International Journal of Scientific Research and Modern Technology (IJSRMT) ijsrmt.com
Volume 4, Issue 8, 2025
DOI: https://doi.org/10.38124/ijsrmt.v4i8.980

Optimizing Application Support Operations with LLMs- Insights from a Self-Healing Interface

Ravikant Singh[1]

[1]Sr. Data Engineering Manager

Publication Date: 2025/08/15

Abstract

Application support operations are increasingly challenged by complex system architectures, high service expectations, and the need for rapid incident resolution. This paper introduces an LLM-powered self-healing framework that automates routine support tasks, reduces mean time to resolution (MTTR), and enhances system resilience through proactive anomaly detection and autonomous remediation. Leveraging data observability as a foundational layer, the framework integrates telemetry aggregation, intelligent root cause analysis, and automated runbook execution to minimize downtime while maintaining transparency and trust. Positioned at the intersection of DevOps and AIOps, this approach aligns with emerging trends in autonomous incident resolution, continuous learning systems, and operational excellence. By reducing manual intervention and enabling proactive maintenance, the framework represents a forward- leaning strategy for optimizing modern application support operations.

Keywords: *Large Language Models (LLMs), Self-Healing Systems, Application Support Operations, Data Observability, AIOps, DevOps, Operational Excellence, Incident Resolution, Autonomous Remediation.*

I. INTRODUCTION

Modern software environments require application support operations to function as an advanced discipline which needs fast responses and precise actions and continuous adaptability. The adoption of cloud-native architectures and microservices and distributed systems has created an exponential increase in enterprise application components and dependencies and failure points (Indukuri, 2025). The operational requirements for support teams have increased because the evolution of technology has enabled scalability and agility but also generated more incidents that need to be addressed quickly while maintaining strict service-level agreements (SLAs) (Indukuri, 2025).

The traditional support approach operates reactively because it starts with incident detection followed by manual investigation and root cause analysis before remediation. The traditional approach results in extended Mean Time to Resolution (MTTR) and produces unpredictable troubleshooting results and fails to maximize operational data potential (Algomox, n.d.). The repetition of basic tasks uses up substantial time and workforce that could be used for developing strategic improvements and solving complex problems (Algomox, n.d.).

Large Language Models (LLMs) provide a revolutionary chance to transform application support operations from reactive firefighting into proactive intelligent autonomous management (Kumar, 2024).

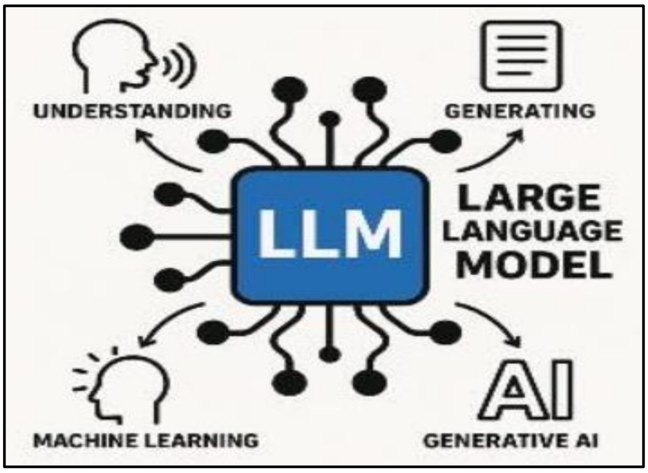

Fig 1 LLMs (GitHub, n.d.).

The combination of natural language understanding with contextual reasoning and pattern recognition through data observability practices enables LLMs to detect anomalies and execute self-healing actions such as service restarts and configuration rollbacks and scaling adjustments without human involvement (Kumar, 2024). Figure 1. Shows the LLMs capabilities.

II. THE LLM-POWERED SELF-HEALING FRAMEWORK

The observability data pipelines and decision-making capabilities of the framework work together to deliver autonomous system healing and proactive operational maintenance that minimizes the need for human interventions (Kumar, 2024).

➢ *Architecture Overview:*

The framework consists of four fundamental architectural pillars:

- *Observability Stack:*

This layer collects logs, metrics, traces and events produced by application components and infrastructure services and external integrations. The LLM benefits from a full system overview through this layer because it processes diverse and context-sensitive data (Kumar, 2024).

