International Journal of Advances in Applied Sciences (IJAAS)
Vol. 15, No. 2, June 2026, pp. 656∼663
ISSN: 2252-8814, DOI: 10.11591/ijaas.v15.i2.pp656-663

Anomaly detection in containerized virtual environments using a process mining approach

Richard Angkawinata, I Made Murwantara
Master of Informatics, Faculty of Artificial Intelligence and Data Science, Universitas Pelita Harapan, Jakarta, Indonesia

Article Info

Article history:

Received May 4, 2025
Revised Apr 23, 2026
Accepted May 22, 2026

Keywords:

AIOps
Anomaly
Container
Process mining
Virtual environment

ABSTRACT

The growing adoption of container-based virtual environments in cloud computing introduces new challenges in anomaly detection due to the systems' dynamic nature. Traditional monitoring approaches often fail to capture inefficiencies or security risks effectively. This study proposes a process mining approach to identify anomalies by analyzing event logs from containerized systems. Event data from the AIOps challenge 2020 was converted into extensible event stream (XES) format and processed using the inductive visual miner in the ProM tool to generate Petri Net models, accurately visualizing container activity flows. A conformance checking analysis was conducted to evaluate the alignment between modeled and actual behavior. Results demonstrated a high fitness score, confirming the model's precision in reflecting true operational processes and its ability to reveal minor deviations indicative of potential anomalies. These findings highlight process mining as a promising method to enhance security, transparency, and performance monitoring in virtual environments. The research also recommends integrating process mining with real-time monitoring systems for proactive anomaly detection, thereby improving responsiveness and resilience in cloud-based infrastructures.

Corresponding Author:

I Made Murwantara
Master of Informatics, Faculty of Artificial Intelligence and Data Science, Universitas Pelita Harapan
Jakarta, Indonesia
Email: made.murwantara@uph.edu

1. INTRODUCTION

Containerization has become a fundamental technology in modern cloud computing [1], [2], offering scalable and efficient solutions for application deployment. Containers provide lightweight virtualization, enabling faster provisioning and improved resource utilization. In researchers [3], [4] concern about container performances within virtualization environment because of low packet processing. However, their dynamic and distributed nature poses challenges in performance monitoring [5], anomaly detection, scheduling [6], and process optimization. Traditional monitoring tools often fail to capture hidden inefficiencies, bottlenecks, and security risks within containerized environments [7], [8], making it crucial to explore new methodologies for comprehensive analysis.

Process mining [9] has emerged as a powerful approach for analyzing event logs to gain insights into system behavior, uncover deviations, and optimize business process models. By applying process mining techniques, organizations can systematically evaluate containerized workflows, detect inefficiencies, and enhance overall system performance. This study focuses on leveraging process mining, particularly the inductive miner [10] technique, to analyze event logs, identify anomalies, and improve operational efficiency

within virtualized container environments.

An anomaly [11], [12], also referred to as an outlier, is a data point or pattern that deviates significantly from the expected distribution within a dataset. These deviations are of particular interest as they may indicate critical phenomena such as fraudulent activities, system malfunctions, or security breaches. Anomaly detection is the process of identifying such irregularities using a range of methodologies, including statistical analysis, machine learning algorithms, and process mining techniques. However, in high-dimensional datasets, the detection process becomes increasingly complex due to the "curse of dimensionality," which leads to data sparsity and challenges in distinguishing genuine anomalies from noise. To address these limitations, effective strategies such as feature selection, dimensionality reduction, and specialized anomaly detection techniques are essential to enhance detection accuracy and reliability [13].

The primary objective of this research is to bridge the gap between anomaly detection and performance optimization [14] in containerized environments. By identifying deviations from expected workflows, this study aims to enhance system reliability, mitigate risks, and provide actionable insights for optimizing business process models. The ability to detect and address anomalies not only strengthens security and enables organizations to improve resource allocation, workload balancing, and process efficiency.

