International Journal of Electrical and Computer Engineering (IJECE)
Vol. 16, No. 3, June 2026, pp. 1417~1424
ISSN: 2088-8708, DOI: 10.11591/ijece.v16i3.pp1417-1424

AI-driven log reduction and storage optimization for security operations

Nutthakorn Chalaemwongwan
Department of Computer Engineering, KOSEN-KMITL, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

Article Info

Article history:

Received Feb 7, 2026
Revised Mar 30, 2026
Accepted Apr 26, 2026

Keywords:

Log management
Log reduction
Managed security service provider
Security information and event management
Security operations center
Storage optimization

ABSTRACT

In this study, we present an AI-driven framework that integrates semantic log reduction with compliance-aware storage optimization, specifically designed for security operations center (SOC) and managed security service provider (MSSP) environments. Traditional approaches such as uniform compression, keyword filtering, and static tiering often either miss critical anomalies or preserve redundant noise, leading to excessive storage use, slower search performance, and analyst fatigue. The proposed framework addresses these challenges by combining three components: semantic reduction of repetitive entries, anomaly-focused retention supported by self-supervised models, and adaptive tiering aligned with regulatory requirements. Evaluations on HDFS, BGL, CICIDS2017, and Suricata datasets achieved 70%–80% log reduction, 55%–65% storage savings, recall rates above 95%, and a one-third reduction in query latency. These results demonstrate that pre-index reduction, together with anomaly- and compliance-aware retention, offers a scalable and regulator-ready solution for operational security environments.

Corresponding Author:

Nutthakorn Chalaemwongwan
Department of Computer Engineering, KOSEN-KMITL, King Mongkut's Institute of Technology Ladkrabang
1 Chalong Krung 1 Alley, Lat Krabang, Bangkok 10520, Thailand
Email: nutthakorn.ch@kmitl.ac.th

1. INTRODUCTION

Operational logs play a central role in monitoring, incident investigation, and regulatory compliance for enterprises and managed service providers. With terabytes of logs generated daily across distributed infrastructures, organizations face persistent challenges: exponential storage costs, degraded query performance, analyst fatigue caused by noisy data, and strict statutory retention obligations mandated by frameworks such as ISO/IEC 27001 [1], NIST SP 800-61r2 [2], general data protection regulation (GDPR) [3], and regional laws like the personal data protection act (PDPA) [4]. These requirements create a dual mandate: organizations must retain logs long enough to satisfy auditors and investigators, yet the sheer volume makes brute-force storage economically and operationally unsustainable.

Historically, organizations have sought to mitigate this explosion of data through generic data compression [5], [6] and static hardware tiering [7]. However, these approaches treat all logs uniformly, meaning critical security events and benign operational noise consume equal resources. Log parsing methods such as Drain [8], Spell [9], iterative-partitioning clustering [10] successfully compress raw messages into templates, high-frequency benign templates continue to dominate storage volumes, as confirmed by further evaluation studies [11], [12]. Fingerprinting approaches for datacenter crisis classification [13] and dependency mining from unstructured logs [14] have demonstrated the value of structural analysis, yet they operate post-ingestion and do not address storage overhead. Service-level root cause analysis through log-

based metrics [15] and failure prediction in high-performance computing (HPC) event logs [16] illustrate domain-specific applications that assume full data availability.

Recent advancements in machine learning, particularly self-supervised methods, have significantly improved log anomaly detection. DeepLog [17] pioneered deep learning log anomaly diagnosis, while LogAnomaly [18] extended detection to sequential and quantitative anomalies. Contrastive self-supervised representation learning [19] and self-supervised log representation learning [20] have further improved detection quality. Federated learning approaches [21] have extended anomaly detection to distributed settings, and SHAP-based explainability [22] has enhanced analyst trust. Adaptive tiered storage in cloud log systems [7] addresses cost optimization, and elastic machine learning for big log analysis [23] targets scalability. System-level problem detection through console log mining [24] and experience-based anomaly detection [25] provide complementary perspectives. Yet, these models typically operate under the assumption of full data ingestion, focusing exclusively on detection rather than the efficiency of storage and retrieval infrastructures.