- *LLM Reasoning Engine:*

The analytical core functions as the LLM Reasoning Engine which processes observability data for anomaly detection and incident classification and resolution strategy generation. The system employs prompt-engineering methods with domain-specific knowledge to generate accurate and explainable recommendations (Kumar, 2024).

- *Automated Runbook Executor:*

The Automated Runbook Executor executes fixes by communicating with CI/CD pipelines as well as orchestration tools including Kubernetes and Ansible and infrastructure APIs to avoid human involvement (Evans, 2025). The automated process reduces system downtime and accelerates incident resolution time.

- *Feedback and Learning Loop:*

The LLM achieves improved decision-making through continuous feedback loops that use historical incident outcomes to reinforce successful actions and modify anomaly detection thresholds according to system pattern changes (Ahmed, 2023).

➢ *Functional Components:*

- *Data Ingestion & Preprocessing:*

✓ The system collects real-time telemetry data from application performance monitoring tools including Datadog and New Relic and Prometheus (Mireles, 2024).
✓ Raw data receives cleaning operations followed by normalization steps and tagging with contextual information (deployment ID and service owner) to improve analysis quality.

- *Anomaly Detection:*

✓ The system unites statistical anomaly detection with LLM-powered log interpretation to find performance metric deviations from established baselines (Khan, 2025).
✓ The system detects performance degradations at their roots together with symptomatic issues such as CPU spikes and memory leaks before they cause additional damage (Khan, 2025).

- *Root Cause Analysis (RCA):*

✓ The LLM transforms observability signals into human-readable RCA summaries through its synthesis of logs, traces and metrics data (Khan, 2025).
✓ Human operators receive a list of possible causes with associated confidence levels from which they can accept or modify the automated diagnosis results (Khan, 2025).

- *Self-Healing Actions:*

✓ The system performs predefined remediation scripts alongside dynamically generated action plans (Yang, 2025).
✓ The system integrates with IaC frameworks to execute actions which include rolling restarts as well as patch deployments and auto-scaling adjustments (Yang, 2025).

- *Post-Resolution Validation:*

✓ The system tracks essential performance indicators during and after remediation procedures to confirm service health recovery.
✓ The system maintains an incident resolution database which helps improve future incident handling and runbook library development (Khan, 2025).

III. DATA OBSERVABILITY AS THE FOUNDATION

LLM-powered application support operations depend on data quality, timeliness and completeness. The effectiveness of AI-driven frameworks relies on robust data observability, as without it they generate inaccurate diagnoses and ineffective remediation actions. The self-healing framework depends on data observability as its operational foundation (Livens, 2024).

Observability enables LLMs to monitor applications in real-time, detect anomalies, analyze causes and execute autonomous actions. Below Figure 2. Illustrates the observability framework.

Fig 2 Observability (Livens, 2024).

The framework incorporates several important data observability practices which include:

➢ *End-to-End Telemetry Collection:*

- The system collects logs and metrics and traces from every tier of the application stack including APIs, databases, microservices and infrastructure layers (Livens, 2024).
- The LLM maintains complete operational and performance visibility which allows it to identify both individual system failures and broader systemic problems (Livens, 2024).

➢ *Context-Rich Metadata:*

- The system links incidents to deployment records and configuration updates together with dependency connections and ownership assignments (Livens, 2024).
- The LLM uses this contextual information to understand why anomalies occurred along with the fact that they occurred.

➢ *Data Quality Monitoring:*

- The system continuously checks data freshness, accuracy and completeness to stop model decisions from being influenced by outdated or incomplete signals (Atlan., 2024).
- The system generates automated notifications when telemetry pipeline performance drops or when vital data sources lose access.

➢ *Semantic Data Layer:*

- Observability data undergoes standardization through a semantic model structure that matches the reasoning abilities of LLMs.
- The system converts machine data into human-friendly context-specific information that allows the LLM to identify patterns and generate proper recommendations (DataGalaxy., 2025).

