COLOMBO SCIENTIFIC PUBLISHING

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE & INFORMATION SYSTEM
E-ISSN: 2536-7919
P-ISSN: 2536-7900

PAGE NO: 21-28

Algorithmic AIOps and AI-Driven DevOps for Intelligent Software Deployment and Operations in Cloud-Native Enterprises

Michael J. Andersson
Department of Computer and Systems Sciences, Stockholm University, Sweden

ARTICLE INFO

Article history:

Submission: January 01, 2026
Accepted: January 17, 2026
Published: February 03, 2026
VOLUME: Vol.11 Issue 02 2026

Keywords:
AIOps, AI-Driven DevOps, Cloud-Native Systems, Log Analytics, Operational Intelligence, Software Reliability, Intelligent Automation

ABSTRACT

The rapid acceleration of cloud-native software delivery, microservice-oriented architectures, and continuous integration and deployment pipelines has fundamentally altered the operational fabric of modern software engineering. Organizations today no longer struggle only with writing correct code but with governing, monitoring, and evolving large-scale, continuously changing software ecosystems. Within this environment, the convergence of Artificial Intelligence for IT Operations and AI-driven DevOps has emerged as one of the most consequential paradigms of contemporary digital infrastructure management. This article develops a comprehensive theoretical and empirical synthesis of how AIOps and AI-driven DevOps jointly reshape software deployment, reliability engineering, and operational decision-making. Building on machine learning-based automation for deployment and maintenance articulated in the recent synthesis of AI-driven DevOps (Varanasi, 2025), this study integrates broader AIOps research on anomaly detection, log analytics, tracing, governance, and predictive reliability engineering into a single unified analytical framework.

The article positions AIOps not as a standalone toolset but as a socio-technical intelligence layer embedded within DevOps pipelines. Drawing on the extensive literature on log-based anomaly detection, time-series modeling, distributed tracing, and failure prediction, it argues that AI-driven DevOps represents a shift from reactive, human-centered operations toward proactive, data-centric and semi-autonomous operational governance. This transformation is examined historically, theoretically, and methodologically. Historically, the work situates AIOps within the evolution from traditional system administration to automated operations and continuous delivery. Theoretically, it draws on systems theory, reliability engineering, and organizational learning to conceptualize how machine learning alters the epistemology of operational knowledge. Methodologically, it develops a structured qualitative-analytical synthesis of prior empirical studies, surveys, and industrial case analyses.

Ultimately, the article contributes a comprehensive, integrative theory of AI-driven DevOps as a foundational pillar of modern software engineering. It proposes that future cloud enterprises will increasingly rely on algorithmic operational intelligence not merely to keep systems running, but to actively shape how software is designed, deployed, and evolved over time, consistent with the emerging evidence from AIOps research and industrial practice (Dang et al., 2019; Gulenko et al., 2020; Varanasi, 2025).

INTRODUCTION

In The evolution of software engineering over the last two decades has been characterized by a relentless drive toward speed, scale, and continuous change. The transition from monolithic applications to microservice-based cloud-native systems has fundamentally transformed how software is built, deployed,

and operated. Continuous integration and continuous deployment pipelines now enable organizations to release new versions of software multiple times per day, while elastic cloud infrastructures dynamically allocate computing resources to meet fluctuating demand. Yet this unprecedented flexibility has also produced an operational environment of extraordinary complexity, in which thousands of distributed components generate vast volumes of telemetry, logs, traces, and performance metrics. Traditional forms of system administration and monitoring, grounded in manual inspection and rule-based alerting, have proven increasingly inadequate to manage this complexity, a challenge repeatedly documented in AIOps literature (Dang et al., 2019; Rijal et al., 2022).

It is within this context that Artificial Intelligence for IT Operations and AI-driven DevOps have emerged as transformative paradigms. AIOps applies machine learning, data mining, and advanced analytics to operational data in order to automate anomaly detection, root cause analysis, and remediation, while AI-driven DevOps integrates these capabilities directly into the software delivery lifecycle. Rather than treating operations as a downstream activity that reacts to failures after they occur, AI-driven DevOps embeds algorithmic intelligence into deployment, testing, scaling, and maintenance processes, enabling systems to anticipate, diagnose, and sometimes even correct their own problems. The significance of this shift has been articulated in contemporary reviews of intelligent automation for deployment and maintenance, which describe AI-driven DevOps as the convergence of machine learning and continuous delivery into a new operational paradigm (Varanasi, 2025).