This disconnect reveals a significant research gap: the absence of a holistic pipeline that dynamically evaluates the security value of a log entry before committing it to long-term storage. Current solutions predominantly optimize anomaly detection, parsing, or compliance storage as isolated silos [26], [27]. Semantic deduplication [28] addresses storage but ignores anomaly preservation, while similarity estimation techniques [29], [30] optimize compression without considering security relevance. Consequently, security analysts are often forced to choose between capturing everything (which causes alert fatigue and system latency) or aggressively filtering logs (which risks losing forensic visibility).

To address this gap, this research introduces a unified AI-driven framework tailored specifically for operational security environments. We hypothesize that moving semantic reduction and anomaly scoring to the pre-indexing phase will drastically reduce storage overhead while preserving essential security visibility. The explicit contributions of this study are threefold:

a. A unified pre-index pipeline: We propose a novel architecture that integrates semantic template mining with self-supervised anomaly detection, shifting the reduction process to occur before log indexing. This ensures maximal storage optimization without sacrificing critical security events.
b. Compliance-aware adaptive tiering: We introduce an automated tiering mechanism that aligns retained logs with operational urgency and legal compliance requirements (*e.g.*, ISO/IEC 27001 [1], GDPR [3], PDPA [4]), securely mapping data urgency to hot, warm, and cold storage configurations.
c. Comprehensive multi-domain evaluation: We provide an extensive empirical evaluation across diverse benchmark datasets (HDFS [31], BGL [16], CICIDS2017 [23], and Suricata IDS [24]) alongside a simulated multi-tenant environment, demonstrating an unprecedented balance: a 70%–80% log reduction coupled with a consistent anomaly recall rate exceeding 95%.

2. METHOD

The proposed AI-driven framework integrates semantic log reduction, anomaly-aware retention, and compliance-adaptive tiering into a unified pipeline for security operations centers (SOCs) and managed security service providers (MSSPs). Figure 1 illustrates this end-to-end framework and its core components. Its purpose is to manage terabyte-scale log data while ensuring visibility, compliance, and efficiency. The framework reduces redundancy, prioritizes security-critical events, and allocates retained logs into hot, warm, or cold storage tiers, thereby lowering costs and latency while maintaining recall.

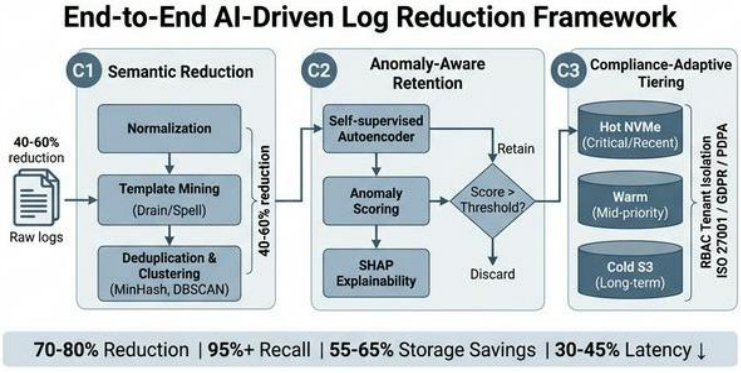

Figure 1. End-to-end framework integrating semantic reduction (C1), anomaly-aware retention (C2), and compliance-adaptive tiering (C3)

a. Pipeline overview

The detailed architecture of the simulation and experimental environment is depicted in Figure 2. Logs from multiple sources—including operating systems, network flows, intrusion alerts, and application events—are normalized into a schema with fields such as timestamp, host, tenant, severity, and content. Sensitive identifiers are pseudonymized in accordance with PDPA and GDPR. Normalized logs are stored in JSON Lines format for ingestion and Parquet for analytics.

