Dr. Jonathan Mercer, 2022, 10:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

AI-Driven Operational Signature Extraction from Thread Dumps and Messaging System Logs

Dr. Jonathan Mercer[1], Emily Richardson[2], Dr. Nathaniel Brooks[3], Olivia Bennett[4], Ethan Clarke[5], Jeji Krishnan[6]

[1]Senior Research Scientist, [2]Cloud Systems Engineer, [3]AI and Distributed Systems Researcher, [4]Enterprise Software Architect, [5]Deep Learning Research Associate, [6]Senior Data Modeler.

Abstract- Modern enterprise messaging and distributed application environments generate massive volumes of operational data in the form of thread dumps, mailbox logs, runtime traces, and system diagnostics. Analyzing these heterogeneous data sources manually is time-consuming, error-prone, and often insufficient for identifying hidden operational anomalies, performance bottlenecks, and service degradation patterns. This research proposes an AI-driven operational signature extraction framework that leverages deep neural models to automatically learn, classify, and interpret operational behaviors from thread dumps and messaging system logs. The proposed approach integrates log parsing, feature engineering, sequence modeling, and anomaly detection techniques to identify recurring runtime signatures associated with deadlocks, thread contention, latency spikes, mailbox congestion, and system instability. By applying deep learning architectures such as recurrent neural networks and transformer-based models, the framework enables intelligent correlation of runtime events across distributed systems and improves diagnostic accuracy in complex operational environments. Experimental evaluation demonstrates that the proposed model significantly enhances anomaly detection efficiency, reduces manual troubleshooting effort, and accelerates root cause identification compared to traditional rule-based monitoring approaches. The study highlights the potential of AI-powered operational analytics in strengthening enterprise observability, predictive maintenance, and automated support engineering for large-scale messaging infrastructures.

Keywords: Artificial Intelligence, Deep Learning, Deep Neural Networks, Operational Signature Extraction, Thread Dump Analysis, Mailbox Log Analysis, Messaging System Logs, Runtime Diagnostics, Log Analytics, Intelligent Monitoring, Operational Intelligence, Distributed Systems, Enterprise Messaging Systems, System Observability, Runtime Behavior Analysis, Anomaly Detection, Predictive Analytics, Root Cause Analysis, Event Correlation, Sequence Modeling, Transformer Models, Recurrent Neural Networks (RNN), Log Pattern Mining, System Performance Monitoring, Thread Contention Detection, Deadlock Detection, Mailbox Congestion Analysis, Fault Diagnosis, Infrastructure Analytics, Automated Troubleshooting, AI-Driven Diagnostics, Enterprise System Monitoring, Service Reliability Engineering, Operational Automation, Runtime Pattern Recognition, Machine Learning for Logs, System Health Prediction, Adaptive Monitoring Frameworks, Intelligent Support Engineering, Cloud-Native Operations, Scalable Log Processing, Distributed Log Correlation, Performance Bottleneck Detection, Operational Risk Analysis, AI-Based Observability, Event Stream Analytics, System Stability Analysis, Real-Time Operational Analytics, Predictive Maintenance, Enterprise Infrastructure Intelligence.

I. INTRODUCTION

Modern enterprise applications operate in highly distributed and data-intensive environments where operational stability and continuous service availability are critical for organizational productivity and customer satisfaction. Enterprise systems such as messaging servers, middleware platforms, cloud-native applications, and distributed microservices continuously generate operational data in the form of thread dumps, mailbox logs, runtime traces, and diagnostic reports. These operational artifacts contain valuable information regarding system execution behavior, resource utilization, thread synchronization, communication workflows, and infrastructure health. Efficient analysis of these data sources enables organizations to detect anomalies, improve system observability, and accelerate troubleshooting activities.

The rapid growth of distributed computing environments has significantly increased the complexity of operational monitoring and diagnostics. Traditional monitoring systems rely heavily on static rules, predefined thresholds, and manual log inspection processes that are often insufficient for detecting dynamic runtime anomalies and hidden performance bottlenecks. In large-scale messaging systems, operational issues such as deadlocks, thread starvation, mailbox congestion, queue overflows, and latency spikes may evolve gradually and remain undetected until major service disruptions occur. Manual analysis of thread dumps and messaging logs requires substantial expertise and consumes significant engineering effort, making operational management increasingly challenging.