The intellectual roots of this transformation can be traced to several converging traditions. From the perspective of software engineering, the DevOps movement sought to break down the silos between development and operations, enabling faster feedback loops and more reliable releases (Sen, 2020). From the perspective of artificial intelligence, advances in deep learning, anomaly detection, and sequence modeling made it possible to extract meaningful patterns from high-volume, high-dimensional operational data (Chalapathy and Chawla, 2019; Akoglu et al., 2015). From the perspective of digital transformation, organizations recognized that data-driven automation could fundamentally reshape enterprise operations, allowing them to scale innovation at unprecedented speed (Davidovski, 2018). AIOps and AI-driven DevOps sit at the intersection of these trajectories, representing not merely a technological upgrade but a reconfiguration of how organizations understand and govern their software systems.

Despite the growing body of research on AIOps methods, anomaly detection, and intelligent monitoring, the integration of these techniques into a coherent DevOps framework remains under-theorized. Surveys of log analysis, tracing, and failure management have cataloged an impressive array of machine learning models and data pipelines, yet they often treat these techniques as isolated tools rather than as components of a broader socio-technical system (He et al., 2021; Notaro et al., 2021). At the same time, industry-focused analyses of DevOps and AIOps tend to emphasize practical benefits without fully engaging with the theoretical and methodological challenges of algorithmic operations (Battina, 2021; Paradkar, 2020). This creates a literature gap in which the strategic and epistemic implications of AI-driven DevOps remain insufficiently explored.

One of the most important unresolved issues concerns how machine learning changes the nature of operational knowledge. In traditional operations, engineers developed mental models of system behavior based on experience, documentation, and direct observation. In AIOps-enabled environments, by contrast, predictive models infer patterns from data that may be too complex or high-dimensional for humans to interpret directly. Studies of log anomaly detection, for example, demonstrate that deep neural networks can identify subtle deviations in event sequences that correlate with failures, even when no explicit rules exist (Le and Zhang, 2022; Han and Yuan, 2021). Yet this also raises questions about explainability, trust, and governance, as operators must decide whether to act on algorithmic recommendations whose internal logic may be opaque (Gulenko et al., 2020).

Another critical gap lies in understanding how AIOps capabilities interact with the continuous deployment pipelines of DevOps. Varanasi (2025) emphasizes that AI-driven DevOps enables intelligent automation across the entire deployment and maintenance lifecycle, from code integration to production monitoring. However, much of the AIOps literature still focuses on post-deployment monitoring and failure management rather than on how predictive models can influence release decisions, testing strategies, and

resource allocation before problems occur (Li et al., 2020; Lyu et al., 2021). Bridging this gap requires a holistic perspective that situates anomaly detection, log mining, and predictive analytics within the dynamic workflows of modern software delivery.

This article addresses these gaps by developing a comprehensive analytical synthesis of AIOps and AI-driven DevOps. Rather than presenting a narrow technical evaluation of specific algorithms, it constructs a multi-layered framework that integrates historical context, theoretical foundations, methodological approaches, and empirical insights from the literature. The central argument is that AI-driven DevOps represents a new form of operational intelligence, in which algorithmic models function as cognitive agents that augment and reshape human decision-making in software engineering. This argument is grounded in the extensive body of research on log analytics, time-series anomaly detection, distributed tracing, and predictive reliability, as well as in recent conceptualizations of AI-enabled DevOps automation (Varanasi, 2025; Zhaoxue et al., 2021; Zhao et al., 2021).

The remainder of this article unfolds as follows. The methodology section explains how the literature-based analytical synthesis was constructed, including criteria for selecting and interpreting sources. The results section presents an integrated account of how AIOps techniques function within AI-driven DevOps pipelines, drawing on empirical findings from surveys, case studies, and benchmark datasets. The discussion then critically examines the theoretical, organizational, and governance implications of algorithmic operations, comparing competing scholarly viewpoints and outlining future research directions. The conclusion synthesizes these insights and reflects on the broader significance of AI-driven DevOps for the future of software engineering.