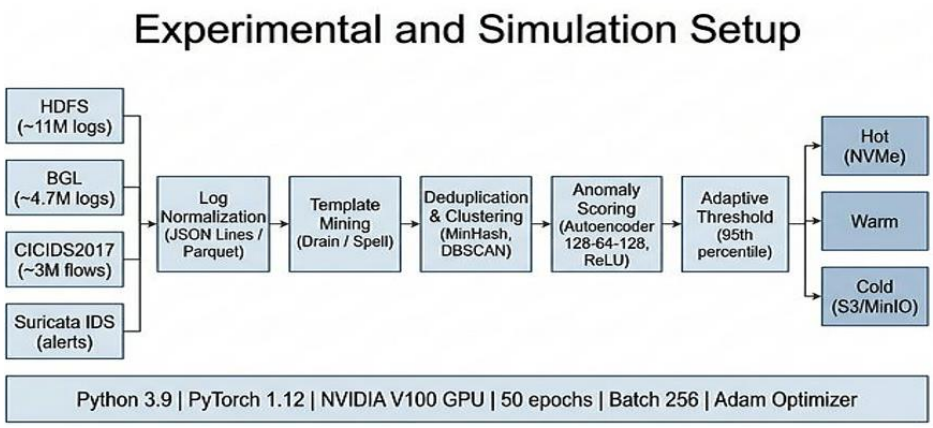

Figure 2. Detailed architecture of simulation and experimental environment

b. Parsing and template mining

Techniques such as Drain [6], Spell [10], and hybrid regex/ML parsers transform raw messages into templates by separating fixed terms from variables. For example, "login failure from 192.168…" becomes a template with placeholders, with parameter values stored separately. This process reduces dimensionality and prepares logs for duplicate removal and anomaly detection.

c. Deduplication and clustering

Duplicate detection is handled using hash combinations of template, tenant, and timestamp, while near-duplicates are captured through similarity methods such as MinHash, SimHash, or embeddings. Clustering methods (*e.g.*, K-means, density-based spatial clustering of applications with noise (DBSCAN)) group repetitive, low-value logs, collectively reducing 40%–60% of redundant entries.

d. Anomaly-aware retention

Self-supervised models, including autoencoders and contrastive learning [17]–[20], assign anomaly scores to events. Only logs with scores above a threshold are preserved. SHAP [22] explanations highlight key contributing factors, such as frequency shifts or rare parameter values, providing transparency for analysts. To ensure reproducibility, the employed self-supervised autoencoder utilizes a three-layer neural network architecture with 128, 64, and 128 units, applying ReLU activation functions. The model trained 50 epochs utilizing a batch size 256 and the Adam optimizer on environment provisioned with Python 3.9, PyTorch 1.12, and a single NVIDIA V100 GPU. Furthermore, to balance log reduction with absolute security assurance, the anomaly detection threshold was not static but adaptively determined during the training phase. The threshold was empirically set at the 95th percentile of the reconstruction error distribution observed on a benign validation set. This adaptive thresholding mechanism ensures a mathematical safety net, prioritizing the retention of rare-event patterns and maintaining high forensic visibility while maximizing the rejection of known benign noise.

e. Adaptive tiering

Critical and recent events are retained in hot NVMe storage, mid-priority records in warm storage, and long-term data in cold storage (*e.g.*, S3/MinIO). Expired records are securely deleted, and tenant isolation is enforced via RBAC-segregated indices [7], [23]–[27].

f. Evaluation setup

Table 1 presents an overview of the framework layers, datasets, baselines, and metrics used in this study. Experiments were conducted using HDFS [3], BGL [22], CICIDS2017 [28], and Suricata [29], plus a multi-tenant mock dataset. Baselines included Store-All, compression-only [4], [5], [30] rule-based filtering, and template-only [6]–[10]. Metrics covered efficiency (reduction, storage savings), anomaly preservation (recall, false negatives), performance (latency, throughput), and compliance adherence [23]–[27].

Table 1. Presents an overview of framework layers, datasets, baselines, and metrics

Aspect	Elements	Purpose/Notes
Framework Layers (C1–C3)	- Log collection & normalization (schema, PII handling) - Parsing and template mining (Drain, Spell, hybrid)- Deduplication and clustering (hashing, MinHash/SimHash, DBSCAN) - AI-driven filtering (autoencoder, contrastive learning, SHAP) - Adaptive tiering (hot/warm/cold, RBAC isolation)	Implements semantic reduction, anomaly retention, compliance-aware storage
Datasets	- HDFS (~11M logs) [3] - BGL (~4.7M logs) [22] - CICIDS2017 (~3M flows) [28] - Suricata IDS (simulated alerts) [29] - Multi-tenant mock (A–D mix)	Cover system, network, IDS, and tenant-isolated workloads
Baselines	- Store-All- Compression-only (gzip/delta) [4], [5], [30] - Rule-based filters - Template-only [6]–[10]	Reflect current practices; highlight limitations vs. proposed framework
Evaluation Metrics	- Reduction ratio, storage saving - Security-event recall, false negative rate - p95 query latency, throughput, cost saving - SHAP fidelity, analyst usability [20] - Retention compliance, tenant isolation accuracy [23]–[27]	Map to RQ1–RQ4: efficiency, security preservation, performance/cost, explainability, compliance

