Contents available at the publisher website: GAFTIM.COM

International Journal of Technology, Innovation, and Management (IJTIM)

Journal homepage: https://journals.gaftim.com/index.php/ijtim/index

Exploring the Impact of Artificial Intelligence on the Evolution of Observability Tools

Norah Sanad Alsubaie[1*], *Ohoud Mousa Alharbi*[2]

[1,2]*King Saud University, Riyadh, Saudi Arabia*

**Corresponding Author*

ARTICLE INFO

Keywords:
Observability Tools, Artificial Intelligence, AIOps, Anomaly Detection, IT Operations, Qualitative Study

Received: Jul, 20, 2025
Accepted: Oct, 23, 2025
Published: Dec, 25, 2025

ABSTRACT

As modern software systems become more complex with the rise of cloud computing, containers, and distributed services, traditional monitoring methods have become increasingly inadequate for managing and observing these systems. This study investigates the transformative impact of artificial intelligence (AI) on observability tools, focusing on the perspectives of IT professionals who work with these technologies. Observability tools, which enable IT teams to understand system behavior through logs, metrics, and traces, are essential for effective system management. However, AI integration is changing how these tools detect anomalies, predict failures, and improve operational efficiency. Despite the promising potential, challenges remain, particularly regarding the effectiveness of AI-powered tools in real-world environments. Using semi-structured interviews with 15 IT professionals, this study explores their experiences with AI-integrated observability tools, focusing on the impact on daily workflows, incident detection, operational efficiency, and the challenges practitioners face. The findings highlight the distinction between basic and advanced AI features, the role of external AI tools like ChatGPT and Copilot in enhancing observability, the trust gap in AI-generated alerts, and the critical importance of data quality for effective AI functionality. This research contributes valuable insights into the practical implications of AI in observability and provides recommendations for both organizations and tool developers to improve AI adoption and effectiveness. Future research should further explore how AI adoption evolves within different industries and investigate the long-term impact of AI-powered observability tools.

1. INTRODUCTION

In recent years, organizational infrastructure has evolved significantly, becoming heavily dependent on containers, cloud computing, and distributed services . This evolution in software systems has made them challenging to monitor and manage, generating vast amounts of data that are difficult to analyze using traditional monitoring methods. In contrast, with the development of artificial intelligence and machine learning technologies, new possibilities have emerged to address these challenges through their ability to process big data, discover complex patterns, and predict problems before they occur. These developments in AI and ML technologies have particularly impacted the observability tools that organizations rely on according to Cheng et al. (2023).

Observability tools are essential components in managing modern software systems, as they enable IT teams to understand the internal state of the system by analyzing three core outputs: logs, metrics, and traces. Unlike traditional monitoring tools that are limited to notifying the team about the problems, observability tools reveal the root causes of problems and provide insights into the behavior of distributed systems. These tools have therefore become critical for ensuring rapid response to failures, improving performance, and making decisions based on accurate and comprehensive data. Observability tools also contribute to reducing the time required to detect and resolve problems, which has a positive impact on service stability and the end-user experience according to Shen et al., (2020).

As distributed systems continue to grow in complexity and data sources become increasingly diverse, the integration of artificial intelligence with observability tools has become a fundamental requirement to overcome the challenges imposed by traditional monitoring methods. This integration enables the transformation of raw data from logs, metrics, and traces into actionable insights by applying machine learning techniques to automatically detect anomalies, predict failures before they occur, and identify root causes of issues with greater accuracy and speed. AI also helps reduce false alarms that burden operations teams and automates incident response, enhancing system management efficiency and improving reliability. This integration of AI and observability tools represents a major shift in how software systems are observed and managed in modern work environments according to Carnevale et al. (2025).

Despite the significance of this integration, previous studies on observability tools have lacked in-depth analysis of practitioner's actual experiences. These studies primarily focused on technical and commercial comparisons or limited application contexts, which restricts our understanding of the real-world impact of these tools. Furthermore, most studies have focused on how practitioners utilize AI capabilities in real work environments, leaving a gap in understanding the actual impact of this integration from the perspective of professionals Biswas (2025).