2. METHOD

2.1. Research design

This research adopts a quantitative methodology, utilizing process mining techniques to identify anomalies within virtual container environments. Event log data, representing the execution of containerized processes, is systematically collected and prepared for analysis. The inductive visual miner method, implemented through the ProM framework, is employed to transform the event logs into process models that accurately capture the actual behavior of the system. By constructing detailed Petri Net models, the study visualizes the operational flow of container activities and facilitates the identification of deviations from expected behavior. These deviations, often subtle and easily overlooked by traditional monitoring tools, are critical indicators of potential inefficiencies, operational bottlenecks, or security vulnerabilities. Through conformance checking between the derived models and recorded event data, the research evaluates the alignment and uncovers instances where anomalies are present. The findings demonstrate the effectiveness of process mining in providing deep insights into the dynamic and complex nature of containerized virtual environments. Ultimately, this study highlights the value of integrating process mining into the monitoring frameworks of cloud computing systems to enhance transparency, improve performance, and support proactive anomaly detection.

2.2. Data collection

Dataset for this study consists of event logs extracted from virtualized container environments. These logs capture key process execution details, including itemid, name, bomcid, timestamp, value, and cmbdid. Data sets are collected from AIOps challenge 2020 [15].

2.3. Data preprocessing

The collected datasets have been preprocessed to facilitate analysis. The timestamps, initially recorded in UNIX format, have been converted to greenwich mean time (GMT) to ensure consistency and compatibility with analytical methods. Additionally, the value field has been removed due to the presence of null values, which are not suitable for the applied analysis techniques.

2.4. Analysis methods

The analysis is performed using the ProM framework to facilitate process mining and anomaly detection. This process involves leveraging ProM's capabilities to convert CSV data into an extensible event stream (XES) event log, construct a Petri Net from the XES log, and perform conformance checks to enable complete performance analysis. An anomaly, also referred to as an outlier, is a data point or pattern that deviates significantly from the expected distribution within a dataset. These deviations are of particular interest as they may indicate critical phenomena such as fraudulent activities, system malfunctions, or security breaches. Anomaly detection is the process of identifying such irregularities using a range of methodologies, including statistical analysis, machine learning algorithms, and process mining techniques. However, in high-dimensional datasets, the detection process becomes increasingly complex due to the "curse of dimensionality," which leads

to data sparsity and challenges in distinguishing genuine anomalies from noise. To address these limitations, effective strategies such as feature selection, dimensionality reduction, and specialized anomaly detection techniques are essential to enhance detection accuracy and reliability [13].

2.4.1. Inductive visual miner

Process mining is a powerful technique for extracting valuable insights from recorded event data, enabling a more comprehensive understanding of real-world workflows. By analyzing event logs, it facilitates the discovery, evaluation, and optimization of processes across various fields, including business operations, IT infrastructure, and virtual environments. At its core, process mining relies on discovery algorithms to generate process models from event data. However, ensuring these models accurately represent real processes requires multiple iterations of refinement and evaluation — a practice known as process exploration. This phase involves adjusting algorithm parameters to produce a model that correctly reflects the recorded events.

To overcome these limitations, the inductive visual miner [16] was developed as an advanced process exploration tool that combines the strengths of both commercial and academic approaches. By integrating executable semantics with enhanced exploration features, it provides a balance between usability and analytical rigor. In addition, it introduces animation and deviation visualization functionalities, allowing users to observe process flows interactively and identify deviations from expected behavior. These capabilities significantly improve the efficiency of process discovery, making it easier to detect and address anomalies. As process mining continues to evolve, tools such as the inductive visual miner play a crucial role in optimizing workflows, ensuring compliance, and enhancing decision-making across various domains [17]. The inductive visual miner result of dcos_container in this research shown in Figure 1.

3. RESULTS AND DISCUSSION

3.1. Data preparation

The event log data used in this study originates from the AIOps challenge 2020 [15] competition. The dataset contains various types of data, but this research will focus specifically on container data. The provided data is in the form of CSV files, with a sample attached in Table 1. In creating event logs for process mining, as shown in Table 1, the data must include the following attributes: i) name, representing the container performing a specific process, ii) timestamp, recording the time when a process is successfully executed, and iii) bomc_id, representing a single container process that includes multiple events in a structured sequence.