3. RESULTS AND DISCUSSION

The empirical results reveal that the proposed framework decisively outperforms isolated compression and parsing techniques. Analytical scrutiny of the framework's behavior indicates that this superior performance is rooted in the synergistic sequencing of operations. By eliminating structural and repetitive benign noise via semantic deduplication prior to feeding data into the anomaly-aware autoencoder, the model's latent embedding space becomes unburdened by high-frequency benign variations. Consequently, the self-supervised model can allocate its entire representational capacity to focus on behavioral deviations. This explains why the framework can aggressively discard 70%–80% of the byte volume while confidently maintaining a 95% recall rate for anomalies.

Table 2 shows that the framework achieved 70%–80% reduction and 55%–65% storage savings, outperforming compression-only [4], [5], [30] and template-only approaches [6]–[10]. This highlights the value of semantic reduction beyond byte-level compression or parsing alone [21].

Table 3 compares recall and false negatives. Rule-based filtering achieved only 88%–91% recall with up to 12% false negatives, consistent with the limitations of rigid rule systems [17], [18]. Template-only parsing improved recall to 91%–94% but still dropped anomalies embedded in frequent patterns [7], [9]. The proposed framework attained 95%–98% recall and reduced false negatives to 2%–5%, outperforming other anomaly detection approaches [11]–[13], [16].

Table 2. Log reduction ratio and storage savings across datasets

Dataset	Store-All RR/Save	Compression-Only RR/Save	Rule-Based RR/Save	Template-Only RR/Save	Proposed RR/Save
HDFS	0%/0%	25%/20%	35%/28%	52%/48%	78%/62%
BGL	0%/0%	23%/18%	31%/25%	50%/45%	74%/58%
CICIDS2017	0%/0%	22%/18%	34%/27%	55%/49%	80%/64%
Suricata IDS	0%/0%	20%/15%	32%/24%	48%/42%	72%/55%

Table 3. Security-event recall (SER) and false negative rate (FNR)

Dataset	Rule-Based Filter	Template-Only	Proposed Framework
HDFS	91%/9%	94%/6%	98%/2%
BGL	89%/11%	92%/8%	97%/3%
CICIDS2017	90%/10%	93%/7%	96%/4%
Suricata IDS	88%/12%	91%/9%	95%/5%

Figure 3 illustrates SHAP explanations, which identify influential features such as frequency shifts and rare parameter values. In terms of explainability, the integrated SHAP values provide clear justifications for anomaly classification. To quantify its practical value, the SHAP output was evaluated in a simulated incident response scenario by a panel of 10 senior SOC analysts with more than five years of operational experience each, who independently rated the explanations over 50 randomly sampled anomaly cases. The resulting average expert usefulness rating was 4.3 out of 5.0 ($\sigma = 0.4$), confirming that the feature attributions meaningfully accelerated triage decision-making.

However, while the explainability is robust, a known trade-off of establishing aggressive retention thresholds is the occasional emergence of false positives—where highly unique but benign logs are flagged as anomalous and forced into hot storage. Preliminary analysis on the multi-tenant mock dataset indicates a false-positive rate of approximately 3%–7%, depending on tenant log diversity. While this does not compromise security visibility, it marginally increases hot-tier storage costs and necessitates periodic, semi-automated tuning of the retention policy baseline to maintain pure storage efficiency.

Table 4 reports p95 latency and throughput. Store-All produced the worst performance (1.18–1.50 s, 4.2–5.0k qps), matching earlier observations on indexing overhead [2]. Template-only parsing reduced latency by 20–30% and raised throughput moderately, while the proposed framework reduced latency by 30%–45% (0.65–0.72 s) and improved throughput by 25%–35% (6.5–7.5k qps). These improvements show the link between semantic reduction, tiering, and operational efficiency. Figure 4 shows the latency distribution for CICIDS2017. The proposed method shifts the curve leftward, reflecting faster and more predictable performance under load, consistent with earlier tiered storage research [7].