METHODOLOGY

The methodological foundation of this study is a qualitative-analytical synthesis of the existing scholarly and technical literature on AIOps, AI-driven DevOps, and intelligent operations in cloud-native software systems. Rather than adopting a quantitative meta-analysis or a narrow systematic review, this research employs an interpretive, theory-building approach that is particularly well suited to complex, rapidly evolving technological domains. This approach recognizes that the value of AIOps research lies not only in performance metrics but also in how conceptual frameworks, design paradigms, and organizational practices evolve in response to algorithmic automation, as emphasized in multivocal literature reviews of AIOps (Rijal et al., 2022; Korzeniowski and Goczyla, 2022).

The primary corpus for this synthesis was constructed from the references provided, which collectively represent a comprehensive cross-section of the AIOps and DevOps research landscape. These sources include doctoral dissertations on enterprise AI adoption (Siddique, 2018), practitioner-oriented monographs on AIOps implementation (Sabharwal, 2022), theoretical surveys of anomaly detection and log analytics (Chalapathy and Chawla, 2019; He et al., 2021), and empirical studies of large-scale cloud platforms (Li et al., 2020; Zhao et al., 2021). The inclusion of the recent IEEE conference review on AI-driven DevOps (Varanasi, 2025) ensures that the synthesis reflects the current state of intelligent automation in deployment and maintenance.

The analytical process unfolded in several interrelated stages. First, the literature was conceptually coded according to its primary focus, such as log analysis, anomaly detection, distributed tracing, predictive failure modeling, governance, or DevOps integration. This thematic coding enabled the identification of recurring patterns and debates across otherwise diverse sources, consistent with established methods in qualitative literature analysis (Notaro et al., 2021; Soldani and Brogi, 2022). Second, these themes were mapped onto the lifecycle of software delivery, from development and deployment to monitoring and incident response, reflecting the DevOps perspective emphasized by Sen (2020) and Paradkar (2020). This mapping made it possible to analyze how different AIOps techniques interact with specific stages of the operational pipeline.

Third, the study engaged in what can be described as theoretical triangulation. Insights from systems theory, reliability engineering, and organizational learning were used to interpret the technical findings of AIOps research. For example, predictive models for node failure (Li et al., 2020) were not only examined for their algorithmic properties but also for what they imply about anticipatory governance in large-scale

systems. Similarly, log anomaly detection methods (Le and Zhang, 2022; Han and Yuan, 2021) were interpreted in terms of how they reconfigure the epistemology of fault diagnosis. This interpretive strategy aligns with the view that AIOps is not merely a technical toolkit but a new form of organizational intelligence (Gulenko et al., 2020; Shen et al., 2020).

The methodological rigor of this synthesis also derives from its engagement with empirical benchmarks and datasets that ground theoretical claims in observed system behavior. Studies based on large-scale trace and log repositories, such as TraceBench (Zhou et al., 2014) and KPI anomaly detection benchmarks (Zhang et al., 2021), provide a shared empirical reference point for evaluating algorithmic approaches. Although this article does not reproduce numerical results, it relies on the interpretive conclusions of these studies to assess the practical viability of different AIOps techniques, as recommended in comprehensive surveys of automated log analysis (He et al., 2021; Zhaoxue et al., 2021).

A key methodological challenge in synthesizing AIOps and AI-driven DevOps research is the heterogeneity of data sources, algorithms, and evaluation criteria. Some studies focus on supervised learning with labeled anomalies, while others emphasize unsupervised or semi-supervised approaches for environments where labels are scarce (Zhao et al., 2021; Braei and Wagner, 2020). Similarly, some research examines centralized monitoring architectures, whereas others explore cross-system or domain-adaptation models that can generalize across heterogeneous services (Han and Yuan, 2021). Rather than privileging any single methodological paradigm, this study adopts a pluralistic stance, recognizing that the diversity of approaches reflects the complexity of real-world operations, a point repeatedly emphasized in AIOps mapping studies (Notaro et al., 2021; Rijal et al., 2022).