Table 1. Sample log from AIOps challenge 2020

itemid	name	bomc_id	timestamp	value	cmdb_id
9999999996470837	container_fail_percent	ZJ-004-065	1587312034000	0.000000	container_001
9999999996470922	container_fail_percent	ZJ-004-065	1587312035000	0.000000	container_002
9999999996470922	container_fail_percent	ZJ-004-065	1587312092000	0.000000	container_002
9999999996470837	container_fail_percent	ZJ-004-065	1587312092000	0.000000	container_001
9999999996470837	container_fail_percent	ZJ-004-065	1587312143000	0.000000	container_001
9999999996470922	container_fail_percent	ZJ-004-065	1587312143000	0.000000	container_002
9999999996470837	container_fail_percent	ZJ-004-065	1587312211000	0.000000	container_001
9999999996470922	container_fail_percent	ZJ-004-065	1587312263000	0.000000	container_002

In process mining, a case ID uniquely identifies a process instance, grouping related events in a sequential manner. Each event ID represents an activity within the process and is associated with attributes such as timestamp, resource, and activity description. In container environments, the bomc_id serves as the case ID, enabling the tracking of container execution and detecting anomalies. For example, a container (bomc_id = ZJ-004-063) can transition through events such as container_cpu_used, container_thread_idle, and container_session_used, each recorded with timestamps. By structuring event logs this way, process mining tools like ProM can reconstruct workflows and identify deviations from normal behavior, facilitating the detection of performance bottlenecks or security risks in virtualized environments [18].

A sequence of events that represents a single process instance in an event log is referred to as a trace. A trace contains a sequence of events from the initial execution of the process to its completion, such as the trace [abc], which indicates that events a, b, and c are related. $A \rightarrow B \rightarrow C$, indicates that event A must be completed before event B begins and event B must be executed before event C. This sequence ensures that the event log follows the predefined process case in a structured manner [19].

3.2. Process discovery

Process discovery methods derive process models from event logs, typically using Petri Nets, which consist of places and transitions. The model's state is defined by how tokens are distributed, and transitions are triggered based on token presence. Various models may capture different levels of trace behavior, with some failing to generalize properly (overfitting) and others allowing too many variations (underfitting). The objective of process discovery is to generate a well-structured model that effectively represents observed processes while maintaining a balance between accuracy and simplicity [20]. The data from dcos containers are processed through XES event logs and filtered using the inductive visual miner method to generate a Petri Net, as shown in Figure 1.

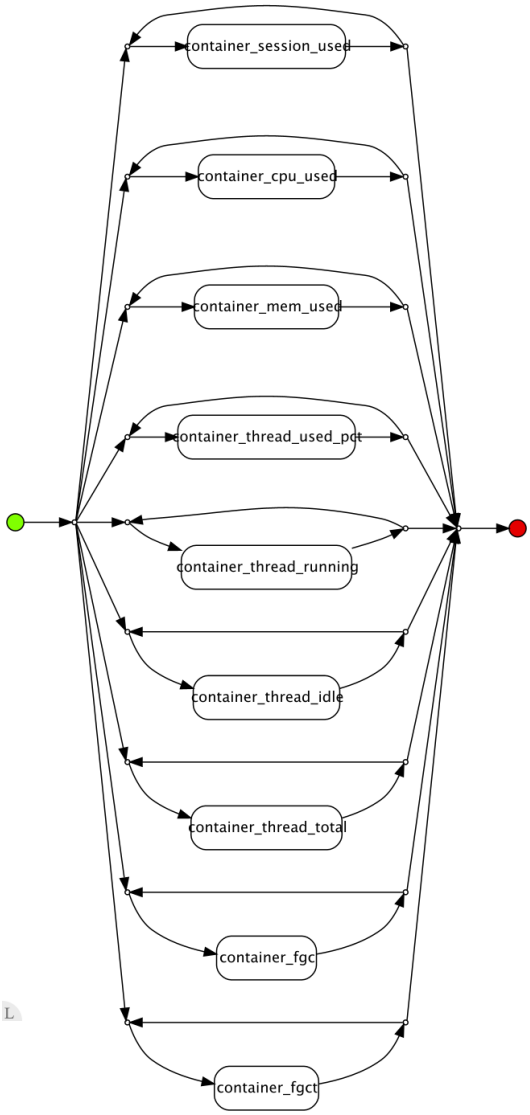

Figure 1. Inductive miner result of dcos_container

3.3. Conformance checking

Conformance checking [21] is a process mining technique used to evaluate the accuracy of an event log aligned with a given process model. It helps identify and measure deviations between the expected and actual process behavior, providing insights into process accuracy and compliance. The main focus of the conformance checking is to assess fitness [22], which determines whether the observed execution traces can be replayed within the model. A model with perfect fitness (value of 1) which means all traces align, while lower