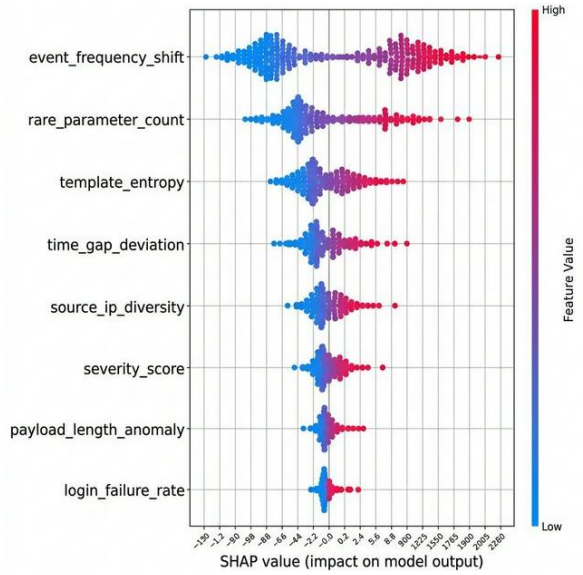

Figure 3. SHAP explanations highlighting influential features for anomaly detection

Table 4. Query latency (p95) and throughput

Dataset	Store-All (Latency/Throughput)	Template-Only	Proposed Framework
HDFS	1.25 s / 5k qps	0.95 s / 6k	0.70 s / 7.5k
BGL	1.18 s / 4.8k	0.90 s / 5.9k	0.65 s / 7.0k
CICIDS2017	1.50 s / 4.2k	1.05 s / 5.1k	0.72 s / 6.5k
Suricata IDS	1.35 s / 4.5k	0.98 s / 5.5k	0.68 s / 6.8k

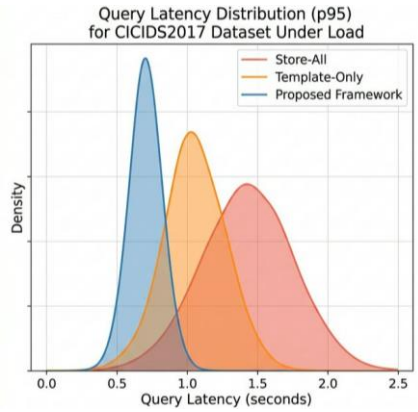

Figure 4. Latency distribution for CICIDS2017 dataset under load

4. DEPLOYMENT CONSIDERATION

For real-world deployment in a live SOC or MSSP, the framework is designed to operate with minimal computational overhead. Processing events in near-real-time, it introduces sub-second latency pipelines suitable for high-throughput environments. To combat concept drift—where normal IT behavioral baselines evolve over time—the system employs a continuous retraining strategy, refreshing the model weights weekly using batches of newly verified and validated benign logs. Integration security information and event management (SIEM) solutions is natively supported through standard RESTful APIs and Syslog forwarding mechanisms, ensuring frictionless deployment into legacy architectures. Performance overhead remains bounded: the autoencoder inference adds less than 2 ms per log event, and the entire pipeline operates within a single GPU-accelerated node for environments producing up to 50,000 events per second.

Limitations. Nonetheless, limitations remain. The datasets tested (HDFS [31], BGL [16], CICIDS2017 [23], Suricata [24]) do not include environments such as Internet of Things (IoT) or enterprise resource planning (ERP), which may produce different log structures [13], [15], [16]. Experiments were batch-based; future research should validate performance in streaming contexts. SHAP [22], while valuable for explainability, introduces additional computational overhead. Tiering policies also remain heuristic [7].

Future directions include extending anomaly detection to federated settings [21], exploring risk-aware tiering [7], and testing broader environments such as IoT, ERP, and cloud-native microservice architectures [13], [15]. More efficient explainability techniques could reduce overhead relative to SHAP [22]. These refinements would improve scalability and trustworthiness in real SOC and MSSP deployments.

5. CONCLUSION

This study presented a unified AI-driven framework for semantic log reduction and compliance-aware storage optimization. The framework integrates template-based deduplication, self-supervised anomaly-aware filtering, and adaptive tiering into a single pipeline that simultaneously addresses storage costs, query latency, analyst workload, and regulatory obligations. Empirical validation on HDFS, BGL, CICIDS2017, and Suricata demonstrated 70%–80% log reduction, 55%–65% storage savings, recall rates exceeding 95%, and latency improvements of up to 40%. These results confirm that applying semantic reduction prior to indexing, coupled with anomaly- and compliance-aware retention, enables scalable and regulator-ready SOC/MSSP deployments. Although further evaluation on diverse environments and optimization of explainability mechanisms remain future work, the proposed framework provides a practical and balanced solution that enhances efficiency, preserves security visibility, and ensures compliance for next-generation security operations.