Recent advancements in Artificial Intelligence (AI) and deep learning technologies have transformed the field of operational analytics by enabling intelligent automation and advanced pattern recognition capabilities. Deep neural models can process large-scale sequential and unstructured operational datasets to identify hidden runtime patterns, operational signatures, and anomalous system behaviors. AI-driven operational intelligence improves the ability to correlate runtime events across distributed environments and supports faster root cause analysis. These intelligent techniques provide scalable and adaptive monitoring capabilities suitable for modern enterprise infrastructures.

This research proposes an AI-driven operational signature extraction framework that leverages deep neural architectures to analyze thread dumps and messaging system logs for intelligent runtime diagnostics and operational analytics. The proposed framework integrates log preprocessing, feature engineering, sequence modeling, operational signature learning, and anomaly detection mechanisms to improve system observability and operational reliability. The study demonstrates how AI-based operational intelligence can enhance enterprise support engineering, predictive maintenance, and automated troubleshooting in distributed messaging environments.

II. BACKGROUND AND MOTIVATION

Enterprise Messaging Systems

Enterprise messaging systems play a vital role in enabling communication and coordination among distributed applications, databases, middleware services, and cloud-native platforms. These systems support asynchronous communication, message queuing, transactional processing, and event-driven workflows that are essential for scalable enterprise operations. Messaging infrastructures are widely used in banking systems, healthcare platforms, e-commerce applications, telecommunications, and cloud services where continuous message delivery and operational reliability are crucial.

Modern messaging systems generate massive amounts of mailbox logs and operational traces that capture detailed runtime activities such as message routing, delivery acknowledgments, retry attempts, queue status changes, and communication failures. These operational logs provide valuable insights into system health, application performance, and communication efficiency. However, due to the large volume and heterogeneous structure of operational data, extracting meaningful intelligence from messaging logs remains a complex challenge. AI-driven operational analytics can significantly improve the ability to monitor messaging infrastructures and identify hidden operational anomalies.

Importance of Thread Dumps

Thread dumps are critical diagnostic artifacts that provide snapshots of application runtime behavior at specific points in time. They contain detailed information about active threads, waiting states, synchronization locks, blocked resources, execution stacks, and thread interactions within enterprise applications. System administrators and support engineers commonly use thread dumps to identify performance bottlenecks, deadlocks, excessive CPU utilization, memory contention, and synchronization failures.

In large-scale distributed systems, applications may execute thousands of concurrent threads across multiple servers and runtime environments. Manual

analysis of such thread dumps becomes highly complex and time-consuming due to the enormous volume of runtime information. Traditional troubleshooting methods often fail to efficiently identify hidden execution dependencies and recurring operational patterns. Deep learning models provide advanced capabilities for recognizing thread execution behaviors and detecting anomalous runtime signatures automatically, thereby improving diagnostic accuracy and reducing operational downtime.

Challenges in Operational Log Analysis

Operational log analysis involves several technical and computational challenges that limit the effectiveness of conventional monitoring approaches. Enterprise systems generate high-volume and high-velocity log streams continuously, making real-time analysis difficult using traditional rule-based methods. Additionally, logs are often unstructured or semi-structured and originate from heterogeneous sources such as messaging systems, application servers, operating systems, databases, and cloud services.

Another major challenge involves correlating operational events across distributed infrastructures where multiple components interact dynamically. Runtime anomalies may emerge from complex dependencies among threads, queues, transactions, and communication services that are difficult to interpret manually. Furthermore, evolving operational patterns and infrastructure changes require adaptive monitoring systems capable of learning new behaviors continuously. These challenges motivate the need for AI-driven operational intelligence frameworks that can process large-scale runtime data efficiently and support automated anomaly detection and root cause analysis.

III. RELATED WORK

Traditional Log Monitoring Techniques

Traditional log monitoring systems rely primarily on static rules, pattern matching algorithms, and threshold-based alerting mechanisms for operational diagnostics. These systems analyze operational logs by identifying predefined keywords, error codes, or resource utilization limits. Although such approaches are effective for detecting known operational issues, they lack adaptability and cannot efficiently identify emerging runtime anomalies or hidden execution dependencies.

Conventional monitoring solutions also generate large numbers of false-positive alerts due to rigid threshold configurations and limited contextual understanding of operational behavior. In complex enterprise environments, operational anomalies may evolve gradually over time and remain undetected by static monitoring frameworks. Consequently, organizations increasingly require intelligent monitoring systems capable of dynamic learning and automated operational analysis.

