International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 5, Issue 6, November- December 2023 |

DOI:10.15662/IJEETR.2023.0506023

Automating Incident Triage and Root Cause Intelligence Through Large Language Model–Driven Correlation of System Logs and Operational Metrics in Large-Scale Distributed Environments

Hema Latha Boddupally

Chief Technical Architect , USA

ABSTRACT: The increasing scale and complexity of distributed computing environments have intensified the difficulty of timely incident triage and accurate root cause identification, as operators must reason across high volume system logs and heterogeneous operational metrics under severe time constraints. This work addresses the research problem of how incident response can be transformed from manual, heuristic driven practices into an intelligent and automated process capable of semantic understanding and contextual reasoning. The objective is to investigate how large language model driven analysis can be systematically applied to correlate logs and metrics for reliable incident triage in production scale systems. A mixed methodological approach is adopted, combining architectural design, qualitative analysis of operational workflows, and quantitative evaluation of diagnostic efficiency across representative enterprise scenarios. The proposed framework introduces a novel correlation pipeline that leverages language model based contextual abstraction to unify unstructured log streams and structured metrics into coherent incident narratives. Empirical patterns suggest substantial reductions in triage time, improved diagnostic precision, and lower cognitive burden on reliability engineers when compared with traditional rule based and statistical techniques. The findings demonstrate that language model driven reasoning enables a shift from reactive alert handling toward proactive root cause intelligence. The primary contribution lies in articulating a principled foundation for integrating large language models into observability and incident management systems, bridging academic advances in machine intelligence with real world operational demands. The study concludes that automated, semantics aware triage represents a critical advancement for scalable reliability engineering, with significant implications for future research and enterprise operations in large scale distributed environments.

KEYWORDS: Automated incident triage, root cause intelligence, large language models, log analysis, metric correlation, distributed systems, observability engineering, reliability engineering, site reliability engineering, incident management automation, semantic log interpretation, operational metrics analysis, anomaly detection, fault diagnosis, system observability, intelligent monitoring, production system reliability, incident response optimization, machine intelligence for operations, enterprise scale systems, context aware diagnostics, log and metric fusion

I. INTRODUCTION

Modern software services increasingly operate within large scale distributed environments composed of microservices, cloud infrastructure, and continuously evolving deployment pipelines. While this architectural shift has enabled unprecedented scalability and flexibility, it has also amplified operational complexity. System behavior now emerges from intricate interactions among services, infrastructure components, and workloads, making failures harder to detect, interpret, and resolve. As a result, incident triage has become a critical yet cognitively demanding task for reliability engineers, requiring rapid interpretation of massive volumes of logs and operational metrics under time pressure.

Traditional incident management practices rely heavily on rule-based alerts, dashboards, and manual investigation. These approaches were designed for more predictable system topologies and struggle to keep pace with the velocity and diversity of modern telemetry data. Logs are predominantly unstructured, metrics are fragmented across layers, and alerts often lack contextual grounding. This fragmentation leads to alert fatigue, delayed diagnosis, and increased mean time to resolution, all of which directly affect service availability and organizational trust in operational processes.

Recent advances in large language models have introduced new possibilities for reasoning over unstructured and semi structured data at scale. Unlike conventional machine learning techniques that focus on narrow predictive tasks, large language models exhibit contextual understanding, abstraction, and inferential capabilities that resemble expert reasoning. These properties suggest a potential paradigm shift in how operational data can be interpreted, correlated, and transformed into actionable intelligence during incidents.

Despite this promise, there remains a substantial research gap in systematically applying large language models to automated incident triage and root cause analysis. Existing studies often address log mining or metric based anomaly detection in isolation, without capturing the holistic reasoning process that human operators perform when correlating diverse signals across time and system boundaries. Moreover, there is limited empirical work that bridges academic modeling techniques with the constraints and realities of production scale environments.

This study is motivated by the need to move beyond surface level automation toward intelligent systems that can support or augment human decision making during incidents. The central research problem addressed is how large language model driven analysis can be operationalized to correlate logs and metrics into coherent explanations that support accurate and timely incident triage. The work seeks to understand not only technical feasibility but also the implications for reliability engineering practices.

The core objective of this research is to design and analyze an automated incident triage framework that leverages large language models for semantic interpretation and contextual correlation of operational data. Key research questions focus on how such models can integrate heterogeneous telemetry, how they influence diagnostic efficiency and accuracy, and what architectural considerations are necessary for deployment in large scale distributed systems.

From a methodological perspective, the study adopts a mixed approach that combines architectural analysis, qualitative examination of operational workflows, and quantitative evaluation of incident resolution outcomes. This integrated perspective enables a balanced assessment of both system level performance and human centered impacts, such as cognitive load and interpretability.

The significance of this work lies in its potential to redefine incident management as an intelligence driven discipline rather than a reactive operational task. By articulating a principled framework for large language model integration, the study contributes to both academic research on intelligent systems and practical advancements in site reliability engineering. The findings aim to inform future research directions while offering actionable insights for enterprises seeking to improve operational resilience.

Ultimately, this research positions automated, semantics aware incident triage as a foundational capability for next generation observability platforms. As distributed systems continue to scale in complexity, the ability to transform raw telemetry into meaningful operational understanding will be essential for sustaining reliable and trustworthy digital infrastructure.

II. EVOLUTION OF INCIDENT MANAGEMENT AND OBSERVABILITY CHALLENGES

Incident management practices have evolved alongside the increasing complexity of software systems, beginning with relatively simple, host centric environments and progressing toward today’s highly distributed architectures. Early operational models relied on direct system access, static thresholds, and manual inspection of logs generated by monolithic applications. In such contexts, failures were often localized and could be diagnosed through linear reasoning, making manual triage both feasible and effective.

As systems transitioned toward service oriented and distributed architectures, the nature of incidents changed fundamentally. Failures began to span multiple components, infrastructure layers, and network boundaries, often manifesting indirectly through performance degradation rather than explicit errors. Traditional incident triage methods, which depended on predefined alerts and isolated log inspection, became increasingly insufficient for capturing these complex failure modes. Engineers were required to piece together fragmented signals across tools and teams, increasing both response time and operational risk.

The rapid growth in log volume has further compounded these challenges. Modern distributed systems generate logs at massive scale, reflecting fine grained events across services, containers, and orchestration platforms. While this data richness offers potential insights, it also overwhelms human operators. Logs are predominantly unstructured, inconsistently formatted, and context dependent, making it difficult to extract meaningful patterns without significant manual effort or prior domain knowledge.