The limitations of this methodology are also acknowledged. Because the analysis is based on secondary sources rather than original experiments, it necessarily depends on the quality, scope, and reporting practices of the underlying literature. There is also an inherent risk of publication bias, as successful or novel techniques are more likely to be reported than failures or negative results, a concern noted in empirical studies of AIOps solutions (Lyu et al., 2021). Nevertheless, by triangulating across multiple surveys, empirical investigations, and conceptual frameworks, this study aims to provide a robust and nuanced account of AI-driven DevOps as an emergent field of practice and research (Varanasi, 2025; Dang et al., 2019).

RESULTS

The synthesis of the AIOps and AI-driven DevOps literature reveals a multifaceted landscape in which algorithmic intelligence increasingly permeates every stage of software operations. One of the most striking results is the convergence of previously distinct operational functions into integrated data-driven pipelines. Log analysis, distributed tracing, and KPI monitoring, once treated as separate domains, are now increasingly unified through machine learning models that can correlate patterns across heterogeneous data streams, as documented in recent surveys of automated log and trace analysis (He et al., 2021; Soldani and Brogi, 2022). This convergence is central to the vision of AI-driven DevOps articulated by Varanasi (2025), in which intelligent automation spans deployment, monitoring, and maintenance.

A first major result concerns the role of log data as the backbone of AIOps. Logs record the discrete events generated by software components, capturing both normal behavior and anomalies. Research on log-based anomaly detection demonstrates that deep learning models, particularly those based on sequence modeling, can identify subtle deviations that precede or accompany system failures (Le and Zhang, 2022; Zhao et al., 2021). These models do not merely flag errors but infer probabilistic patterns of normality, enabling systems to detect novel or previously unseen failure modes. In AI-driven DevOps contexts, such capabilities are increasingly integrated into continuous monitoring pipelines, allowing deployment decisions and rollback strategies to be informed by real-time log intelligence (Varanasi, 2025; Shen et al., 2020).

A second important result emerges from the literature on distributed tracing. Tracing frameworks such as X-Trace and Retrace were originally developed to capture causal relationships among distributed components (Fonseca et al., 2007; Sheldon and Weissman, 2007). More recent research has combined these tracing techniques with deep learning to enable automated anomaly detection and root cause analysis

across microservice architectures (Nedelkoski et al., 2019; Soldani and Brogi, 2022). The synthesis shows that tracing-based AIOps provides a form of structural awareness that complements log analysis by revealing how failures propagate through service dependencies. Within AI-driven DevOps pipelines, this structural awareness enables more precise deployment strategies, such as targeted canary releases and intelligent traffic shifting, which reduce the blast radius of potential faults (Varanasi, 2025; Paradkar, 2020).

A third result concerns time-series and KPI-based anomaly detection. Performance metrics such as latency, throughput, and error rates provide continuous signals of system health. Surveys of time-series anomaly detection show that both statistical and deep learning approaches can identify deviations in these metrics that correlate with emerging problems (Braei and Wagner, 2020; Blazquez-Garcia et al., 2021). Empirical studies in large-scale software services demonstrate that even partial labels can be sufficient to train robust KPI anomaly detectors, enabling proactive intervention before users are affected (Zhang et al., 2021; Zhao et al., 2021). In the context of AI-driven DevOps, such KPI intelligence is increasingly used to automate scaling decisions, trigger deployment rollbacks, and optimize resource allocation, thereby closing the loop between operational analytics and delivery pipelines (Varanasi, 2025; Levin et al., 2019).

A fourth result arises from predictive modeling of infrastructure failures. Large-scale cloud platforms generate vast amounts of telemetry about nodes, networks, and storage systems. Research on failure prediction demonstrates that machine learning models can forecast node outages with significant lead time, allowing preemptive migration of workloads and proactive maintenance (Li et al., 2020). These predictive capabilities align closely with the AI-driven DevOps vision of anticipatory operations, in which deployment and scheduling decisions are informed by probabilistic forecasts of infrastructure health rather than reactive alarms (Varanasi, 2025; Sabharwal, 2022). The literature further shows that data splitting and training strategies have a profound impact on the reliability of these models, highlighting the importance of methodological rigor in operational AI (Lyu et al., 2021).