Therefore, this study aims to explore how artificial intelligence is transforming observability tools from the perspective of IT professionals by investigating the practical impact of AI features on monitoring workflows, understanding practitioners' experiences across diverse roles and organizations, and identifying the benefits and challenges of implementing these technologies. The significance of this study lies in helping organizations make informed adoption decisions, enriching academic literature with qualitative insights into practitioners' real-world experiences, and informing future tool development based on actual user needs.

To achieve this goal, we seek to answer the following research questions (RQs):

- What observability tools are IT professionals currently using, and do these tools incorporate AI capabilities?
- How do AI features impact daily monitoring workflows, incident detection, and operational efficiency?
- What challenges and limitations do practitioners encounter when using AI-powered observability tools?

2. LITERATURE REVIEW

2.1 Foundational Concepts

To understand how AI transforms observability tools, it is essential to first establish clear definitions of monitoring and observability, as these terms are often used interchangeably despite their distinct meanings and purposes in system management.

2.1.1 Monitoring:

The process of supervising and verifying the operation of a system or one of its components, including recording and analyzing data to ensure that performance is as expected Parvathinathan (2025). In practice, traditional monitoring relies on a static approach and predefined metrics, such as processor and memory utilization rates and network status, where these values are compared to certain thresholds to trigger alerts when exceeded. This approach is suitable for traditional, stable systems, but it often fails to provide sufficient visibility to detect problems in modern dynamic environments such as cloud computing and distributed systems.

2.1.2 Observability:

The ability to understand the internal state of a system by analyzing its external outputs, such as logs, metrics, and traces. This capability provides engineers with deep insights into why a system behaves in a certain way, enabling them to diagnose issues and uncover root causes in complex, dynamic environments like cloud-native applications and microservices architectures. Unlike traditional monitoring, which focuses on predefined metrics, observability emphasizes comprehensive visibility and context, allowing proactive problem-solving and performance optimization Siddiqui et al. (2023).

To summarize the key distinctions between monitoring and observability, Table 1 provides a comparative overview:

Table 1: Comparison Between Monitoring and Observability

Dimension	Monitoring	Observability
Definition	Track system health and performance metrics, alert on anomalies	Understand the internal state of the system and the reasons behind its behavior
Key Questions	What, where, and when did the issue occur?	Why did the issue occur?
Data Used	Metrics only	Logs, Metrics, and Traces
Primary Goal	Detect failures	Identify root causes and optimize performance

2.2 The Three Pillars of Observability:

As mentioned earlier, observability depends on external outputs, which consist of 3 main components [9]:

2.2.1 Logs:

Logs are timestamped messages that record discrete events occurring within a system over time. They can be stored in various formats, including plain text, structured formats such as JSON, or binary for specialized purposes. In practice, logs capture critical information about errors, transactions, and system state changes, providing an ordered record of activities that enables anomaly detection, root cause analysis, and improved system reliability Mahida (2023).

2.2.2 Metrics:

Metrics are quantitative measurements collected over time that provide insights into system health (eg., CPU, memory, network throughput. They are the foundation for alerting solutions that can inform teams about present or future failures in the system. Moreover, metrics represent the main component of monitoring systems, enabling organizations to build dashboards to proactively analyze performance and plan capacity Mart et al., (2020).

2.2.3 Traces

Traces capture detailed information about requests as they move through systems, recording each service involved, response times, and how services depend on each other . This visibility helps organizations identify performance bottlenecks, optimize request flows, and find the root causes of issues. Traces are especially useful in complex systems where understanding how services interact is essential for troubleshooting Tatineni (2023).

2.3 An Overview of Observability Tools

Building on the three pillars of observability discussed above, various tools have been developed to implement these concepts in practice. Observability tools are categorized into several main groups, including log management solutions (e.g., Logstash, Elasticsearch, Splunk), distributed tracing tools (e.g., Jaeger, Zipkin), Application Performance Monitoring (APM) platforms (e.g., AppDynamics, Dynatrace, New Relic), and infrastructure monitoring tools (e.g., Prometheus, Datadog, Azure Monitor).