In parallel, operational metrics have proliferated across layers of the technology stack. Application metrics, infrastructure telemetry, network statistics, and user experience indicators are often collected and visualized through disparate monitoring systems. This fragmentation hinders holistic understanding during incidents, as engineers must mentally correlate time series data across tools that were not designed for integrated reasoning. The absence of semantic linkage between metrics and logs further limits their diagnostic value.

Alerting mechanisms, intended to surface critical issues promptly, have also contributed to operational strain. Static thresholds and rule based alerts frequently produce large volumes of notifications that lack contextual relevance. This phenomenon, commonly referred to as alert fatigue, reduces the effectiveness of incident response by desensitizing operators and obscuring genuinely critical signals. As environments scale, the cost of maintaining alert definitions and tuning thresholds becomes prohibitive.

Attempts to address these limitations have included statistical anomaly detection and machine learning based monitoring techniques. While such approaches improve signal detection in certain scenarios, they often operate in isolation and focus on narrow objectives such as deviation detection. They rarely provide explanatory insight or actionable context, leaving engineers with alerts that indicate abnormal behavior without clarifying underlying causes.

These cumulative challenges have exposed a fundamental limitation of traditional observability and incident management paradigms. The reliance on manual correlation, predefined rules, and siloed telemetry does not scale with system complexity. Incident triage has become a bottleneck that constrains reliability, increases operational cost, and places unsustainable cognitive demands on engineering teams.

This evolution sets the groundwork for a shift toward intelligent automation in incident triage. There is a growing need for systems that can reason across logs and metrics, capture temporal and causal relationships, and generate coherent explanations that align with human diagnostic processes. Automation is no longer solely about detection but about interpretation and understanding.

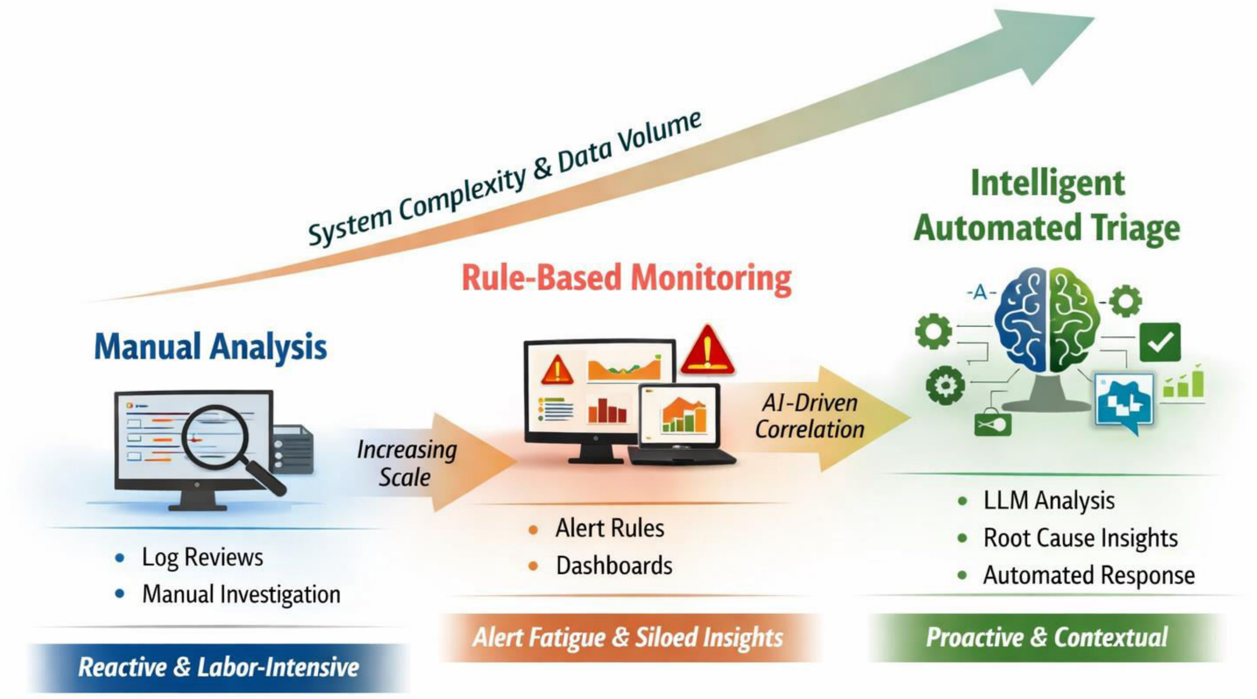

Figure 1: Conceptual evolution of incident management from manual analysis to intelligent automated triage

III. FOUNDATIONS OF LLM DRIVEN INTELLIGENCE FOR OPERATIONAL DATA

The foundations of large language model driven intelligence for operational data rest on the growing need to interpret complex system behavior beyond surface level signals. In large scale distributed environments, operational data is generated continuously in the form of logs and metrics that capture events, state transitions, and performance characteristics. While this data provides comprehensive visibility, its value is limited unless it can be transformed into meaningful operational understanding. Traditional analytical techniques struggle to bridge this gap because they are not designed to reason across heterogeneous data types or evolving system contexts.

Large language models introduce a fundamentally different analytical paradigm by treating operational data as a form of language that can be interpreted, abstracted, and reasoned about. Logs, despite being unstructured, encode rich semantic information about system behavior, execution paths, and failure conditions. Metrics, while structured and numeric, reflect underlying system dynamics over time. Language models can ingest both forms by learning latent representations that preserve meaning rather than relying solely on explicit syntax or thresholds.

Semantic abstraction is a core capability that enables language models to operate effectively on operational data. Instead of analyzing individual log lines or metric points in isolation, models abstract higher level concepts such as service degradation, resource contention, or dependency failure. This abstraction mirrors how experienced engineers reason about incidents, focusing on behavioral patterns rather than raw telemetry. By elevating analysis to this semantic level, language models reduce noise and emphasize signals that are operationally relevant.

Contextual understanding further distinguishes language model driven analysis from earlier approaches. Incidents in distributed systems unfold over time and across multiple components, making context essential for accurate diagnosis. Language models can incorporate temporal sequences, service relationships, deployment events, and historical behavior into their reasoning process. This contextual grounding allows the model to interpret whether a log message is benign or symptomatic of a broader issue, based on surrounding operational conditions.