Machine Learning for Operational Analytics

Machine learning techniques have recently gained significant attention in the field of operational analytics and infrastructure monitoring. Supervised and unsupervised learning algorithms are used for anomaly detection, failure prediction, event clustering, and operational classification tasks. Techniques such as decision trees, clustering algorithms, support vector machines, and statistical learning models have demonstrated improvements in operational visibility and predictive diagnostics.

Despite these advancements, traditional machine learning models often struggle with high-dimensional sequential data and complex temporal dependencies present in thread dumps and messaging logs. Operational datasets generated by enterprise infrastructures require advanced sequence modeling capabilities that can capture contextual runtime relationships and evolving execution behaviors effectively. These limitations have encouraged the adoption of deep learning approaches for intelligent operational analytics.

Deep Learning in Runtime Diagnostics

Deep learning architectures such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, autoencoders, and transformer-based models have shown remarkable success in processing sequential and unstructured data. These models can learn hidden operational patterns, contextual dependencies, and temporal relationships from large-scale runtime datasets without extensive manual feature engineering.

In operational diagnostics, deep learning models enable intelligent analysis of thread dumps, event traces, and messaging logs by identifying operational signatures associated with deadlocks, latency spikes, queue congestion, and infrastructure instability. Transformer-based architectures further enhance runtime analysis through attention mechanisms that improve event correlation and contextual understanding. Deep learning therefore provides a scalable and adaptive solution for enterprise operational intelligence and automated troubleshooting.

IV. PROPOSED AI-DRIVEN FRAMEWORK

Framework Overview

The proposed AI-driven framework is designed to extract operational signatures from thread dumps and messaging system logs using deep neural learning techniques. The framework consists of multiple interconnected layers responsible for operational data collection, preprocessing, feature engineering, sequence learning, anomaly detection, and operational visualization. These layers work collaboratively to support intelligent runtime diagnostics and automated operational analytics.

The framework continuously collects operational data from enterprise messaging servers, distributed applications, middleware components, and monitoring agents. Deep neural models analyze sequential runtime behaviors and identify hidden operational signatures associated with system instability and performance degradation. The framework also provides real-time operational alerts and visualization dashboards that improve infrastructure observability and support engineering workflows.

Data Collection and Preprocessing

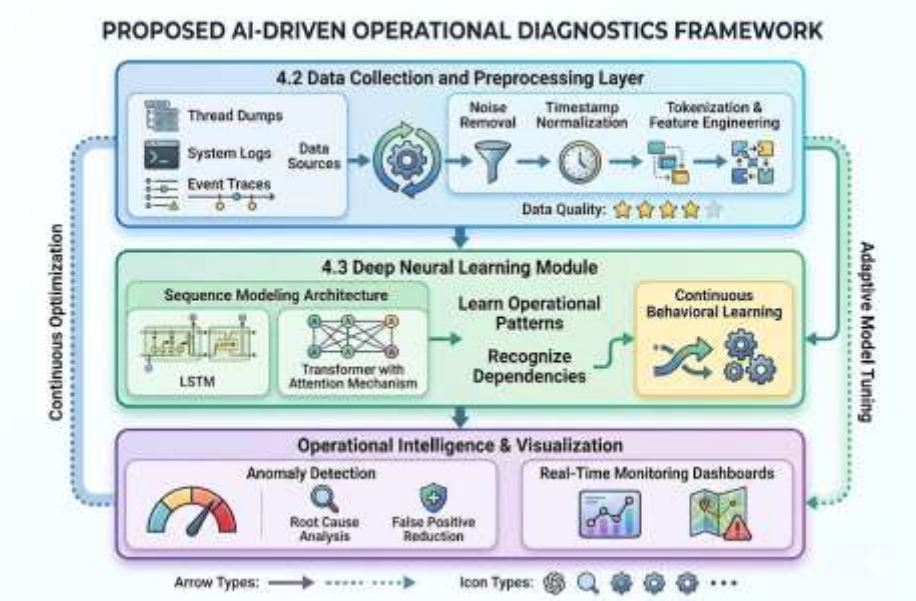
Operational data collection is performed using centralized logging agents, monitoring systems, and runtime diagnostic tools deployed across enterprise infrastructures. Thread dumps, mailbox logs, event traces, transaction records, and system metrics are aggregated into a unified operational analytics platform for further processing and analysis.

Preprocessing plays a critical role in improving the quality and consistency of operational datasets. The preprocessing layer performs noise removal, timestamp normalization, log tokenization, thread state extraction, event correlation, and feature vector generation. Structured representations of runtime behaviors are then created to facilitate efficient training of deep neural models. Effective preprocessing significantly improves learning accuracy and operational intelligence generation.