These tools leverage techniques such as distributed tracing, anomaly detection, and predictive analytics to provide deep insights into application behavior by analyzing metrics like response times, error rates, and resource utilization. A multi-tool strategy offers benefits including flexibility, redundancy, and comprehensive coverage, though organizations face implementation challenges such as high licensing costs, steep learning curves, and complexities in integrating with legacy systems Faseeha et al. (2025).

2.4 Traditional Observability Approaches and Their Limitations

Most current monitoring tools rely on simple anomaly detection methods, specifically threshold-based (comparing values to fixed limits) and baseline-based (comparing values to historical baseline). These conventional approaches face significant limitations, as they cannot handle modern, dynamic systems. Additionally, there is a critical research gap between the advanced techniques proposed by academic research and what commercial tools implement, which still rely on simple approaches. Another problem is that most APM tools are closed source or have limited customization capabilities, making it difficult to add or improve new techniques Thantharate (2023).

Therefore, due to these limitations, companies continue to rely on manual effort to handle data and solve problems, when these problems could be significantly reduced using artificial intelligence and machine learning technologies.

2.5 AI Integration in Observability:

Building on the limitations discussed earlier, the integration of AI into observability tools represents a fundamental shift from rule-based to behavior-based analysis. According to Shen et al., AIOps systems utilize big data and machine learning through five core abilities: perception (data collection), detection (anomaly identification), location (root cause analysis), action (automated remediation), and interaction (human-computer collaboration) Mahida (2024). Research demonstrates substantial improvements, with Mean Time to Detect (MTTD) reduced from 10 minutes to one minute and Mean Time to Repair (MTTR) decreased from 60 minutes to 30 seconds Ramdoss and Rajan (2025).

2.6 AI Techniques in Modern Observability Platforms:

Modern observability platforms leverage various AI/ML techniques to address limitations of traditional monitoring. These include traditional ML models and advanced deep learning architectures such as LSTM and Transformers for anomaly detection and predictive analytics Bauskar (2024). Root cause analysis employs causal discovery algorithms, graph-based analysis, and pattern mining techniques to identify failure sources. Additionally, AI enables automated remediation through reinforcement learning and intelligent alerting systems that reduce false positives Lazarus (2024).

2.7 Challenges of AI Integration:

Despite these benefits, implementation faces significant challenges. Technical challenges include interoperability with legacy systems, data quality issues such as imbalanced data and lack of labels, and scalability concerns when processing millions of metrics per second. Organizational challenges involve culture shifts and gaining trust in AIOps decisions, as "how to evolve from these traditional heterogeneous systems to AIOps is of great challenge". The black-box nature of complex models raises explainability concerns, while high costs and resource requirements pose barriers to adoption Ahmed et al. (2016). Additionally, the lack of public benchmarks hampers the development of new approaches. These challenges underscore the importance of understanding practitioners' real-world experiences with AI-powered observability tools, which form the focus of this study.

3. RESEARCH METHODOLOGY

This research employed a qualitative approach using semi-structured interviews to explore how AI is transforming observability tools. This design was appropriate because it enabled detailed exploration of AI functionalities and real-world user experiences that quantitative methods cannot adequately capture. Semi-structured interviews were conducted via video conferencing platforms (Zoom or Microsoft Teams), with each session lasting approximately 15-20 minutes.

Purposive sampling was used to select 15 IT professionals with at least 1 year of experience using observability tools, including those with and without AI-powered features. Target participants included DevOps Engineers, Site Reliability Engineers (SREs), Software Engineers, IT Operations staff, and System Administrators employed in small-to-large organizations, ensuring diverse exposure to monitoring challenges across different organizational scales.

The interview guide was designed to allow natural conversation flow while ensuring key topics were addressed. Each interview began with questions about the participant's current job role, years of experience in system observability, and the observability tools they use Shen et al., (2020). The discussion then focused on AI-powered features, examining their benefits and operational impact for users, and exploring adoption barriers for non-users. Questions were predominantly open-ended to encourage detailed responses.

Interview transcripts were coded collaboratively by the researcher and a second coder to enhance reliability and reduce bias. Coding was managed using Microsoft Excel. The process began with reading responses and assigning descriptive labels to segments that captured similar concepts or experiences, which generated 11 preliminary themes. These codes were then reviewed and refined through an iterative process, grouping and consolidating related codes based on similarity and relevance to the research objectives. Through this refinement, the initial 11 themes were reduced to 5 final themes that best represented the core patterns in the data.