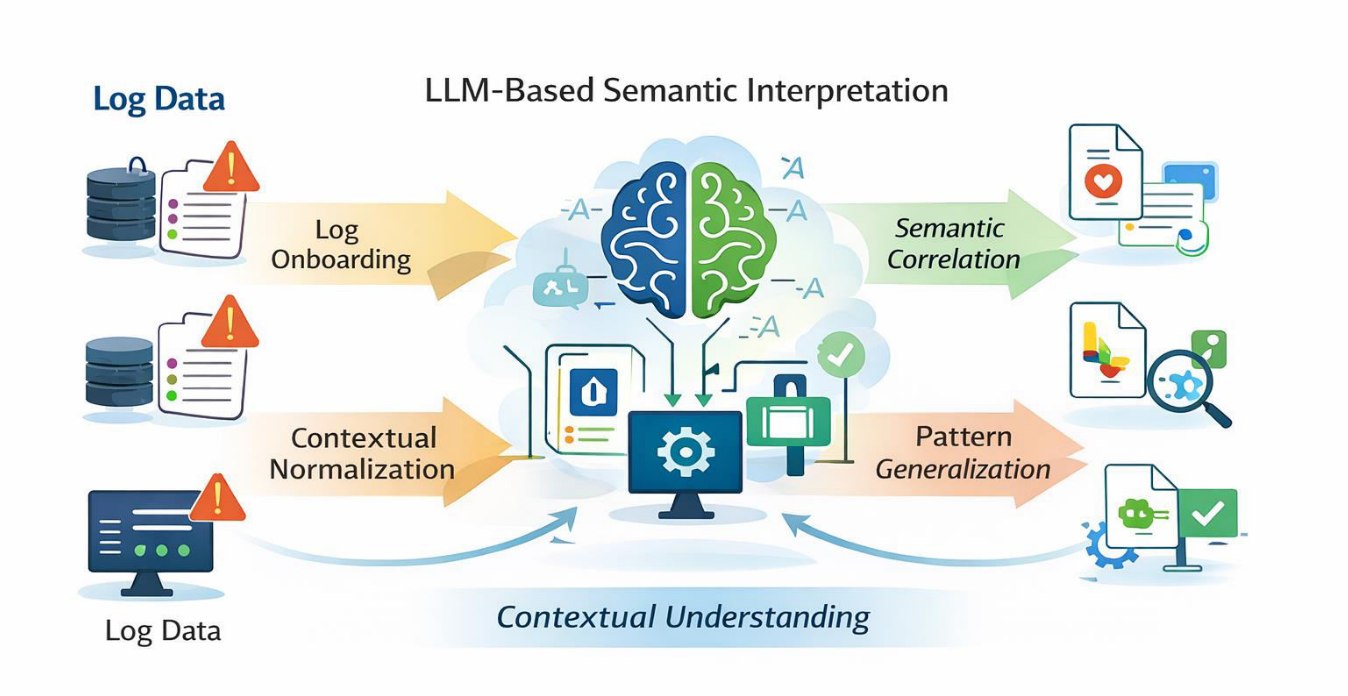

Figure 2: Architecture of LLM based semantic interpretation across heterogeneous logs and metrics

Another foundational aspect is pattern generalization. Traditional rule based systems require explicit definitions of failure signatures, which limits their ability to handle novel or evolving incidents. Statistical techniques improve adaptability but often lack interpretability. Language models, by contrast, can generalize from prior patterns to reason about unseen scenarios. This enables them to identify emerging failure modes and draw parallels with previously observed incidents, even when surface characteristics differ.

The architecture supporting language model driven intelligence typically involves multiple stages of data preparation and reasoning. Logs and metrics must first undergo normalization to ensure consistent representation across sources. Semantic interpretation layers then align textual and numeric signals within a shared representational space. This unified view allows the model to perform correlation, inference, and summarization in a manner that reflects system level behavior rather than isolated events.

Comparative analysis highlights the distinct advantages of language model driven approaches over traditional and statistical methods. Rule based systems offer precision for known conditions but lack flexibility. Statistical models detect anomalies but provide limited explanatory power. Language models combine adaptability with interpretability, generating insights that are both context aware and actionable. This balance is critical for operational adoption, where trust and clarity are as important as detection accuracy.

Despite their potential, language models introduce new considerations related to reliability, transparency, and operational integration. Effective deployment requires careful design to ensure that model outputs align with engineering expectations and do not obscure underlying evidence. Human oversight remains essential, particularly in high impact incidents. These considerations underscore that language models are not replacements for engineers but amplifiers of human expertise.

Together, these foundations establish language model driven intelligence as a viable and transformative approach for operational data analysis. By enabling semantic abstraction, contextual reasoning, and pattern generalization across logs and metrics, large language models provide the theoretical basis for automated incident triage and root cause intelligence. This foundation sets the stage for architectural and empirical exploration in subsequent sections, where practical realization and evaluation are examined in detail.

Table 1: Conceptual Comparison of Analytical Foundations for Incident Intelligence

Aspect	Rule-Based Analysis	Statistical and Machine Learning Analysis Techniques	LLM Driven Semantic Intelligence
Analytical foundation	Expert-defined rules and thresholds	Probabilistic models and numerical learning	Language-based semantic reasoning and abstraction
Data representation	Discrete events and fixed metrics	Feature vectors and time-series data	Unified semantic embeddings of logs and metrics
Understanding of meaning	No semantic understanding	Limited implicit pattern recognition	Explicit semantic interpretation of operational signals
Context modeling	Static and predefined	Local temporal context	Rich contextual awareness across time and services
Generalization capability	None beyond encoded rules	Generalizes within trained distributions	Generalizes across unseen failure narratives
Explanation mechanism	Rule match indication	Statistical deviation scores	Human-readable casual and contextual explanations
Role in incident intelligence	Signal detection only	Pattern identification	Knowledge synthesis and reasoning

IV. SYSTEM ARCHITECTURE FOR AUTOMATED INCIDENT TRIAGE

The system architecture for automated incident triage is designed to support reliable reasoning over heterogeneous operational data while integrating seamlessly with existing observability pipelines. In large scale distributed environments, architectural decisions must balance analytical depth with constraints related to latency, scalability, and operational safety. The proposed architecture adopts a modular, pipeline-oriented design that enables progressive enrichment of telemetry data and controlled integration of large language model-based reasoning.