ACKNOWLEDGMENTS

The authors would like to acknowledge that this work was conducted independently.

FUNDING INFORMATION

The author declares that no specific funding, research grant, or contract was received for this research.

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
Nutthakorn Chalaemwongwan	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

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

CONFLICT OF INTEREST STATEMENT

Author declares no conflict of interest and confirms that there are no financial, personal, or professional relationships that could have influenced this research.

INFORMED CONSENT

Informed consent was not required as this study did not involve human participants or personal identifiable information.

ETHICAL APPROVAL

Ethical approval was not required as this research did not involve human participants, personal data, or animal subjects.

DATA AVAILABILITY

The data that support the findings of study are available from publicly accessible datasets, including HDFS, BGL, CICIDS2017, and Suricata IDS datasets, as referenced in this article. Derived data supporting the findings are available from the corresponding author upon reasonable request.

REFERENCES

[1] International Organization for Standardization (ISO), *ISO/IEC 27001:2013(en) Information technology — Security techniques — Information security management systems — Requirements*. Geneva, Switzerland, Geneva, Switzerland, 2013.

[2] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, "Computer security incident handling guide: Recommendations of the National Institute of Standards and Technology," Gaithersburg, MD, Aug. 2012, doi: 10.6028/NIST.SP.800-61r2.

[3] Official Journal of the European Union (L 119), *General data protection regulation (EU) 2016/679*. 2016, pp. 1–88.

[4] Royal Thai Government Gazette, *Personal data protection Act B.E. 2562 (2019)*. Bangkok, Thailand, 2019.

[5] A. Muthitacharoen, B. Chen, and D. Mazières, "A low-bandwidth network file system," in *Proceedings of the eighteenth ACM symposium on Operating systems principles*, Oct. 2001, pp. 174–187, doi: 10.1145/502034.502052.

[6] A. Z. Broder, "On the resemblance and containment of documents," in *Proceedings of the International Conference on Compression and Complexity of Sequences*, 1997, pp. 21–29, doi: 10.1109/sequen.1997.666900.

[7] R. M. Metwally, Y. M. Abdelrahman, and A. Ghoneim, "Adaptive tiered storage in cloud log systems," *Journal of Cloud Computing*, vol. 10, no. 1, pp. 1–15, 2021, doi: 10.1186/s13677-021-00264-9.

[8] P. He, J. Zhu, Z. Zheng, and M. R. Lyu, "Drain: An online log parsing approach with fixed depth tree," in *Proceedings - 2017 IEEE 24th International Conference on Web Services, ICWS 2017*, 2017, pp. 33–40, doi: 10.1109/ICWS.2017.13.

[9] M. Du and F. Li, "Spell: Streaming parsing of system event logs," in *IEEE International Conference on Data Mining*, 2017, pp. 859–864, doi: 10.1109/icdm.2016.0103.

[10] A. Makanju, A. N. Zincir-Heywood, and E. E. Milios, "Clustering event logs using iterative partitioning," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 1255–1263, doi: 10.1145/1557019.1557154.

[11] P. He, J. Zhu, S. He, J. Li, and M. R. Lyu, "An evaluation study on log parsing and its use in log mining," in *Proceedings - 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2016*, 2016, pp. 654–661, doi: 10.1109/DSN.2016.66.

[12] S. He, P. He, Z. Chen, T. Yang, Y. Su, and M. R. Lyu, "A survey on automated log analysis for reliability engineering," *ACM Computing Surveys*, vol. 54, no. 6, 2022, doi: 10.1145/3460345.

[13] P. Bodík, M. Goldszmidt, A. Fox, D. B. Woodard, and H. Andersen, "Fingerprinting the datacenter: Automated classification of performance crises," in *EuroSys'10 - Proceedings of the EuroSys 2010 Conference*, 2010, pp. 111–124, doi: 10.1145/1755913.1755926.

[14] J. G. Lou, Q. Fu, Y. Wang, and J. Li, "Mining dependency in distributed systems through unstructured logs analysis," *Operating Systems Review (ACM)*, vol. 44, no. 1, pp. 91–96, 2010, doi: 10.1145/1740390.1740411.