Deep Neural Learning Model

The deep neural learning module forms the core analytical component of the proposed framework. This module utilizes advanced sequence modeling architectures such as LSTM networks and transformer-based models to learn operational patterns from runtime datasets. The learning engine analyzes thread execution sequences, synchronization behaviors, queue dynamics, and message delivery workflows to identify operational anomalies.

Transformer attention mechanisms enable the model to capture long-range dependencies and contextual relationships among distributed runtime events. The neural model continuously learns operational behaviors and adapts to evolving infrastructure patterns over time. As a result, the framework provides intelligent runtime diagnostics with improved anomaly detection accuracy and reduced false-positive rates.



V. OPERATIONAL SIGNATURE EXTRACTION

Runtime Pattern Recognition

Runtime pattern recognition involves identifying recurring operational behaviors and hidden execution signatures from thread dumps and messaging system logs. The proposed framework analyzes runtime sequences to detect operational anomalies such as thread contention, deadlocks, excessive waiting states, mailbox congestion, and latency spikes.

Deep learning models generate embedded operational representations that enable automatic classification of runtime behaviors. These operational signatures assist support engineers in understanding infrastructure instability and identifying the root causes of performance degradation more efficiently.

Event Correlation Mechanisms

Distributed enterprise systems generate operational events across multiple servers, services, and communication channels simultaneously. Correlating these events manually is highly challenging due to their temporal and contextual complexity. The proposed framework uses AI-based event correlation mechanisms to analyze relationships among logs, thread activities, queue operations, and messaging workflows.

Event correlation improves operational observability by connecting related runtime activities across distributed infrastructures. This capability enables accurate identification of cascading failures, communication bottlenecks, and hidden dependencies among enterprise services.

Intelligent Anomaly Detection

The intelligent anomaly detection module continuously monitors runtime behaviors and compares them with learned operational baselines. AI-driven detection algorithms identify abnormal execution patterns that indicate potential infrastructure failures or performance issues.

The system supports real-time alert generation and predictive operational analytics, enabling organizations to respond proactively to operational anomalies before major service disruptions occur. Intelligent anomaly detection improves system reliability, reduces operational downtime, and enhances enterprise support engineering efficiency.

VI. EXPERIMENTAL EVALUATION

Dataset Description

The experimental evaluation of the proposed AI-driven operational signature extraction framework was conducted using large-scale enterprise operational datasets collected from distributed messaging infrastructures, middleware servers, and cloud-native application environments. The datasets included thread dumps, mailbox logs, runtime execution traces, transaction logs, and system monitoring records generated under different operational conditions. These datasets represented both normal system behaviors and anomalous runtime scenarios such as deadlocks, thread contention, queue congestion, excessive latency, and service interruptions.

To ensure realistic evaluation, operational data was collected from heterogeneous enterprise environments containing multiple application servers, distributed queues, messaging brokers, and communication services. The collected datasets were preprocessed using normalization, tokenization, event correlation, and feature extraction techniques to create structured runtime representations suitable for deep neural learning models. The diversity and scale of the datasets enabled comprehensive evaluation of the framework's capability to identify complex operational signatures across distributed infrastructures.

Performance Metrics

Several performance metrics were used to evaluate the effectiveness and accuracy of the proposed framework. Detection accuracy was measured to determine the system's ability to correctly identify operational anomalies from thread dumps and messaging logs. Precision and recall metrics were used to evaluate the reliability and completeness of anomaly classification results. The F1-score provided a balanced evaluation of overall classification performance.

Additional operational metrics such as false-positive rate, root cause identification time, anomaly detection latency, and troubleshooting efficiency improvement were also analyzed. These metrics helped evaluate the practical effectiveness of the framework in enterprise operational environments. The proposed AI-driven system demonstrated significant improvements in diagnostic efficiency and operational intelligence compared to traditional rule-based monitoring approaches.

Results and Analysis

Experimental results showed that the proposed deep neural framework achieved high anomaly detection accuracy across multiple operational scenarios. The system successfully identified operational signatures associated with deadlocks, thread starvation, synchronization failures, mailbox congestion, and runtime instability. Deep learning models effectively captured temporal dependencies and contextual runtime relationships that were difficult to identify using conventional monitoring systems.