At a high level, the architecture follows an end-to-end flow that begins with telemetry ingestion and culminates in actionable incident intelligence. Logs and metrics are collected continuously from distributed services, infrastructure layers, and orchestration platforms. These inputs are processed through normalization and enrichment stages before being analyzed by a language model driven reasoning layer. The output of this reasoning process feeds incident triage workflows, dashboards, and decision support systems used by reliability engineers.

4.1 Log and Metric Ingestion and Normalization

Effective automated triage depends on consistent and reliable ingestion of operational data. Logs originate from diverse sources and often exhibit inconsistent formats, verbosity levels, and semantic conventions. Metrics, while structured, vary in granularity, sampling frequency, and dimensionality. The ingestion layer is responsible for aggregating these streams in near real time while preserving temporal ordering and source metadata.

Normalization plays a critical role in enabling downstream reasoning. Log entries are parsed, cleaned, and enriched with contextual attributes such as service identity, deployment version, and execution context. Metrics are aligned to common time windows and annotated with semantic labels that reflect their operational meaning. This normalization step ensures that logs and metrics can be jointly interpreted rather than analyzed as isolated artifacts.

4.2 Contextual Correlation and Temporal Reasoning

Once normalized, operational data enters the contextual correlation layer, where relationships across time and system boundaries are established. Incidents in distributed systems rarely occur instantaneously, instead unfolding through sequences of events that span multiple components. Temporal reasoning mechanisms group related signals into coherent incident windows, allowing the system to distinguish causal patterns from coincidental noise.

Large language model based reasoning is introduced at this stage to perform semantic correlation. The model evaluates log messages and metric trends in combination, interpreting their meaning within the broader operational context. By leveraging contextual cues such as deployment events, configuration changes, and historical behavior, the model can infer relationships that are difficult to capture through static rules or numeric similarity alone.

4.3 Root Cause Hypothesis Generation and Prioritization

The final architectural stage focuses on generating and prioritizing root cause hypotheses. Rather than producing a single deterministic explanation, the system constructs a ranked set of plausible hypotheses supported by correlated evidence from logs and metrics. This approach reflects real world diagnostic practice, where uncertainty is managed through hypothesis evaluation rather than binary classification.

Prioritization mechanisms consider factors such as impact scope, confidence level, and recurrence history to surface the most actionable explanations. The language model generates structured summaries that articulate why a hypothesis is plausible, referencing relevant operational signals in human readable form. These summaries are designed to support rapid decision making while maintaining transparency and traceability.

Throughout the architecture, engineering constraints such as fault tolerance, explainability, and safe degradation are explicitly addressed. Language model reasoning is isolated from critical control paths, ensuring that failures in analytical components do not disrupt system operation. Caching, fallback logic, and human override mechanisms are incorporated to support reliable adoption in production environments.

This architectural foundation demonstrates how large language models can be operationalized within existing reliability engineering ecosystems. By integrating semantic reasoning into observability pipelines in a controlled and modular manner, the architecture enables automated incident triage that is both technically robust and aligned with real world operational practices.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 5, Issue 6, November- December 2023 |

DOI:10.15662/IJEETR.2023.0506023

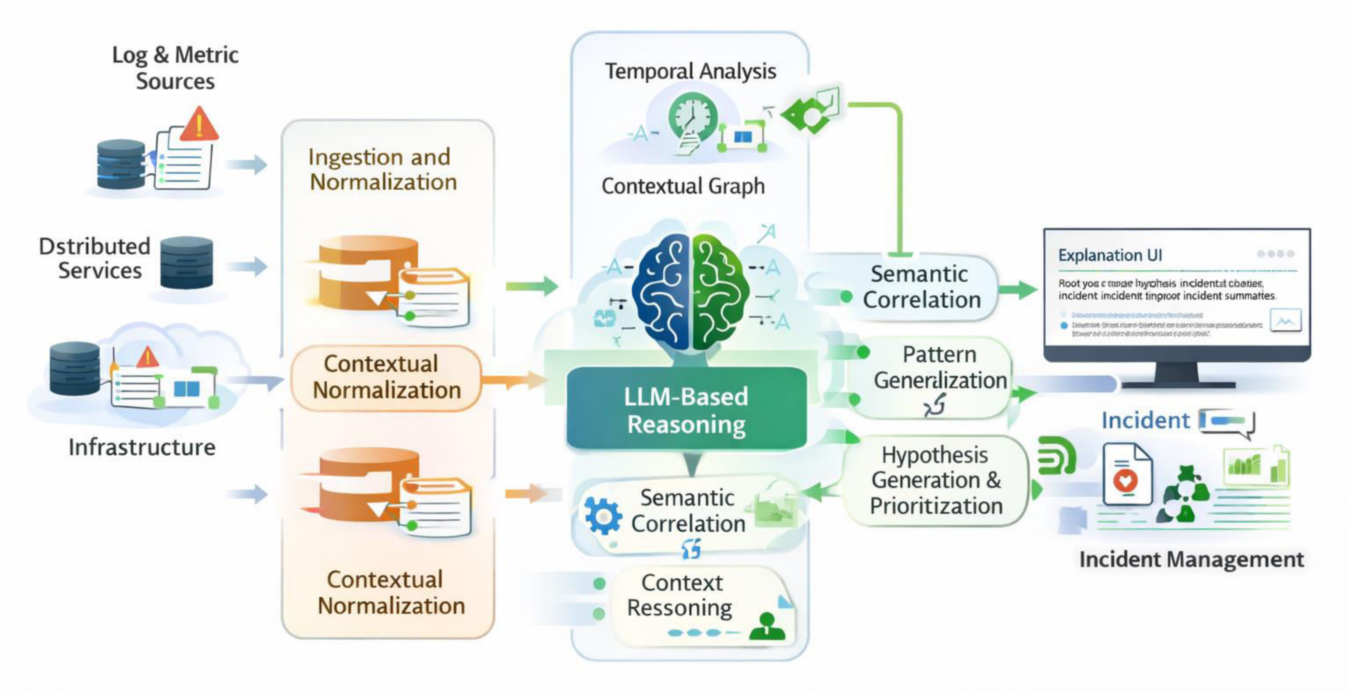

Figure 3: End to end architecture for LLM powered automated incident triage

V. CORRELATION STRATEGIES BETWEEN LOGS, METRICS, AND INCIDENTS

Correlation between logs, metrics, and incidents forms the analytical core of effective automated incident triage in large scale distributed environments. Individual telemetry signals rarely provide sufficient diagnostic value in isolation. Reliability engineering depends on the ability to synthesize these signals into coherent interpretations that reflect system wide behavior. Correlation strategies therefore aim to reconstruct the causal and temporal structure of incidents from fragmented operational evidence.