A fifth result concerns the organizational and governance dimensions of AIOps. Studies of AI-supported system administration emphasize that increasing levels of automation require corresponding frameworks for human oversight, accountability, and trust (Gulenko et al., 2020; Shen et al., 2020). Rather than fully autonomous operations, most organizations adopt a graduated model in which AI systems provide recommendations or execute routine actions under human supervision. This hybrid governance model is particularly evident in AI-driven DevOps environments, where automated testing, deployment, and monitoring coexist with human decision-making about release strategies and risk tolerance (Varanasi, 2025; Battina, 2021). The result is a socio-technical system in which algorithmic and human intelligences are deeply intertwined.

Collectively, these results paint a picture of AI-driven DevOps as a dynamic ecosystem of data, models, and workflows. The literature consistently shows that the technical effectiveness of AIOps depends not only on algorithmic accuracy but also on integration with deployment pipelines, organizational processes, and governance structures (Dang et al., 2019; Notaro et al., 2021). This integrated perspective provides the empirical foundation for the deeper theoretical analysis developed in the discussion.

DISCUSSION

The results of this synthesis invite a deeper theoretical reflection on what AI-driven DevOps represents for the epistemology and practice of software engineering. At its core, the integration of AIOps into DevOps pipelines transforms how organizations know their systems. Traditional monitoring and operations relied on explicit rules, dashboards, and human interpretation. By contrast, machine learning models infer latent structures and probabilistic patterns from massive volumes of operational data, creating a form of algorithmic perception that operates alongside, and sometimes beyond, human cognition (Chalapathy and Chawla, 2019; Le and Zhang, 2022). This shift raises fundamental questions about trust, control, and responsibility in digital infrastructure, questions that are increasingly foregrounded in AI governance research (Gulenko et al., 2020; Shen et al., 2020).

One of the most significant theoretical implications concerns the nature of causality in complex systems. In distributed microservice architectures, failures often emerge from nonlinear interactions among

components rather than from isolated faults. AIOps techniques such as trace-based deep learning and graph-based anomaly detection attempt to model these interactions explicitly, constructing representations of how events propagate through networks of services (Akoglu et al., 2015; Nedelkoski et al., 2019). In AI-driven DevOps contexts, these causal models inform deployment and maintenance decisions, effectively embedding a theory of system behavior into the operational pipeline itself (Varanasi, 2025). This represents a departure from the traditional engineering approach, in which causal reasoning was primarily the domain of human experts.

At the same time, the literature reveals persistent tensions between model-driven inference and human judgment. Empirical studies show that while deep learning models can achieve impressive detection accuracy, they often struggle with interpretability and generalization across environments (Han and Yuan, 2021; Lyu et al., 2021). This creates a risk of over-reliance on models that may be poorly calibrated or biased by training data. Governance frameworks therefore emphasize the need for human oversight and adaptive control, advocating for levels of automation that can be tuned according to organizational risk tolerance (Gulenko et al., 2020; Battina, 2021). In AI-driven DevOps, this translates into deployment strategies that combine automated triggers with human approval gates, ensuring that algorithmic intelligence enhances rather than undermines accountability (Varanasi, 2025; Sen, 2020).

Another important debate concerns the scalability and sustainability of AIOps solutions. Surveys of automated log analysis and anomaly detection consistently highlight the challenges of data drift, evolving system architectures, and changing workload patterns (He et al., 2021; Korzeniowski and Goczyla, 2022). Models trained on historical data may become obsolete as services are updated and usage patterns shift, a phenomenon that is particularly acute in continuous deployment environments. AI-driven DevOps must therefore incorporate mechanisms for continuous learning, validation, and retraining, effectively treating models as living components of the software system (Varanasi, 2025; Zhaoxue et al., 2021). This blurs the boundary between software code and operational intelligence, further reinforcing the notion of DevOps as a holistic, data-centric discipline.