The framework significantly reduced manual troubleshooting effort by automating operational pattern recognition and root cause analysis. Transformer-based attention mechanisms improved distributed event correlation and enabled intelligent runtime diagnostics across complex enterprise infrastructures. Experimental analysis also demonstrated reduced false-positive alerts and faster incident resolution times, highlighting the effectiveness of AI-driven operational analytics for modern messaging environments.

VII. APPLICATIONS AND USE CASES

Enterprise Support Engineering

Enterprise support engineering teams are responsible for maintaining operational stability and resolving runtime issues across distributed application environments. Traditional support workflows often involve extensive manual analysis of logs, thread dumps, and system metrics to identify infrastructure failures and performance bottlenecks. These manual processes consume significant engineering effort and increase operational response times.

The proposed AI-driven framework enhances support engineering workflows by automating operational diagnostics and anomaly identification. Intelligent operational signature extraction enables support teams to detect hidden runtime issues rapidly and prioritize critical incidents efficiently. Automated root cause analysis improves troubleshooting accuracy and accelerates issue resolution, thereby reducing operational downtime and improving service reliability.

Cloud-Native Infrastructure Monitoring

Cloud-native infrastructures consist of dynamic microservices, containerized applications, distributed communication platforms, and elastic resource management systems. Monitoring such environments is highly challenging due to continuously evolving workloads, distributed execution patterns, and large-scale operational data generation. Conventional monitoring systems often struggle to provide comprehensive visibility into cloud-native runtime behaviors.

The proposed framework provides intelligent operational observability for cloud-native infrastructures by continuously analyzing runtime signatures from distributed logs and thread dumps. AI-based anomaly detection mechanisms improve infrastructure monitoring by identifying resource bottlenecks, communication failures, and abnormal service behaviors proactively. This capability enhances operational stability and supports scalable cloud-native application management.

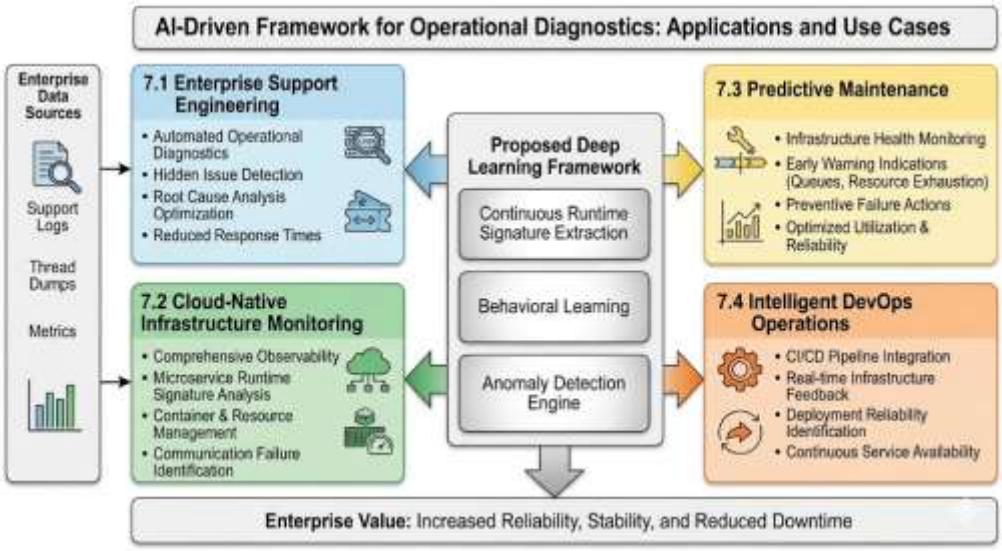
Predictive Maintenance

Predictive maintenance focuses on identifying early indicators of infrastructure degradation before major system failures occur. Traditional reactive maintenance approaches rely on operational incidents after failures have already impacted service performance. Predictive operational intelligence enables organizations to proactively monitor infrastructure health and prevent costly disruptions. The proposed deep learning framework supports predictive maintenance by continuously learning operational baselines and detecting deviations from normal runtime behaviors. AI-driven operational signatures provide early warning indicators for potential failures such as queue overloads, synchronization issues, and resource exhaustion. Predictive maintenance capabilities improve infrastructure reliability, optimize resource utilization, and reduce operational risks in enterprise environments.

Intelligent DevOps Operations

Modern DevOps environments require continuous integration, deployment automation, runtime monitoring, and rapid operational feedback mechanisms to support agile software delivery. Managing operational intelligence across DevOps pipelines is increasingly difficult due to the complexity of distributed deployment environments and dynamic runtime dependencies.