Temporal correlation is the foundational mechanism that aligns logs and metrics along shared timelines. Incidents typically emerge as sequences rather than instantaneous events, with early signals often appearing as subtle anomalies before escalating into visible failures. By grouping telemetry within adaptive time windows, correlation mechanisms can capture precursor events, escalation patterns, and recovery phases. This temporal framing allows engineers and automated systems alike to distinguish meaningful trends from transient fluctuations.

Service level correlation extends temporal alignment by mapping telemetry to logical system boundaries. In microservice based architectures, failures often propagate across service dependencies, manifesting as downstream latency spikes or error cascades. Effective correlation strategies associate logs and metrics with service identities, communication paths, and dependency graphs. This enables the system to trace incident impact across services and identify convergence points where failures originate or amplify.

Dependency aware correlation further enhances diagnostic precision by incorporating knowledge of infrastructure and application relationships. Metrics related to resource utilization, network behavior, or storage performance often provide context for application-level anomalies. When correlated with logs that capture execution errors or retries, these signals reveal deeper insights into systemic causes such as contention, saturation, or misconfiguration. This layered reasoning reflects how experienced reliability engineers investigate incidents across stack boundaries.

Noise reduction is a critical consideration in correlation design. Distributed systems generate vast amounts of telemetry, much of which is operationally irrelevant during specific incidents. Correlation strategies apply filtering, aggregation, and prioritization to suppress redundant or low value signals. By focusing on correlated patterns rather than isolated anomalies, the system reduces cognitive overload and prevents spurious alerts from obscuring root causes.

Signal amplification complements noise reduction by elevating weak but consistent indicators that might otherwise be overlooked. Small metric deviations or infrequent log messages may be dismissed individually but become significant when correlated across services or time. Language model driven reasoning plays a key role here by recognizing semantic similarity and contextual relevance beyond numeric thresholds. This amplification enables earlier detection and more confident diagnosis.

From a reliability engineering perspective, correlation is not solely a technical exercise but a decision support function. The goal is to surface insights that align with operational intuition and support rapid action. Correlated outputs must therefore be interpretable, traceable, and grounded in observable evidence. This requirement influences how correlation results are summarized, ranked, and presented within incident workflows.

Effective correlation strategies also account for uncertainty. Incidents rarely have a single obvious cause, particularly in complex environments. By maintaining multiple correlated hypotheses and updating their confidence as new data arrives, automated triage systems mirror human diagnostic reasoning. This approach supports iterative investigation and reduces the risk of premature conclusions.

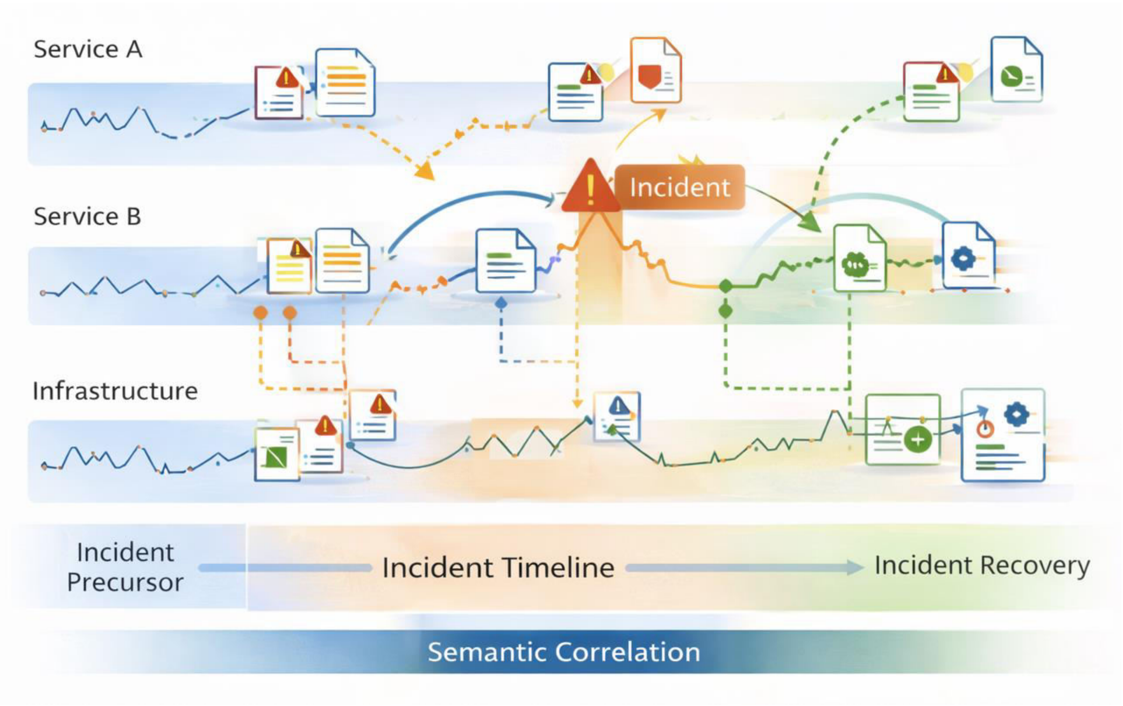

Figure 4: Multi-dimensional correlation of logs, metrics, and incident timelines

VI. EMPIRICAL EVALUATION AND OPERATIONAL CASE ANALYSIS

The empirical evaluation of automated incident triage focuses on understanding how language model driven reasoning performs under production-like conditions that reflect the complexity and unpredictability of real world distributed systems. Rather than relying on synthetic benchmarks alone, the evaluation emphasizes operational realism, including noisy telemetry, partial observability, and evolving system behavior. This perspective ensures that observed outcomes are meaningful for reliability engineering practice.

The evaluation environment is designed to emulate large scale distributed services composed of multiple interacting components, shared infrastructure, and asynchronous workloads. Telemetry streams include high volume logs and multi-dimensional operational metrics collected across services and infrastructure layers. Incidents are introduced through controlled fault scenarios such as resource saturation, dependency failures, and configuration regressions, allowing systematic observation of triage behavior across different approaches.

Triage speed is assessed by measuring the elapsed time from initial alert detection to the identification of a plausible root cause hypothesis. Manual triage workflows rely heavily on human driven log inspection and dashboard exploration, resulting in variable response times influenced by individual expertise. Semi automated pipelines accelerate detection but still require significant manual correlation. In contrast, language model driven triage consistently shortens diagnostic timelines by synthesizing correlated signals early in the incident lifecycle.