Ethical considerations were addressed by obtaining informed consent from all participants prior to interviews. Participants were assured of confidentiality, and all data were stored securely in accordance with institutional guidelines.

4. RESULTS

This section presents the key findings from semi-structured interviews with 15 IT professionals who use observability tools in their daily work. The analysis revealed five major themes that characterize practitioners' experiences with AI integration in observability platforms. These themes highlight the practical realities of AI adoption, including the distinction between basic and advanced AI capabilities, deployment challenges, the role of external AI tools, skill gaps, and data quality requirements. Figure 1 provides an overview of the distribution of these themes across participants.

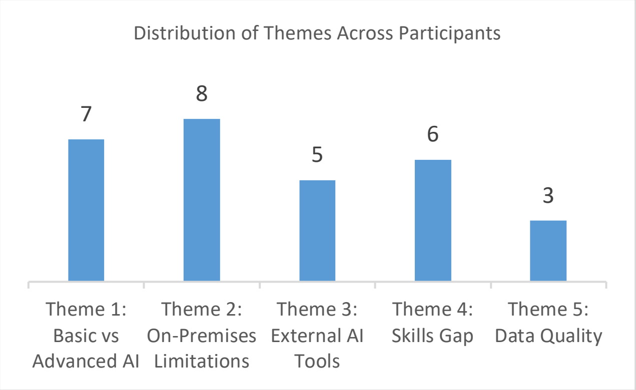

Figure 1 Distribution of Themes Across Participants

4.1. The Contrast Between Integrated AI and Advanced AI in Observability Tools:

Most observability platforms include basic AI capabilities like anomaly detection and automated alerts as part of their standard features. On the other hand, some platforms incorporate advanced AI functionalities as optional extensions available at an additional cost, as noted in a personal interview: "When we explored Dynatrace, we found they have a powerful AI engine called Davis AI, but it came at a higher cost." This highlights the decision organizations' face between standard AI capabilities and investing in premium AI services with more advanced features.

4.2 On-Premises Deployment and AI Performance Limitations

Some organizations use observability tools on-premises rather than using cloud-based solutions due to security and compliance requirements. This can reduce the effectiveness of AI features even though it solves privacy concerns.

As one of the interviewers said,” We deployed Grafana locally due to data regulations”, and after being asked about the advantage of using monitoring tools on the cloud, he replied, "For example, on the cloud, we can use ML to predict when traffic will occur so that we can act before a problem occurs."

4.3. Using External AI Tools for Query Generation

Several engineers have shown that they use external AI engines such as ChatGPT and Copilot to help them write complex configurations files and queries for their work on monitoring platforms.

Although there are some tools that provide chatbots, as one of the participants said, "We use Datadog and they launched a chatbot recently and it has many advantages, including fixing codes” but he made it clear that using external chatbot is cost-effective, "I can fix it with Copilot at a lower cost."

Another participant explained that writing the prompt correctly and accurately helps make the code more accurate, saying, "I use ChatGPT to write a YAML file, but the accuracy of the code depends on the prompt I send."

4.4 Skills Gap in AI-Powered Observability

There is a noticeable skills gap in AI-powered observability, as the junior participants in the DevOps field, and several of them mentioned that it is often difficult to know whether an issue flagged by AI is a false alarm or a real problem. One participant expressed "I'm still learning observability, and when the AI flags an anomaly, I'm not sure if it's real or false,” while another stated, “Although our team has experience, they do not always trust the results provided by AI.”

4.5 Data Quality Requirements for AI Effectiveness

AI-powered features in observability tools rely on high-quality, well-structured data to function effectively. Poor data quality, inconsistent formats, or incomplete information can significantly reduce AI accuracy and increase false alerts, as noted “there is one time AI anomaly detection triggering alerts for everything, and it was because there are gaps in the data”, so that will be requiring the team to invest time in data cleansing and validation. This highlights the often-overlooked requirement of data quality as a foundation for effective AI implementation.