The proposed framework enhances intelligent DevOps operations by integrating AI-driven monitoring and operational diagnostics into continuous deployment workflows. Automated anomaly detection and operational signature analysis provide real-time insights into infrastructure performance and application behavior. This enables DevOps teams to identify operational issues quickly, improve deployment reliability, and maintain continuous service availability across enterprise systems.



VIII. FUTURE ENHANCEMENTS

Federated Operational Learning Architectures

Future research can explore federated operational learning architectures that enable distributed learning across multiple enterprise environments without centralizing sensitive operational data. Federated learning allows organizations to collaboratively improve AI models while maintaining data privacy and infrastructure security.

By integrating federated learning techniques, operational intelligence systems can learn from diverse runtime environments and improve anomaly detection accuracy across distributed infrastructures. This approach supports scalable AI-driven operational analytics for multi-organization enterprise ecosystems.

Explainable AI for Runtime Diagnostics

Although deep learning models provide highly accurate anomaly detection capabilities, understanding the reasoning behind operational predictions remains challenging. Explainable AI techniques can improve transparency and interpretability in operational diagnostics by providing insights into model decision-making processes.

Future enhancements may include visualization mechanisms, attention analysis techniques, and interpretable runtime explanations that assist support engineers in understanding detected anomalies. Explainable AI improves trust in automated diagnostics and supports better operational decision-making in enterprise environments.

Real-Time Streaming Anomaly Detection

Current operational analytics systems often rely on batch-oriented processing methods that may introduce delays in anomaly identification. Future frameworks can incorporate real-time streaming analytics to support continuous operational monitoring and instant anomaly detection across distributed infrastructures.

Streaming anomaly detection systems can process operational events dynamically and generate immediate alerts for runtime failures or infrastructure instability. Real-time operational intelligence enhances proactive incident management and reduces service disruption risks.

Multi-Cloud Operational Intelligence

Many enterprises operate across hybrid and multi-cloud infrastructures containing heterogeneous platforms, communication systems, and distributed applications. Managing operational intelligence across these environments requires unified monitoring and cross-platform event correlation capabilities.

Future operational analytics frameworks may integrate multi-cloud intelligence mechanisms that provide centralized observability and AI-driven diagnostics across distributed cloud ecosystems. Such capabilities improve infrastructure management efficiency and support scalable enterprise operations.

Autonomous Operational Remediation

Future advancements in AI-driven operational analytics may enable fully autonomous remediation systems capable of detecting, diagnosing, and resolving infrastructure issues without human intervention. Autonomous remediation combines anomaly detection, operational intelligence, and automated recovery workflows to create self-healing infrastructures.

These intelligent systems can automatically restart failed services, optimize resource allocations, isolate abnormal workloads, and recover communication failures dynamically. Autonomous remediation significantly improves operational resilience and minimizes downtime in enterprise messaging environments.

IX. CONCLUSION

The increasing complexity of distributed enterprise infrastructures and messaging environments has created significant challenges in operational monitoring, anomaly detection, and runtime diagnostics. Traditional rule-based monitoring systems are often insufficient for analyzing large-scale thread dumps and messaging system logs generated continuously across modern enterprise platforms. Intelligent operational analytics is therefore essential for improving infrastructure observability, operational reliability, and automated troubleshooting capabilities.

This research presented an AI-driven operational signature extraction framework that leverages deep neural learning models to analyze thread dumps and messaging system logs for intelligent runtime diagnostics. The proposed framework integrates operational data preprocessing, sequence modeling, anomaly detection, and event correlation mechanisms to identify hidden runtime behaviors and operational anomalies efficiently. Deep learning architectures such as LSTM networks and transformer-based models enable accurate recognition of operational signatures associated with deadlocks, thread contention, mailbox congestion, latency spikes, and infrastructure instability.

Experimental evaluation demonstrated that the proposed framework significantly improves anomaly detection accuracy, root cause identification efficiency, and operational troubleshooting performance compared to traditional monitoring approaches. AI-driven operational intelligence reduced manual engineering effort, minimized false-positive alerts, and enhanced enterprise support engineering workflows. The framework also improved predictive maintenance capabilities and strengthened operational observability across distributed messaging infrastructures.

The study highlights the transformative potential of Artificial Intelligence and deep learning technologies in modern operational analytics and enterprise infrastructure management. Future enhancements involving explainable AI, federated learning, streaming analytics, and autonomous remediation systems can further advance intelligent operational diagnostics and support the development of fully adaptive self-healing enterprise environments.