The discussion also highlights the strategic implications of AI-driven DevOps for enterprise competitiveness. By enabling faster detection of anomalies, more precise root cause analysis, and predictive maintenance, AIOps reduces downtime and improves service reliability, which directly impacts customer satisfaction and business performance (Levin et al., 2019; Li et al., 2020). More subtly, however, algorithmic operations also create a feedback loop between product development and operational data, allowing organizations to experiment, learn, and iterate at unprecedented speed. This dynamic capability is a key component of digital transformation, as organizations leverage data-driven insights to continuously optimize their software ecosystems (Davidovski, 2018; Siddique, 2018).

Yet this transformation is not without risks. The automation of operational decisions raises concerns about resilience and brittleness. If deployment and remediation pipelines become too tightly coupled to specific models, failures or mispredictions can cascade rapidly through the system, potentially amplifying rather than mitigating risk (Dang et al., 2019; Zhao et al., 2021). Scholars therefore argue for diversity and redundancy in AIOps architectures, combining multiple models, data sources, and validation mechanisms to avoid single points of failure (Notaro et al., 2021; Soldani and Brogi, 2022). In AI-driven DevOps, this translates into layered defense strategies that integrate log analytics, KPI monitoring, and tracing into a robust operational fabric (Varanasi, 2025).

From a future research perspective, the synthesis suggests several promising directions. One is the development of more explainable and transparent AIOps models that can support human understanding and trust. Another is the exploration of cross-system and transfer learning approaches that allow models to generalize across heterogeneous environments, reducing the cost and complexity of deployment (Han and Yuan, 2021; Zhu et al., 2021). A third direction involves the integration of AIOps with security and compliance frameworks, extending the principles of AI-driven DevOps into the realm of DevSecOps (Sen, 2020; Masood and Hashmi, 2019). All of these directions build on the foundational insight that intelligent automation is becoming an integral part of how software systems are designed, deployed, and governed (Varanasi, 2025; Shen et al., 2020).

CONCLUSION

This article has presented a comprehensive analytical synthesis of AIOps and AI-driven DevOps as foundational paradigms of modern software engineering. By integrating insights from log analytics, distributed tracing, time-series anomaly detection, predictive failure modeling, and governance research, it has argued that algorithmic operational intelligence is transforming not only how systems are maintained but how they are conceived and evolved. The recent articulation of AI-driven DevOps as a framework for intelligent deployment and maintenance underscores the strategic importance of this transformation (Varanasi, 2025).

The evidence reviewed here shows that AIOps provides powerful tools for understanding and managing the complexity of cloud-native systems, yet its true value lies in its integration with DevOps workflows that embed intelligence into the continuous delivery pipeline. As organizations continue to scale their digital infrastructures, the challenge will be to balance automation with human judgment, innovation with reliability, and efficiency with governance. AI-driven DevOps offers a compelling vision of how this balance can be achieved, but realizing that vision will require ongoing research, methodological rigor, and thoughtful organizational design.

REFERENCES

1. Blazquez-Garcia, A., Conde, A., Mori, U., and Lozano, J. A. (2021). A review on outlier and anomaly detection in time series data. ACM Computing Surveys, 54(3), 1–33.

2. Varanasi, S. R. (2025, August). AI-Driven DevOps in Modern Software Engineering—A Review of Machine Learning Based Intelligent Automation for Deployment and Maintenance. In 2025 IEEE 2nd International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS), IEEE.

3. Zhang, S., Zhao, C., Sui, Y., Su, Y., Sun, Y., Zhang, Y., Pei, D., and Wang, Y. (2021). Robust KPI anomaly detection for large-scale software services with partial labels. IEEE International Symposium on Software Reliability Engineering.

4. Rijal, L., Colomo-Palacios, R., and Sanchez-Gordon, M. (2022). AIOps: A multivocal literature review. Artificial Intelligence for Cloud and Edge Computing, 31–50.

5. Gulenko, A., Acker, A., Kao, O., and Liu, F. (2020). AI-governance and levels of automation for AIOps-supported system administration. International Conference on Computer Communications and Networks.