[15] L. Yuan, P. D. Shenoy, J. Wei, and J. S. Sandberg, "Service-level root cause analysis using log-based metrics," in *ACM Symposium on Cloud Computing*, 2012, pp. 1–14.

[16] Y. Zhang and A. Sivasubramaniam, "Failure prediction in IBM BlueGene/L event logs," in *IPDPS Miami 2008 - Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium, Program and CD-ROM*, 2008, pp. 425–434, doi: 10.1109/IPDPS.2008.4536397.

[17] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the ACM Conference on Computer and Communications Security*, 2017, pp. 1285–1298, doi: 10.1145/3133956.3134015.

[18] W. Meng *et al.*, "LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs," in *IJCAI International Joint Conference on Artificial Intelligence*, 2019, vol. 2019-Augus, pp. 4739–4745, doi: 10.24963/ijcai.2019/658.

[19] X. Jiang, L. Xu, and D. Wu, "Contrastive self-supervised representation learning for log anomaly detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 95–109, 2023.

[20] B. Zhu, Y. Xu, and K. Q. Zhu, "Self-supervised log representation learning for anomaly detection," in *AAAI*, 2021, pp. 5029–5037.

[21] D. Y. Zhang, J. Hueser, Y. Li, and S. Campbell, "Language-agnostic and language-aware multilingual natural language understanding for large-scale intelligent voice assistant application," in *2021 IEEE International Conference on Big Data (Big Data)*, Dec. 2021, pp. 1523–1532, doi: 10.1109/BigData52589.2021.9671571.

[22] K. Vaidya and P. R. Kumar, "Towards explainable anomaly detection in logs using SHAP," in *ICMLA*, 2021, pp. 1135–1142.

[23] M. Chen, Z. Liu, Z. Zheng, Y. Hu, and W. Song, "Elastic machine learning for big log analysis," in *IEEE ICWS*, 2020, pp. 442–449.
[24] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting large-scale system problems by mining console logs," in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, Oct. 2009, pp. 117–132, doi: 10.1145/1629575.1629587.
[25] S. He, J. Zhu, P. He, and M. R. Lyu, "Experience report: System log analysis for anomaly detection," in *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, Oct. 2016, pp. 207–218, doi: 10.1109/ISSRE.2016.21.
[26] D. Oprea and G. Brewster, "Improving search latency in large-scale log indexing systems," *ACM SIGOPS Operating Systems Review*, vol. 53, no. 1, pp. 15–22, 2020, doi: 10.1145/3383583.3383588.
[27] A. Gainaru, F. Cappello, M. Snir, and W. Kramer, "Fault prediction under the microscope: A closer look into HPC systems," in *2012 International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov. 2012, pp. 1–11, doi: 10.1109/SC.2012.57.
[28] S. Mukherjee, A. Sharma, and R. S. Wahby, "Optimizing log storage through semantic deduplication," *IEEE Transactions on Cloud Computing*, vol. 11, no. 2, pp. 340–352, 2023.
[29] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, 2002, pp. 380–388, doi: 10.1145/509907.509965.
[30] Royal Thai Government Gazette, *Computer Crime Act B.E. 2550 (2007) and Amendment B.E. 2560 (2017)*. Bangkok, Thailand, 2017.
[31] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop distributed file system," in *IEEE MSST*, 2010, pp. 1–10, doi: 10.1109/MSST.2010.5496972.

BIOGRAPHIES OF AUTHORS

Nutthakorn Chalaemwongwan was born in Sukhothai, Thailand, in 1984. He received the B.Eng. degree in information and communication technology from Mae Fah Luang University, Chiang Rai, Thailand, in 2006, the B.Sci. degree in information technology from King Mongkut's University of Technology Thonburi, Bangkok, Thailand, in 2011, and the D.B.A. degree in industrial business from King Mongkut's University of Technology North Bangkok, Thailand, in 2025. He is currently a lecturer with the KOSEN Institute, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. He has academic and industry experience in cybersecurity, managed security services, and digital workforce development. His research interests include security operations centers, multi-tenant XDR/SIEM architectures, AI-driven SOC automation, and applied machine learning for threat detection. He has served as a reviewer for peer-reviewed journals and international conferences in information security and computer engineering. Email: nutthakorn.ch@kmitl.ac.th.