5. DISCUSSION

This study explored how AI is transforming observability tools from practitioners' perspectives. The findings reveal that successful AI adoption in observability extends beyond technology implementation, it requires strategic planning, skilled teams, and robust data practices.

5.1 Deployment Trade-offs

On-premises deployment meets security and compliance needs but limits AI effectiveness. Participants reported losing cloud-based features like predictive analytics. This extends prior discussions on technical challenges, showing how organizational constraints can restrict AI benefits even when technology is mature.

5.2 External AI Tools Fill the Gap

Practitioners widely use external tools (ChatGPT, Copilot) to complement observability platforms, an aspect not covered in earlier studies. One participant said, “I can fix it with Copilot at a lower cost,” but others noted accuracy depends on prompt quality, introducing new challenges.

5.3 Trust Remains a Critical Barrier

The trust gap aligns with previous findings on organizational issues. Despite research showing AI can reduce Mean Time to Detect from 10 minutes to one, participants expressed skepticism toward AI alerts. This highlights human factors and model explainability as critical concerns.

5.4 Data Quality as Foundation

Poor data quality remains a fundamental challenge, consistent with prior research. One participant noted “AI anomaly detection triggering alerts for everything” due to data gaps, confirming that AI effectiveness depends on robust data infrastructure, a challenge persisting even after AI adoption.

6. IMPLICATION

This study provides empirical evidence that AI adoption in observability is more complex than simple technology implementation. Future research should examine the entire ecosystem of tools practitioners use, including both integrated and external solutions, rather than focusing solely on platform capabilities. The gap between AI's demonstrated improvements and practitioners' confidence suggests that adoption models must prioritize trust, explainability, and organizational readiness. The findings also support calls for bridging academic research and commercial implementations, while highlighting how practitioners creatively use external AI tools to address existing gaps.

Adopting AI technologies, whether within observability tools or independently, requires an integrated approach that includes verifying security policies to avoid compliance risks, assessing costs in line with budget, and ensuring technical teams receive continuous training. Investment must also be made in data quality, as it represents the infrastructure for the effectiveness of AI-based analytics. In addition, it is recommended to integrate practices such as prompt engineering into training programs and build trust in teams by distinguishing real anomalies from false positives.

7. LIMITATIONS

Despite the valuable results, the study had some limitations. Conducting the study with a sample of 15 participants is suitable for a qualitative experiment, but we cannot generalize these results to all observability tools used in organizational contexts. Moreover, the study included participants with varying levels of expertise in AI and observability tools, which may have affected the depth and technical accuracy of their responses.

Additionally, the findings rely entirely on participants' self-reported experiences rather than on practical testing or direct observation of the tools. The study did not include practical experiments to verify participants' claims about the capabilities and limitations of AI.

8. CONCLUSION

This study explored the impact of AI on observability tools through the experiences of IT professionals. The findings indicate that AI adoption is not just a technical upgrade, but a multidimensional process influenced by deployment approaches, organizational awareness, and data quality.

Five key insights emerged from this research. First, practitioners must navigate between basic integrated AI features and premium advanced capabilities, each with distinct cost implications. Second, deployment choices significantly impact AI effectiveness, with on-premises solutions addressing compliance needs but sacrificing predictive capabilities. Third, practitioners increasingly rely on external AI tools like ChatGPT and Copilot to complement their observability platforms, revealing gaps in existing tool capabilities. Fourth, trust in AI-generated alerts remains limited, particularly among less experienced practitioners who struggle to distinguish genuine anomalies from false positives. Finally, data quality proves fundamental to AI effectiveness, with poor data undermining even the most sophisticated AI features.

These findings have important implications for multiple stakeholders. Organizations should approach AI adoption strategically, ensuring alignment between security requirements, budget constraints, team capabilities, and data infrastructure maturity. Tool vendors need to enhance AI explainability and consider tighter integration with AI assistants to address practitioners' evolving needs. Researchers should examine the broader ecosystem of tools practitioners use rather than focusing solely on individual platform capabilities.

While this study provides valuable insights into current practices, future research should investigate how practitioners' experiences evolve with increased AI exposure and explore the long-term organizational impact of AI-powered observability adoption across different industries and contexts.