REFERENCES

1. Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly detection and diagnosis from system logs through deep learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 1285–1298. https://doi.org/10.1145/3133956.3134015
2. Thota, M. R. (2022). Next generation observability: AI techniques for predictive performance and reliability in data intensive systems. Journal of Scientific and Engineering Research, 9(3), 360–374. https://doi.org/10.5281/zenodo.17839948
3. Seetala, S. R. (2016). Architectural evolution in enterprise data modeling: From dimensional

leadership to hybrid integration frameworks. International Journal of Technology, Management and Humanities, 2(1), 52–66. https://doi.org/10.21590/ijtmh.2.01.5

4. Vollem, S. (2017). An architectural and strategic analysis of enterprise-scale re-engineering approaches for modernizing legacy financial systems through Java-centric software paradigms and intelligent cloud automation frameworks. International Journal of Scientific Research in Science, Engineering and Technology, 3(3), 878–896. https://doi.org/10.32628/IJSRSET1773170
5. Studiawan, H., Sohel, F., & Payne, C. (2019). A survey on forensic investigation of operating system logs. Digital Investigation, 29, 1–20. https://doi.org/10.1016/j.diin.2019.02.005
6. Menda, J. R. (2021). Building resilient and compliance-driven observability architectures for modern BFSI enterprises using unified monitoring, telemetry correlation, and proactive incident intelligence. International Journal of Science, Engineering and Technology, 9(1). https://doi.org/10.5281/zenodo.18107872
7. Yamsani, N. (2017). Enterprise-scale data stewardship enablement using workflow-driven governance mechanisms in financial services. International Journal of Technology, Management and Humanities, 3(1). https://doi.org/10.21590/ijtmh.3.03.3
8. Parepalli, S. (2021). Hybrid control strategies for efficient scheduling and flow management in ETL pipelines. International Journal of Scientific Research & Engineering Trends, 7(3). https://doi.org/10.5281/zenodo.17896504
9. He, S., Zhu, J., He, P., & Lyu, M. R. (2016). Experience report: System log analysis for anomaly detection. IEEE International Symposium on Software Reliability Engineering, 207–218. https://doi.org/10.1109/ISSRE.2016.21
10. Thompson, D., Bennett, O., Walker, J., Collins, H., & Richard, A. (2021). AI-driven autonomic control with machine learning for self-healing distributed systems. Zenodo. https://doi.org/10.5281/zenodo.20064011
11. Ghanta, S. (2017). From broker-centric queues to distributed logs: Reliable messaging models for enterprise applications using Apache Kafka. Journal of Scientific and Engineering Research, 4(6), 253–260. https://doi.org/10.5281/zenodo.18084828
12. Reddy BasiReddy, S. (2016). Java-centric workflow orchestration for enhancing telecom service provisioning and CRM operations. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 1(3), 111–119. https://doi.org/10.32628/CSEIT11833644
13. Seetala, S. R. (2018). A comprehensive framework for cloud migration of enterprise data warehouses: Architectural transformation, performance optimization, and governance considerations. International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), 4(1), 1861–1878. https://doi.org/10.32628/IJSRSET1874102
14. Thota, M. R. (2021). Cognitive workload placement models: Integrating AI analytics for cost efficient and resilient cloud operations. European Journal of Advances in Engineering and Technology, 8(6), 172–184. https://doi.org/10.5281/zenodo.17839006
15. Lin, Q., Zhang, H., Lou, J.-G., Zhang, Y., & Chen, X. (2016). Log clustering based problem identification for online service systems. Proceedings of the 38th International Conference on Software Engineering Companion, 102–111. https://doi.org/10.1145/2889160.2889232
16. Vollem, S. (2022). Streaming-first enterprise decision systems: Architectural evolution from batch dataflows to stateful, exactly-once real-time processing. International Journal of Engineering & Extended Technologies Research (IJEETR), 4(1), 4326–4335. https://doi.org/10.15662/IJEETR.2022.0401005
17. Xu, W., Huang, L., Fox, A., Patterson, D., & Jordan, M. (2009). Detecting large-scale system problems by mining console logs. Proceedings of the ACM SIGOPS Symposium on Operating Systems Principles, 117–132. https://doi.org/10.1145/1629575.1629587
18. Nagender, Y. (2017). Constructing master data to be auditable by design: How lineage transparency and change discipline are engineered in enterprise-scale data estates.