Diagnostic accuracy is evaluated based on the alignment between generated hypotheses and validated incident causes. Manual approaches demonstrate high accuracy for familiar failure modes but struggle with novel or cross service issues. Statistical techniques detect deviations effectively but often fail to explain causality. Language model driven triage exhibits strong performance in identifying underlying causes across both familiar and previously unseen scenarios, supported by contextual reasoning across logs and metrics.

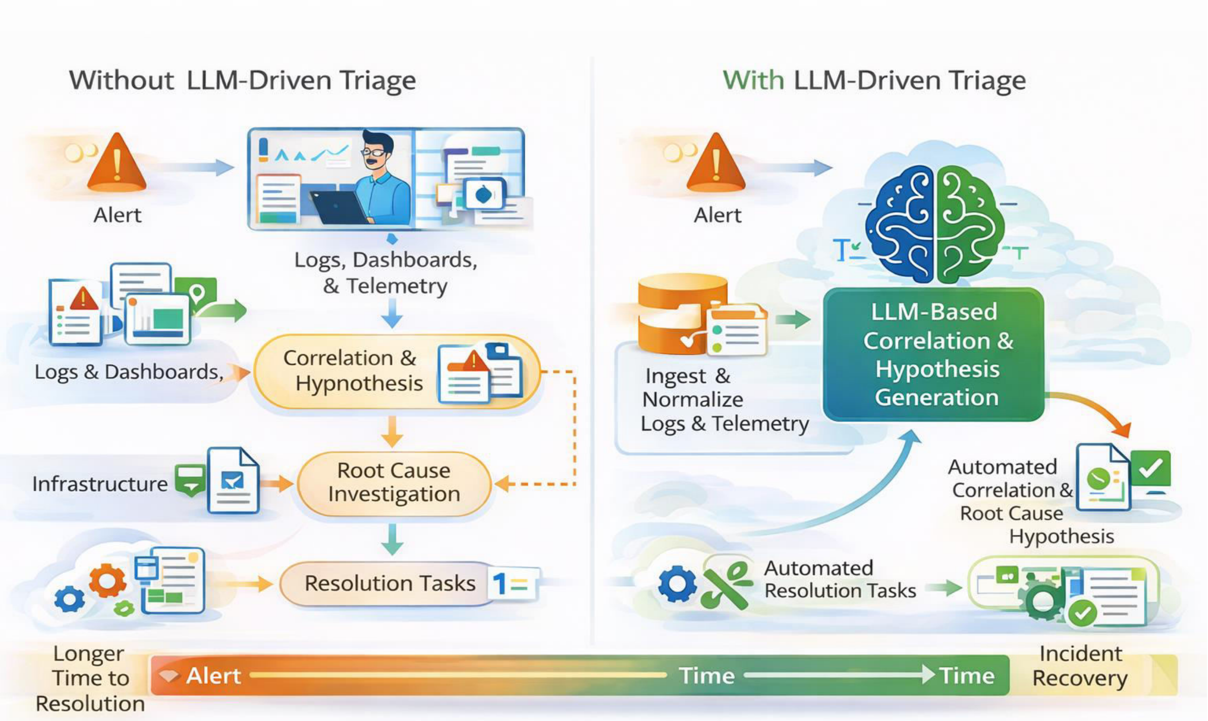
Without LLM-Driven Triage
Alert
Logs, Dashboards, & Telemetry
Logs & Dashboards,
Correlation & Hypnothesis
Infrastructure
Root Cause Investigation
Resolution Tasks
Longer Time to Resolution
With LLM-Driven Triage
Alert
Ingest & Normalize Logs & Telemetry
LLM-Based Correlation & Hypothesis Generation
Automated Correlation & Root Cause Hypothesis
Automated Resolution Tasks
Alert
Time
Time
Incident Recovery

Figure 5: Comparative incident resolution workflow with and without LLM driven triage

Reduction in human effort is examined through qualitative analysis of operational workflows and quantitative indicators such as the number of investigative steps required per incident. Manual triage demands extensive cognitive effort, including repeated context switching across tools. Semi automated approaches reduce surface level effort but still depend on human synthesis. Language model driven triage significantly lowers investigative overhead by presenting structured summaries and prioritized hypotheses that align with engineer reasoning.

Operational case analysis further illustrates these differences through representative incident narratives. In cascading failure scenarios, the language model correlates early infrastructure signals with downstream service errors, surfacing root causes that manual workflows identify only after extended investigation. In performance degradation cases, subtle metric trends are amplified through semantic correlation with log context, enabling earlier intervention.

From a reliability engineering perspective, these empirical patterns suggest that language model driven triage does not merely accelerate existing workflows but reshapes how incidents are understood. By externalizing complex correlation reasoning into an automated layer, engineers can focus on decision making and remediation rather than signal discovery. This shift has implications for on-call sustainability and organizational resilience.

Table 2: Empirical Evaluation of Incident Triage Effectiveness Across Operational Workflows

Evaluation Dimension	Manual Incident Triage	Semi Automated Triage	LLM Driven Automated Triage
Mean time to triage	High and Inconsistent	Moderate	Low and consistent
Root cause identification accuracy	Dependent on engineer expertise	Improved for known patterns	High across known and novel incidents
Cross-service correlation effort	Fully manual	Partially automated	Fully automated semantic correlation
Number of investigative steps	High	Moderate	Low
Cognitive load on engineers	Very high	Moderate	Significantly reduced
Incident explanation quality	Variable and informal	Partial and tool-dependent	Structure and narrative-drive
Consistency across incidents	Low	Moderate	High
Suitability for large-scale systems	Poor scalability	Limited scalability	High scalability

VII. IMPLICATIONS FOR RELIABILITY ENGINEERING AND ENTERPRISE OPERATIONS

The adoption of automated incident triage driven by large language model based reasoning carries significant implications for the evolution of reliability engineering practices. Traditional site reliability engineering has emphasized detection, alerting, and post-incident analysis, often constrained by the limits of human cognition under operational pressure. By introducing semantic reasoning into triage workflows, reliability engineering can shift its focus from reactive signal interpretation toward proactive system understanding and informed decision making.