fitness values indicate discrepancies. Various alignment techniques are used to quantify these deviations, such as optimal trace matching, which identifies the corresponding behavior closest to the model for each trace in the event log [23].

To obtain the conformance checking in the ProM application, an XES event log [24] and a Petri Net are required. The Petri Net [25] is obtained during the process mining step using the inductive visual miner, while the XES event log is filtered using the conformance checking filter in the ProM application. The results obtained are as shown in Figure 2.

3.4. Performance analysis

Through the conformance checking results obtained, the process of replaying the log on the net can be performed to evaluate performance. The results of this evaluation can then be further analyzed. The log-replay process, as shown in Figure 3, can be implemented in the ProM application with advanced filtering based on the previous conformance checking results.

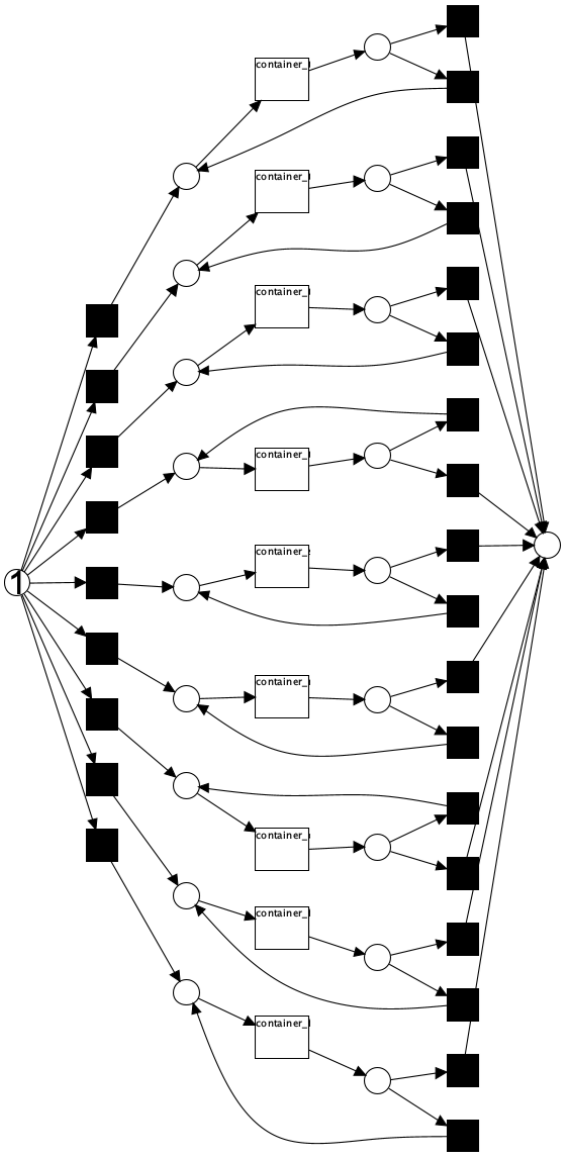

Figure 2. Conformance checking

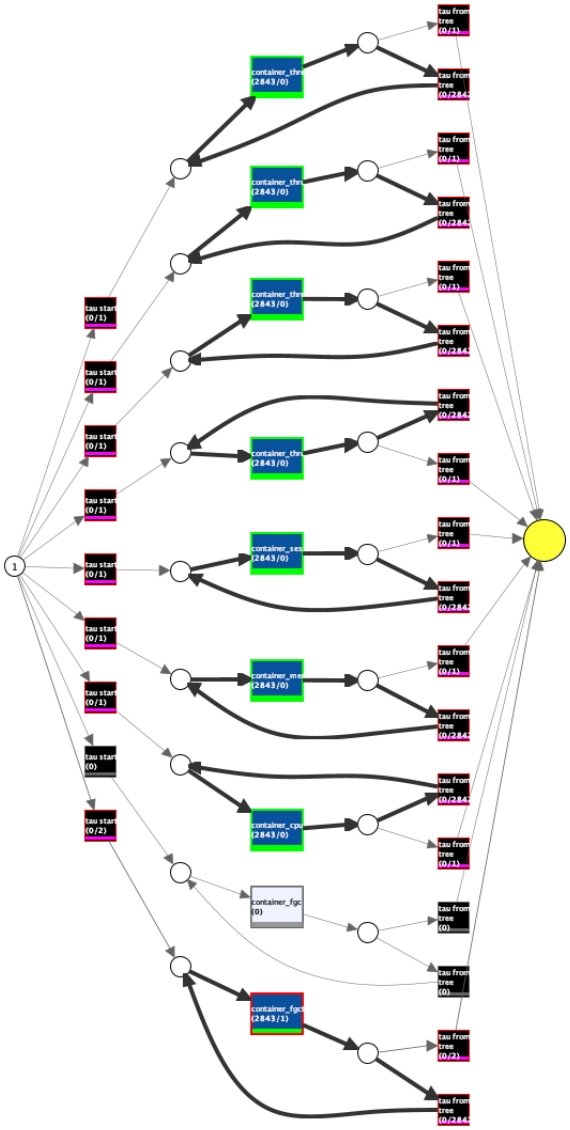

Figure 3. Replay net log dcos container

From a time performance perspective, the replay computation was completed efficiently, requiring approximately 46.66 milliseconds as depicted in Table 2. This result demonstrates that the system is capable of handling the alignment process between the event log and the process model with high computational efficiency. Such a fast computation time suggests the feasibility of applying this method even in environments that require frequent or near real-time conformance checking. However, it is crucial to emphasize that this measurement reflects only the computation time needed for log replay within the ProM tool, not the actual execution or throughput time of the processes being monitored. To obtain deeper insights into process performance, such as identifying potential delays, bottlenecks, or process inefficiencies, a detailed analysis of event timestamp data would be necessary. Incorporating timestamp-based performance analysis would enable the identification of slow process paths and contribute to a more comprehensive evaluation of containerized system behavior.

Table 2. Result replay log

Property	Value
Traversal arcs	19658.66
Calculation time (ms)	46.66
Raw Fitness cost	316
Max move-log cost	2843
Num. states	8255.77
Trace fitness	0.88
Move-model fitness	0.88
Move-log fitness	0.88
Max fitness cost	2844
Trace length	2843
Queued states	14383.55