6. He, S., He, P., Chen, Z., Yang, T., Su, Y., and Lyu, M. R. (2021). A survey on automated log analysis for reliability engineering. ACM Computing Surveys, 54(6).

7. Dang, Y., Lin, Q., and Huang, P. (2019). AIOps: Real-world challenges and research innovations. IEEE ACM International Conference on Software Engineering Companion.

8. Li, Y., Jiang, Z. M. J., Li, H., Hassan, A. E., He, C., Huang, R., Zeng, Z., Wang, M., and Chen, P. (2020). Predicting node failures in an ultra-large-scale cloud computing platform. ACM Transactions on Software Engineering and Methodology, 29(2), 1–24.

9. Notaro, P., Cardoso, J., and Gerndt, M. (2021). A systematic mapping study in AIOps. Service-Oriented Computing Workshops.

10. Le, V.-H., and Zhang, H. (2022). Log-based anomaly detection with deep learning: How far are we. International Conference on Software Engineering.

11. Korzeniowski, L., and Goczyla, K. (2022). Landscape of automated log analysis. IEEE Access, 10, 21892–21913.

12. Sabharwal, N. (2022). Hands-on AIOps. Springer.

13. Sen, A. (2020). DevOps, DevSecOps, AIOps paradigms to IT operations. Lecture Notes in Electrical Engineering.

14. Nedelkoski, S., Cardoso, J., and Kao, O. (2019). Anomaly detection and classification using distributed tracing and deep learning. IEEE ACM CCGRID.

15. Davidovski, V. (2018). Exponential innovation through digital transformation. International Conference on Applications in Information Technology.

16. Siddique, S. (2018). The road to enterprise artificial intelligence: A case studies driven exploration. PhD Dissertation.

17. Zhaoxue, J., Tong, L., Zhenguo, Z., Jingguo, G., Junling, Y., and Liangxiong, L. (2021). A survey on log research of AIOps: Methods and trends. Mobile Networks and Applications, 26(6), 2353–2364.

18. Zhao, N., Wang, H., Li, Z., Peng, X., Wang, G., Pan, Z., Wu, Y., Feng, Z., Wen, X., Zhang, W., Sui, K., and Pei, D. (2021). An empirical investigation of practical log anomaly detection for online service systems. European Software Engineering Conference and Symposium on the Foundations of Software Engineering.

19. Masood, A., and Hashmi, A. (2019). AIOps: Predictive analytics and machine learning in operations. Cognitive Computing Recipes.

20. Paradkar, S. (2020). APM to AIOps core transformation. Global Journal of Enterprise Information System.

21. Shen, S., Zhang, J., Huang, D., and Xiao, J. (2020). Evolving from traditional systems to AIOps. IEEE Conference on Advances in Electrical Engineering and Computer Applications.

22. Braei, M., and Wagner, S. (2020). Anomaly detection in univariate time series. ArXiv.

23. Akoglu, L., Tong, H., and Koutra, D. (2015). Graph based anomaly detection and description. Data Mining and Knowledge Discovery, 29(3), 626–688.

24. Chalapathy, R., and Chawla, S. (2019). Deep learning for anomaly detection. ArXiv.

25. Fonseca, R., Porter, G., Katz, R. H., and Shenker, S. (2007). X-Trace. USENIX Symposium on Networked Systems Design and Implementation.

26. Sheldon, M., and Weissman, G. V. B. (2007). Retrace. Workshop on Modeling, Benchmarking and Simulation.

27. Zhou, J., Chen, Z., Wang, J., Zheng, Z., and Lyu, M. R. (2014). TraceBench. IEEE Cloud Computing Technology and Science.

28. Zhu, Y., Meng, W., Liu, Y., Zhang, S., Han, T., Tao, S., and Pei, D. (2021). Unilog. CoRR.

29. Soldani, J., and Brogi, A. (2022). Anomaly detection and failure root cause analysis in microservice-based cloud applications. ACM Computing Surveys.

30. Levin, A., Garion, S., Kolodner, E. K., Lorenz, D. H., Barabash, K., Kugler, M., and McShane, N. (2019). AIOps for a cloud object storage service. IEEE Big Data Congress.