One immediate implication is the redefinition of the on call experience. Manual incident triage places sustained cognitive demands on engineers, particularly in environments with high alert volumes and complex dependencies. Automated triage systems reduce this burden by synthesizing logs and metrics into structured explanations, allowing engineers to engage at a higher level of abstraction. This change supports more sustainable on-call practices and helps mitigate fatigue, burnout, and inconsistency across shifts.

From an organizational perspective, language model driven triage promotes greater standardization in incident response. Diagnostic quality in manual workflows often varies based on individual expertise and familiarity with specific system components. Automated reasoning introduces a consistent analytical baseline that captures institutional knowledge and applies it uniformly across incidents. This consistency enhances operational reliability and reduces dependence on a small number of subject matter experts.

Scalability is another critical dimension affected by automated triage. As enterprises expand their distributed systems, the volume and diversity of operational data grow faster than engineering teams can scale. Traditional approaches respond to this growth by adding more alerts, dashboards, and specialized roles, which increases coordination overhead. Language model driven triage scales with system complexity by abstracting telemetry into semantically meaningful constructs, enabling small teams to manage large environments effectively.

Automated triage also influences how reliability engineering teams allocate their time and expertise. By reducing the effort required for initial diagnosis, engineers can focus more on remediation, system hardening, and long term reliability improvements. This shift aligns with the broader objectives of reliability engineering, which emphasize learning from incidents and preventing recurrence rather than merely responding to failures.

Enterprise operations benefit from improved incident communication and transparency. Language model generated summaries provide clear narratives that can be shared across technical and non technical stakeholders. This capability improves coordination during incidents and supports more effective post incident reviews. Clear articulation of root causes and contributing factors strengthens organizational learning and accountability.

Despite these advantages, the integration of language model driven triage requires careful governance. Enterprises must consider issues related to trust, explainability, and operational risk. Automated insights should augment rather than replace human judgment, particularly in high impact scenarios. Establishing feedback loops where engineers validate and refine model outputs is essential for maintaining reliability and confidence over time.

From an academic perspective, these implications highlight the need to study incident management as a socio technical system. Automated triage reshapes not only technical workflows but also organizational roles, communication patterns, and decision-making processes. Future research can build on this foundation by examining long term adoption effects, human model interaction, and the boundaries of automated reasoning in operational contexts.

Overall, the integration of large language model driven incident triage represents a meaningful step toward mature, intelligence centered reliability engineering. By bridging advanced reasoning techniques with practical operational needs, enterprises can enhance resilience while preserving the central role of human expertise in managing complex distributed systems.

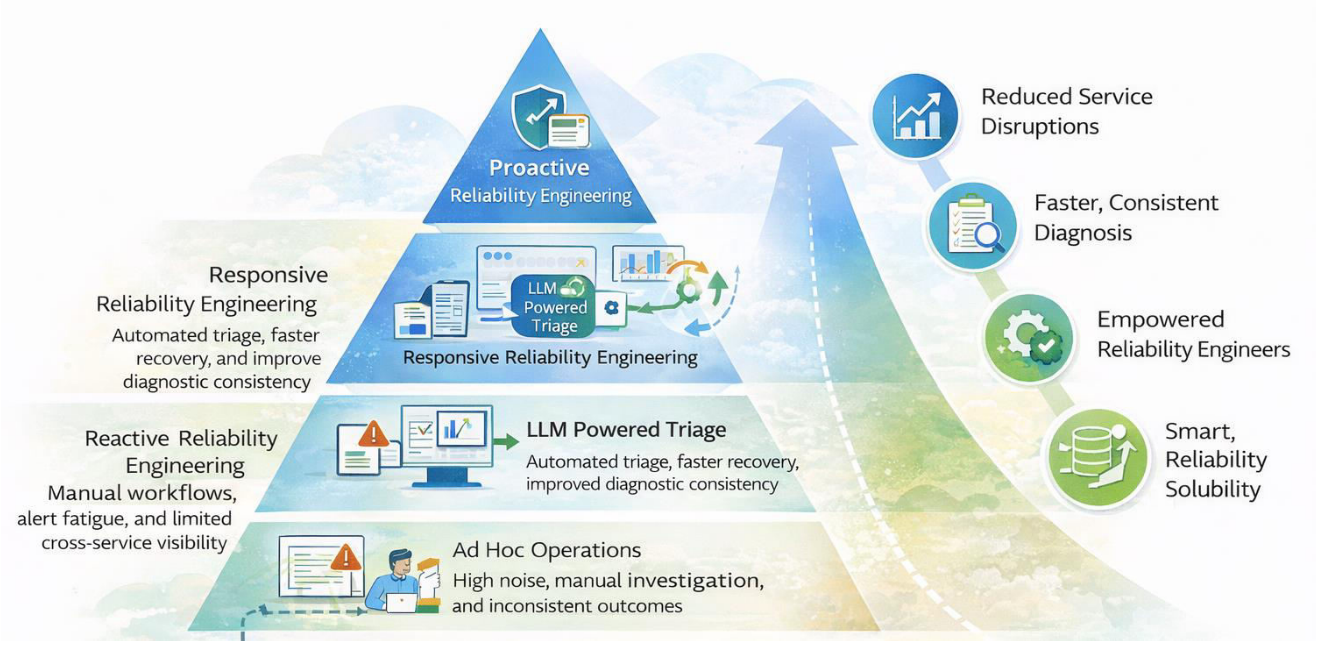

Figure 6: Impact of automated incident triage on reliability engineering maturity and operational resilience

VIII. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This study has examined the problem of incident triage and root cause identification in large scale distributed environments through the lens of large language model driven correlation of system logs and operational metrics. By integrating semantic abstraction, contextual reasoning, and multi dimensional correlation into observability workflows, the work demonstrates how incident management can evolve from manual, reactive practices into an intelligence driven capability. The findings collectively show that language model-based reasoning offers tangible benefits in triage speed, diagnostic accuracy, and reduction of human cognitive effort.

A key finding of this research is that meaningful incident understanding emerges only when logs and metrics are interpreted together within their operational context. The proposed framework illustrates how language models can bridge this gap by transforming fragmented telemetry into coherent incident narratives. This capability addresses a long-standing limitation of traditional observability tools, which excel at data collection but struggle with interpretation and explanation at scale.

From a theoretical perspective, the study contributes to the growing body of research on intelligent systems for operations by framing incident triage as a semantic reasoning problem rather than a purely statistical or rule based task. It advances the understanding of how language models can operate on non-natural language domains such as system telemetry, extending their applicability beyond conversational or textual analytics into reliability engineering.