REFERENCES

Cheng, Q., et al. (2023). AI for IT operations (AIOps) on cloud platforms: Reviews, opportunities, and challenges. arXiv. https://doi.org/10.48550/arXiv.2304.04661

Shen, S., Zhang, J., Huang, D., & Xiao, J. (2020). Evolving from traditional systems to AIOps: Design, implementation, and measurements. Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), 280. https://doi.org/10.1109/AEECA49918.2020.9213650

Carnevale, L., Balouek, D., Sebbio, S., Parashar, M., & Villari, M. (2025). Private distributed resource management data: Predicting CPU utilization with Bi-LSTM and federated learning. In 2025 IEEE 25th International Symposium on Cluster, Cloud and Internet Computing (CCGrid), 266–275. https://doi.org/10.1109/CCGRID64434.2025.00048

Biswas, S. (2025). Comprehensive observability and monitoring strategies using Datadog. SSRN Electron. J. https://doi.org/10.2139/ssrn.5193639

Parvathinathan, K. (2025). Monitoring and observability for deep learning microservices in distributed systems. Connectivity Science.

Siddiqui, I., Pandey, A., Jain, S., Kothadia, H., Agrawal, R., & Chankhore, N. (2023). Comprehensive monitoring and observability with Jenkins and Grafana: A review of integration strategies, best practices, and emerging trends. In 2023 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 1–5. https://doi.org/10.1109/ISMSIT58785.2023.10304904

Mahida, A. (2023). Enhancing observability in distributed systems—A comprehensive review. Journal of Mathematical Computation and Applications, 2, 1–4. https://doi.org/10.47363/JMCA/2023(2)135

Mart, O., Negru, C., Pop, F., & Castiglione, A. (2020). Observability in Kubernetes cluster: Automatic anomalies detection using Prometheus. In 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 565–570. https://doi.org/10.1109/HPCC-SmartCity-DSS50907.2020.00071

Faseeha, U., Syed, H. J., Samad, F., Zehra, S., & Ahmed, H. (2025). Observability in microservices: An in-depth exploration of frameworks, challenges, and deployment paradigms. IEEE Access, 13, 72011–72039. https://doi.org/10.1109/ACCESS.2025.3562125

Thantharate, P. (2023). IntelligentMonitor: Empowering DevOps environments with advanced monitoring and observability. In 2023 International Conference on Information Technology (ICIT), 800–805. https://doi.org/10.1109/ICIT58056.2023.10226123

Mahida, A. (2024). Integrating observability with DevOps practices in financial services technologies: A study on enhancing software development and operational resilience. International Journal of Advanced Computer Science and Applications, 15. https://doi.org/10.14569/IJACSA.2024.0150701

Ramdoss, V., & Rajan, P. (2025). Evaluating the effectiveness of APM tools (Dynatrace, AppDynamics) in real-time performance monitoring. Eastasouth Journal of Information Systems and Computer Science, 2, 399–402. https://doi.org/10.58812/esiscs.v2i03.593

Lazarus, R. (2024). Modernizing mainframe monitoring: A comparative analysis of observability dashboarding tools. International Journal of Research in Applied Science and Engineering Technology, 12(8), 132–138. https://doi.org/10.22214/ijraset.2024.63866

Ahmed, T., Bezemer, C.-P., Chen, T.-H. P., Shang, W., & Hassan, A. E. (2016). Studying the effectiveness of application performance management (APM) tools for detecting performance regressions for web applications: An experience report. Proceedings of the 2016 ACM/IEEE International Conference on Software Engineering

(ICSE). https://doi.org/10.1145/2901739.2901774

Bauskar, S. (2024). Enhancing system observability with machine learning techniques for anomaly detection. International Journal of Computing and Information Technology, 14, 64–70.

Shen, S., Zhang, J., Huang, D., & Xiao, J. (2020). Evolving from traditional systems to AIOps: Design, implementation, and measurements. In 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), 276–280. https://doi.org/10.1109/AEECA49918.2020.9213650

Tatineni, S. (2023). AIOps in cloud-native DevOps: IT operations management with artificial intelligence. Journal of Artificial Intelligence and Cloud Computing, 1–7. https://doi.org/10.47363/JAICC/2023(2)154.