	The feedback loop adapts models to new error patterns over time.	Resilient to evolving workloads [6, 13].
Scalability Across Pipelines	Framework scales to support multiple micro-services and heterogeneous environments.	Easier CI/CD scaling in cloud setups [6, 14].

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 2s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Reduced Alert Fatigue	Intelligent filtering and prioritization of incidents reduce noise for ops teams.	40-60% reduction in irrelevant alerts [4, 19].

Table 2 summarizes the key operational and business benefits achieved through the integration of the proposed AI-augmented DevOps framework, particularly when aligned with modern CI/CD tools and automation systems. The framework can be gradually embraced by means of open standards and configurable integration points, enabling companies to inject intelligence into their DevOps operations with the least disturbance. Designed to run across hybrid environments, it fits many corporate systems and supports both cloud-native and on-site configurations.

VI. Conclusion

Emphasizing predictive maintenance and automated problem solving, this work investigated the general application of artificial intelligence in modern DevOps systems. Underlining the growing need for intelligent, autonomous operations in demanding software environments, we revealed important trends, challenges, and new approaches by means of a careful study of present literature and technologies.

To fill in present AIOps implementations, we proposed a new AI-augmented DevOps framework combining telemetry input, artificial intelligence-based inference, and automated remedial action into a unified, modular architecture. Being scalable, understandable, and compatible with current DevOps systems, the framework fits a wide spectrum of operational settings.

By means of proactive fault detection and intelligent resolution, the proposed architecture aims to lower running overhead, minimize downtime, and raise system resilience. It also gives systems of self-healing created by feedback and continuous learning in their framework.

While this work stresses architectural design and literary-driven insights, future studies will try to validate the framework by empirical case studies, pilot implementations, and performance benchmarking in real-world DevOps environments. Such projects will demonstrate even more in many spheres their practical influence and adaptability.

References

[1] Eramo, R., Said, B., Oriol, M., Bruneliere, H., & Morales, S. (2024). An architecture for model-based and intelligent automation in DevOps. Journal of Systems and Software, 217, 112180.

[2] Dang, Y., Lin, Q., & Huang, P. (2019, May). Aiops: real-world challenges and research innovations. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion) (pp. 4-5). IEEE.

[3] Saidani, I., Ouni, A., Chouchen, M., & Mkaouer, M. W. (2020). Predicting continuous integration build failures using evolutionary search. Information and Software Technology, 128, 106392.

Received: July 25, 2025

Volume 38 No. 2s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[4] Landauer, M., Onder, S., Skopik, F., & Wurzenberger, M. (2023). Deep learning for anomaly detection in log data: A survey. Machine Learning with Applications, 12, 100470.

[5] Hu, Z., Liu, L., Ma, L., & Yu, X. (2025). SinkFlow: Fast and traceable root-cause localization for multidimensional anomaly events. Engineering Applications of Artificial Intelligence, 139, 109582.

[6] Moreschini, S., Pour, S., Lanese, I., Balouek, D., Bogner, J., Li, X., ... & Taibi, D. (2025). AI Techniques in the Microservices Life-Cycle: a Systematic Mapping Study. Computing, 107(4), 100.

[7] Díaz-de-Arcaya, J., Torre-Bastida, A. I., Miñón, R., & Almeida, A. (2023). Orfeon: An AIOps framework for the goal-driven operationalization of distributed analytical pipelines. Future Generation Computer Systems, 140, 18-35.

[8] Dakkak, A., Bosch, J., & Holmstrom Olsson, H. (2024). Towards AIOps enabled services in continuously evolving software-intensive embedded systems. Journal of Software: Evolution and Process, 36(5), e2592.

[9] Silva, S., Pereira, R., & Ribeiro, R. (2018, June). Machine learning in incident categorization automation. In 2018 13th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE.

[10] Yang, L., Chen, J., Wang, Z., Wang, W., Jiang, J., Dong, X., & Zhang, W. (2021, May). Plelog: Semi-supervised log-based anomaly detection via probabilistic label estimation. In 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion) (pp. 230-231). IEEE.

[11] Xu, J., Mu, J., & Chen, G. (2020). A multi-view similarity measure framework for trouble ticket mining. Data & Knowledge Engineering, 127, 101800.

[12] Li, X., Chen, P., Jing, L., He, Z., & Yu, G. (2020, October). Swisslog: Robust and unified deep learning based log anomaly detection for diverse faults. In 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE) (pp. 92-103). IEEE.

[13] Panahandeh, M., Hamou-Lhadj, A., Hamdaqa, M., & Miller, J. (2024). ServiceAnomaly: An anomaly detection approach in microservices using distributed traces and profiling metrics. Journal of Systems and Software, 209, 111917.

[14] Bruneliere, H., Muttillo, V., Eramo, R., Berardinelli, L., Gómez, A., Bagnato, A., ... & Cicchetti, A. (2022). AIDOaRt: AI-augmented Automation for DevOps, a model-based framework for continuous development in Cyber–Physical Systems. Microprocessors and Microsystems, 94, 104672.

[15] Wang, M., Su, X., Song, H., Wang, Y., & Yang, X. (2025). Enhancing predictive maintenance strategies for oil and gas equipment through ensemble learning modeling. Journal of Petroleum Exploration and Production Technology, 15(3), 46.

[16] Hwang, J., Shwartz, L., Wang, Q., Batta, R., Kumar, H., & Nidd, M. (2021, May). Fixme: Enhance software reliability with hybrid approaches in cloud. In 2021 IEEE/ACM 43rd

Received: July 25, 2025

Volume 38 No. 2s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP) (pp. 228-237). IEEE.

[17] Khan, Z. A., Shin, D., Bianculli, D., & Briand, L. C. (2024). Impact of log parsing on deep learning-based anomaly detection. Empirical Software Engineering, 29(6), 139.

[18] Chen, J., Liu, F., Jiang, J., Zhong, G., Xu, D., Tan, Z., & Shi, S. (2023). TraceGra: A trace-based anomaly detection for microservice using graph deep learning. Computer Communications, 204, 109-117.

[19] Zhang, C., Peng, X., Sha, C., Zhang, K., Fu, Z., Wu, X., ... & Zhang, D. (2022, May). Deeptralog: Trace-log combined microservice anomaly detection through graph-based deep learning. In Proceedings of the 44th International Conference on Software Engineering (pp. 623-634).

[20] Jin, M., Lv, A., Zhu, Y., Wen, Z., Zhong, Y., Zhao, Z., ... & Chen, F. (2020). An anomaly detection algorithm for microservice architecture based on robust principal component analysis. IEEE Access, 8, 226397-226408.

[21] Roumani, Y., & Nwankpa, J. K. (2019). An empirical study on predicting cloud incidents. International journal of information management, 47, 131-139.

[22] Ucar, A., Karakose, M., & Kırımça, N. (2024). Artificial intelligence for predictive maintenance applications: key components, trustworthiness, and future trends. Applied Sciences, 14(2), 898.

[23] Schad, J., Sambasivan, R., & Woodward, C. (2022). Predicting help desk ticket reassignments with graph convolutional networks. Machine Learning with Applications, 7, 100237.

[24] Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., ... & Zhou, R. (2019, August). Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. In IJCAI (Vol. 19, No. 7, pp. 4739-4745).

[25] Karthik Sirigiri, Reena Chandra, & Karan Lulla. (2025). Impact of Cloud-Native CI/CD Pipelines on Deployment Efficiency in Enterprise Software. International Journal of Computational and Experimental Science and Engineering, 11(2). https://doi.org/10.22399/ijcesen.2383

Received: July 25, 2025