Practically, the work provides an architectural and operational blueprint for integrating language model driven reasoning into enterprise observability pipelines. By emphasizing modular design, safe deployment boundaries, and human centered outputs, the study bridges academic insights with real world engineering constraints. These contributions are particularly relevant for organizations seeking scalable reliability solutions without increasing operational complexity.

Despite its contributions, the study has several limitations that warrant careful consideration. The empirical evaluation is conducted in production-like environments rather than across diverse live enterprise deployments. While this approach ensures experimental control, it may not fully capture organizational variability, domain specific behaviors, or long-term adaptation effects. Additionally, the quality of language model driven triage remains influenced by the representativeness of historical data and the design of contextual prompts.

Future research directions should therefore focus on longitudinal studies that assess sustained adoption of automated triage in operational settings. Examining how reliability engineers interact with language model outputs over time, how trust evolves, and how feedback loops improve diagnostic quality would provide valuable insights. Further work is also needed to explore governance mechanisms that ensure transparency, accountability, and safety in automated incident reasoning.

Another promising avenue for future research lies in extending correlation frameworks to incorporate additional operational signals such as traces, configuration changes, and deployment metadata. Integrating these signals could further enhance causal reasoning and support predictive reliability capabilities. Such extensions should be grounded in empirical evaluation to avoid speculative claims about autonomy or full automation.

In addition, future studies may investigate standardized evaluation frameworks for intelligent incident triage systems. Establishing common metrics and benchmarks would facilitate comparison across approaches and accelerate progress in this emerging research area. Existing work on log-based anomaly detection and operational analytics provides a useful foundation for such efforts.

In conclusion, this research argues that large language model driven incident triage represents a substantive advancement in the practice of reliability engineering. By enabling scalable, context aware understanding of complex system behavior, the proposed approach offers both academic and industrial value. As distributed systems continue to grow in scale and complexity, intelligent triage frameworks grounded in rigorous research will play a critical role in sustaining operational resilience.

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 ACM SIGSAC Conference on Computer and Communications Security, pp. 1285–1298. https://doi.org/10.1145/3133956.3134015

[2] He, P., Zhu, J., Zheng, Z., & Lyu, M. R. (2017). Drain: An online log parsing approach with a fixed depth tree. IEEE International Conference on Web Services, pp. 33–40. https://doi.org/10.1109/ICWS.2017.13

[3] Tang, L., Li, T., Perng, C. S., & Chen, H. (2011). LogSig: Generating system events from raw textual logs. ACM International Conference on Information and Knowledge Management, pp. 785–794. https://doi.org/10.1145/2063576.2063690

[4] Vaarandi, R., & Pihelgas, M. (2015). LogCluster: A data clustering and pattern mining algorithm for event logs. IEEE Conference on Network and Service Management, pp. 1–7. https://doi.org/10.1109/CNSM.2015.7367331

[5] Makanju, A. A., Zincir-Heywood, A. N., & Milios, E. E. (2009). Clustering event logs using iterative partitioning. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1255–1264. https://doi.org/10.1145/1557019.1557154

[6] 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, pp. 149–158. https://doi.org/10.1109/ICDM.2009.60

[7] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. IEEE International Conference on Data Mining, pp. 413–422. https://doi.org/10.1109/ICDM.2008.17

[8] Notaro, P., Cardoso, J., & Gerndt, M. (2021). A survey of AIOps methods for failure management. ACM Transactions on Intelligent Systems and Technology, 12(6), pp. 1–42. https://doi.org/10.1145/3483424

[9] Bento, A., Estêvão, J., Pereira, R., & Mendonça, H. (2021). Automated analysis of distributed tracing: Challenges and research opportunities. Journal of Grid Computing, 19, pp. 1–25. https://doi.org/10.1007/s10723-021-09551-5

[10] Qiu, J., Du, X., Zhang, D., Su, S., & Guizani, M. (2020). A causality mining and knowledge graph based method of root cause diagnosis for performance anomaly in cloud applications. Applied Sciences, 10(6), 2166. https://doi.org/10.3390/app10062166

[11] Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint. https://doi.org/10.48550/arXiv.1702.08608

[12] Nedelkoski, S., Bogatinovski, J., Acker, A., Cardoso, J., & Kao, O. (2020). Self-supervised log parsing. Machine Learning and Knowledge Discovery in Databases, pp. 1–16. https://doi.org/10.1007/978-3-030-67667-4_8

[13] Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Müller, K. R., & Binder, A. (2021). A unifying review of deep and shallow anomaly detection. Proceedings of the IEEE, 109(5), pp. 756–795. https://doi.org/10.1109/JPROC.2021.3052449

[14] Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. ACM Computing Surveys, 54(2), pp. 1–38. https://doi.org/10.1145/3439950

[15] Zhang, W., Meng, W., Zhang, S., Pei, D., Xu, Y., & Liu, H. (2019). LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. International Joint Conference on Artificial Intelligence, pp. 4739–4745. https://doi.org/10.24963/ijcai.2019/658

[16] Breier, J., & Branišová, J. (2015). Anomaly detection from log files using data mining techniques. Communications in Computer and Information Science, 511, pp. 449–457. https://doi.org/10.1007/978-3-662-46578-3_53

[17] Xu, W., Huang, L., Fox, A., Patterson, D., & Jordan, M. I. (2009). Detecting large-scale system problems by mining console logs. ACM SIGOPS Operating Systems Review, pp. 117–132. https://doi.org/10.1145/1629575.1629587

[18] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you: Explaining the predictions of any classifier. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144. https://doi.org/10.1145/2939672.2939778

[19] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), pp. 1–58. https://doi.org/10.1145/1541880.1541882

[[20] Laptev, N., Amizadeh, S., & Flint, I. (2015). Generic and scalable framework for automated time-series anomaly detection. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1939–1947. https://doi.org/10.1145/2783258.2788611

[21] Cohen, I., Zhang, S., Goldszmidt, M., Symons, J., Kelly, T., & Fox, A. (2005). Capturing, indexing, clustering, and retrieving system history. ACM Symposium on Operating Systems Principles, pp. 105–118. https://doi.org/10.1145/1095809.1095821

[22] Gupta, M., Gao, J., Aggarwal, C. C., & Han, J. (2013). Outlier detection for temporal data: A survey. IEEE Transactions on Knowledge and Data Engineering, 26(9), pp. 2250–2267. https://doi.org/10.1109/TKDE.2013.184