In terms of cost, the analysis shows a raw fitness cost of 316 out of a maximum possible cost of 2843. This relatively low number of deviations indicates that the model closely follows the behavior recorded in the event log. A lower cost signifies fewer mismatches or corrections needed during the alignment, which implies the model is effective in reflecting the real-world process with minimal inefficiencies.

Looking at the quality dimension, the results are also strong, with trace fitness, move-log fitness, and move-model fitness all scoring approximately 88.89%. These values suggest that most of the observed behavior in the log can be reproduced by the model, showing a high level of conformance. While there are some discrepancies, they are minor and likely represent exceptions or rare process variants not fully captured by the model. Further refinement could involve capturing a broader range of process variations for even higher model fidelity.

4. CONCLUSION

This study successfully demonstrates the application of process mining techniques to detect anomalies in containerized virtual environments. By focusing on container data from the AIOps challenge 2020 and structuring it into event logs with appropriate attributes such as bomc_id, timestamp, and name, the research accurately modeled process behavior using the inductive visual miner in the ProM framework. The creation of Petri Nets enabled a clear visualization of container workflows, while conformance checking validated the alignment between modeled and actual event traces. The system showed strong fitness scores, minimal deviations, and high trace quality, indicating that the model closely mirrors real-world container activities with few inefficiencies. The performance analysis further affirmed the model's computational efficiency and low alignment cost, suggesting its practical viability for real-time monitoring applications. Although minor discrepancies were identified, they are attributed to rare process variations rather than systemic flaws. As seen as the whole perspective, the integration of process discovery, conformance checking, and performance analysis provides a comprehensive approach to enhancing transparency, security, and operational efficiency in virtual container environments. Future research is recommended to incorporate real-time event monitoring and timestamp analysis to further optimize anomaly detection and to explore bottleneck identification in dynamic cloud computing infrastructures.