International Journal of Science, Engineering and Technology, 5(5). https://doi.org/10.5281/zenodo.18184902

19. Menda, J. R. (2019). A comprehensive study on designing regulatory compliant multi-cloud resilience architectures for mission-critical Java-based enterprise systems. International Journal of Science, Engineering and Technology, 7(6). https://doi.org/10.5281/zenodo.18107819
20. Collins, M., Reynolds, S., Foster, A., Brooks, D., Bennett, O., & Srinivas, C. (2021). Reimagining enterprise master data management for trusted and intelligent business operations. International Journal of Science, Engineering and Technology, 9(5). https://doi.org/10.5281/zenodo.19704142
21. Parepalli, S. (2017). Evolving enterprise reconciliation: From deterministic validation to AI-supported high-integrity data assurance. Journal of Scientific and Engineering Research, 4(6), 242–252. https://doi.org/10.5281/zenodo.18084791
22. Ghanta S. Quality-Driven Microservice Refactoring of Legacy Java Systems: Patterns, Automation, and Migration Challenges. J Artif Intell Mach Learn & Data Sci 2018 1(1), 3197-3202. DOI: https://doi.org/10.51219/JAIMLD/Sriram-Ghanta/649
23. Fu, Q., Lou, J.-G., Wang, Y., & Li, J. (2009). Execution anomaly detection in distributed systems through unstructured log analysis. IEEE International Conference on Data Mining, 149–158. https://doi.org/10.1109/ICDM.2009.60
24. BasiReddy, S. R. (2018). Modernizing CRM data pipelines through parallel processing and cloud-native orchestration. International Journal of Scientific Research & Engineering Trends, 4(2). Zenodo. https://doi.org/10.5281/zenodo.18014580
25. Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., Chen, Y., & Zhang, X. (2019). LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 4739–4745. https://doi.org/10.24963/ijcai.2019/658
26. Vollem, S. (2021). Architecting zero trust security for distributed hybrid and multi-cloud enterprise systems. International Numeric Journal of Machine Learning and Robots, 5(5). https://injmr.com/index.php/fewfewf/article/view/236
27. Parepalli, S. (2016). Data hygiene and batch optimization in enterprise CRM: A framework for scalable, high-quality customer data integration. Journal of Scientific and Engineering Research, 3(5), 285–292. https://doi.org/10.5281/zenodo.18084598
28. Oliner, A., & Stearley, J. (2007). What supercomputers say: A study of five system logs. International Conference on Dependable Systems and Networks, 575–584. https://doi.org/10.1109/DSN.2007.103
29. Nagender, Y. (2019). Engineering trustworthy enterprise data through structured validation and cleansing controls: Insights from Elavon data quality operations. International Journal of Science, Engineering and Technology, 7(1). https://doi.org/10.5281/zenodo.18194337
30. Collins, A., Turner, R., Walker, J., Bennett, O., Harris, M., & Srinivas, C. (2019). Designing robust CI/CD pipelines for quality assurance in regulated financial systems. International Journal of Scientific Research & Engineering Trends, 5(6). https://doi.org/10.5281/zenodo.19763655
31. Makanju, A., Zincir-Heywood, A. N., & Milios, E. (2009). Clustering event logs using iterative partitioning. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1255–1264. https://doi.org/10.1145/1557019.1557154
32. Thota, M. R. (2020). Architecting secure and compliant hybrid cloud database systems: Frameworks, cryptography, and big data platforms. International Journal of Scientific Research & Engineering Trends, 6(5). Zenodo. https://doi.org/10.5281/zenodo.18479002
33. BasiReddy, S. R. (2019). Resource-oriented API architectures for cross-domain CRM and telecom platforms. European Journal of Advances in Engineering and Technology, 6(7), 89–95. https://doi.org/10.5281/zenodo.18083237

34. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539
35. Seetala, S. R. (2022). Adaptive machine learning frameworks for data quality monitoring: From anomaly detection to continuous pipeline validation. International Journal of Research and Applied Innovations, 5(1), 9467–9477. https://doi.org/10.15662/IJRAI.2022.0501007
36. Ghanta, S. (2018). From monolith to cloud-native: Building Java microservices with Spring Boot, Docker, and Kubernetes. Journal of Scientific and Engineering Research, 5(10), 373–380. https://doi.org/10.5281/zenodo.18085020
37. Menda, J. R. (2017). Distributed in-memory caching as the backbone of real-time banking: Architecture, patterns, and performance. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2(5), 1120–1131. https://doi.org/10.32628/CSEIT1726327