ACKNOWLEDGMENTS

This work has been supported by the Ministry of Education, Culture, Research and Technology of The Republic of Indonesia and LPPM Universitas Pelita Harapan.

FUNDING INFORMATION

This work has funding supported by the Ministry of Education, Culture, Research, and Technology of Indonesia project, no. 0999/LL3/AL.04/2025 and LPPM Universitas Pelita Harapan project no. 022/LPPM-UPH/VI/2025.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Richard Angkawinata		✓	✓	✓		✓		✓	✓		✓			
I Made Murwantara	✓	✓			✓	✓	✓	✓		✓	✓	✓	✓	✓

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject Administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding Acquisition
Fo	: **Fo**rmal Analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available on request from the corresponding author, [IMM].

REFERENCES

[1] A. Bhowmik, M. Sannigrahi, D. Chowdhury, A. Dey, and S. S. Gill, "CloudAISim: a toolkit for modelling and simulation of modern applications in AI-driven cloud computing environments," *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, vol. 3, no. 4, Dec. 2023, doi: 10.1016/j.tbench.2024.100150.

[2] I. M. Murwantara, B. Bordbar, and J. B. F. Filho, "A self-adaptive architecture with energy management in virtualized environments," in *2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT)*, Sep. 2017, pp. 124–130, doi: 10.1109/ICSIIT.2017.18.

[3] L. Abeni, "Virtualized real-time workloads in containers and virtual machines," *Journal of Systems Architecture*, vol. 154, Sep. 2024, doi: 10.1016/j.sysarc.2024.103238.

[4] F. Wiedner, M. Helm, A. Daichendt, J. Andre, and G. Carle, "Performance evaluation of containers for low-latency packet processing in virtualized network environments," *Performance Evaluation*, vol. 166, Nov. 2024, doi: 10.1016/j.peva.2024.102442.

[5] D. A. Tamburri, M. Miglierina, and E. D. Nitto, "Cloud applications monitoring: an industrial study," *Information and Software Technology*, vol. 127, Nov. 2020, doi: 10.1016/j.infsof.2020.106376.

[6] N. Zhao *et al*., "Scheduling virtual machines and containers: a comparative review of techniques, performance, and future trends," *Journal of Systems Architecture*, vol. 168, Nov. 2025, doi: 10.1016/j.sysarc.2025.103583.

[7] M. Thompson and M. A. Kyer, "Securing the containerized environment along the CI/CD pipeline," in *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2025, pp. 00250–00256, doi: 10.1109/CCWC62904.2025.10903704.

[8] D. Mubanda, N. Mandela, T. Mbinda, and C. Ayesiga, "Evaluating docker container security through penetration testing: a smart computer security," in *2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI)*, Nov. 2023, pp. 415–419, doi: 10.1109/ICCSAI59793.2023.10421124.

[9] W. M. P. V. D. Aalst, H. A. Reijers, and L. Maruster, "Process mining beyond workflows," *Computers in Industry*, vol. 161, Oct. 2024, doi: 10.1016/j.compind.2024.104126.

[10] S. N. Araghi, F. Fontanili, E. Lamine, U. Okongwu, and F. Benaben, "Stable heuristic miner: applying statistical stability to discover the common patient pathways from location event logs," *Intelligent Systems with Applications*, vol. 14, May 2022, doi: 10.1016/j.iswa.2022.200071.

[11] Z. Zhang, Z. Zhu, C. Xu, J. Zhang, and S. Xu, "Towards accurate anomaly detection for cloud system via graph-enhanced contrastive learning," *Complex & Intelligent Systems*, vol. 11, no. 1, Nov. 2024, doi: 10.1007/s40747-024-01659-x.
[12] Y. Song, R. Xin, P. Chen, R. Zhang, J. Chen, and Z. Zhao, "Identifying performance anomalies in fluctuating cloud environments: a robust correlative-GNN-based explainable approach," *Future Generation Computer Systems*, vol. 145, pp. 77–86, Aug. 2023, doi: 10.1016/j.future.2023.03.020.
[13] S. Thudumu, P. Branch, J. Jin, and J. Singh, "A comprehensive survey of anomaly detection techniques for high dimensional big data," *Journal of Big Data*, vol. 7, no. 1, Dec. 2020, doi: 10.1186/s40537-020-00320-x.
[14] G. Soliman, K. Mostafa, and O. Younis, "Reinforcement learning for process mining: business process optimization with avoiding bottlenecks," *Egyptian Informatics Journal*, vol. 29, Mar. 2025, doi: 10.1016/j.eij.2024.100595.
[15] Z. Li, "AIOps challenge 2020 data," *GitHub*, 2022. Accessed: May 26, 2026. [Online]. Available: https://github.com/NetManAIOps/AIOps-Challenge-2020-Data
[16] F. Rabbi, D. Banik, N. U. I. Hossain, and A. Sokolov, "Using process mining algorithms for process improvement in healthcare," *Healthcare Analytics*, vol. 5, Jun. 2024, doi: 10.1016/j.health.2024.100305.
[17] S. J. J. Leemans, D. Fahland, and W. M. P. V. D. Aalst, "Process and deviation exploration with inductive visual miner," in *CEUR Workshop Proceedings*, 2014, pp. 46–50.
[18] P. S. Rawat *et al*., "Efficient virtual machine placement in cloud computing environment using BSO-ANN based hybrid technique," *Alexandria Engineering Journal*, vol. 110, pp. 145–152, Jan. 2025, doi: 10.1016/j.aej.2024.09.103.
[19] Z. Tariq, D. Charles, S. McClean, I. McChesney, and P. Taylor, "Anomaly detection for service-oriented business processes using conformance analysis," *Algorithms*, vol. 15, no. 8, Jul. 2022, doi: 10.3390/a15080257.
[20] W. V. D. Aalst, "Process mining," *Communications of the ACM*, vol. 55, no. 8, pp. 76–83, Aug. 2012, doi: 10.1145/2240236.2240257.
[21] S. J. J. Leemans, "Conformance checking," in *Robust Process Mining with Guarantees: Process Discovery, Conformance Checking and Enhancement*, Cham, Switzerland: Springer, 2022, pp. 327–354, doi: 10.1007/978-3-030-96655-3_7.
[22] J. Carmona, B. van Dongen, and M. Weidlich, "Conformance checking: foundations, milestones and challenges," in *Process Mining Handbook*, Cham, Switzerland: Springer, 2022, pp. 155–190, doi: 10.1007/978-3-031-08848-3_5.
[23] A. Rozinat and W. M. P. V. D. Aalst, "Conformance checking of processes based on monitoring real behavior," *Information Systems*, vol. 33, no. 1, pp. 64–95, Mar. 2008, doi: 10.1016/j.is.2007.07.001.
[24] L. Abb and J.-R. Rehse, "Process-related user interaction logs: state of the art, reference model, and object-centric implementation," *Information Systems*, vol. 124, Sep. 2024, doi: 10.1016/j.is.2024.102386.
[25] T. Brockhoff, M. N. Gose, M. S. Uysal, and W. M. P. V. D. Aalst "Process comparison using Petri Net decomposition," *Application and Theory of Petri Nets and Concurrency*, Cham, Switzerland: Springer, pp. 83–105, doi: 10.1007/978-3-031-61433-0_5

BIOGRAPHIES OF AUTHORS

Richard Angkawinata is a graduate informatics student, Universitas Pelita Harapan. Skilled in fullstack development, data science, and cloud computing system. Strong education professional with a bachelor's degree focused in informatics from Faculty of Information Technology, Universitas Pelita Harapan. He can be contacted at email: 01679230012@student.uph.edu.

I Made Murwantara earned his Ph.D. degree in Computer Science from the University of Birmingham, United Kingdom in 2016. His research interest includes software product line engineering for and in cloud computing, AI for biomedical analysis, AI-based software automation, and cloud computing energy management. He can be contacted at email: made.murwantara@uph.edu.