Organizations must embed these operational practices into their fabric to ensure the self-healing interface operates with high fidelity while minimizing both false positives and missed anomalies. The transformation of system noise into actionable intelligence occurs through data observability which acts as the essential enabler for autonomous application support operations (Livens, 2024).

IV. DRIVING OPERATIONAL EXCELLENCE

IT operational excellence depends on the fast and precise identification and resolution of incidents. The proposed LLM-powered self-healing framework ensures operational excellence through its automation features and proactive maintenance capabilities as well as adaptive learning and strategic workload distribution. The research evidence from both academic and industrial sources demonstrates that these advantages are valid as shown below Table 1.

Table 1 Industry Performance of LLM-Powered Self-Healing (Alam & El Saddik, 2024).

Industry	Detection Accuracy (%)	Healing Success (%)
Automotive	96.8	91.2
Pharmaceuticals	95.1	88.7
Heavy Machinery	93.4	85.9

➢ *Reduced Mean Time to Resolution (MTTR):*

- The reported study revealed a 55% reduction in Mean Time to Recovery (MTTR variant) with a 208-fold boost in deployment frequency and sustained developer trust demonstrating actual operational benefits from LLM-driven systems (Kumar & Zhang, 2024).
- Research findings indicated fault detection and self-healing systems achieved a 31.7% decrease in MTTR alongside 97.3% fault detection precision and 89.4% self-healing effectiveness proving that autonomous agents can deliver significant transformative power (Alam & El Saddik, 2024).

➢ *Proactive Maintenance:*

- Transforming operational models from reactive to proactive by using real-time anomaly detection, AI-driven event correlation and self-healing workflows to achieve near-zero downtime and optimal resource utilization (Rohith Samudrala, 2024).
- AI-driven predictive fault analysis in AIOps operation leads to decreased downtime periods while simultaneously improving system reliability together with resource optimization according to research findings (Patel & Nguyen, 2024).

➢ *Knowledge Retention & Continuous Learning:*

- Resolved incidents serve as learning opportunities which improve the AI's remediation strategies through model refinement. The operational engine develops smarter context- based capabilities through ongoing adaptation during this process (Software., 2024).
- Operator feedback combined with increasing model interpretability enables hybrid AI systems that consist of ML and ML-driven anomaly detection and automated root cause analysis to develop operational knowledge through a defined framework (Ahmed, 2023).

➢ *Workload Optimization:*

- The operational agents powered by AI decrease manual incident assessment processes while allowing proactive responses and enabling engineers to shift their efforts toward architectural development and strategic improvements. The blog about reducing MTTR and alert fatigue explains that operational agents powered by AI work as human expertise enhancers rather than replacements which result in better efficiency and reduced cognitive strain (Algomox, n.d.).
- The comprehensive operational efficiency impact of AIOps through its ML techniques and automation capabilities demonstrates the platform's transformative ability to optimize incident response workflows (Hamilton).

V. ALIGNMENT WITH DEVOPS TRENDS

The proposed LLM-powered self-healing framework demonstrates how AI technology merges with current DevOps methods to support industry trends for automated systems and reliable proactive management.

➢ *Autonomous Incident Resolution:*

- The current AIOps research indicates that modern infrastructures are moving away from reactive monitoring to develop self-correcting capabilities that fix problems automatically (Booz Allen Hamilton, n.d.; Algomox, n.d.).
- The framework enables autonomous incident resolution through its anomaly detection and automated runbook execution which decreases MTTR and stops incident recurrence.

➢ *AIOps:*

- AIOps represents the use of artificial intelligence in IT operations which deliver enhanced anomaly detection and predictive analytics together with automated operational processes (Singh, 2023).
- The framework improves decision-making speed and accuracy through LLM reasoning on observability data which matches current enterprise AIOps strategies.

➢ *Shift-Left Reliability:*

- The software delivery lifecycle benefits from “shift-left” principles which require developers to embed resilience together with observability and maintainability at the beginning (Lee, 2023).
- The framework supports this by integrating with CI/CD pipelines and proactively identifying potential failure points during development and staging.

➢ *Continuous Learning Systems:*

- DevOps literature indicates that systems which operate under changing workloads and architectural evolution need to adopt continuous adaptation according to recent research (Indium Software, 2024).

- The feedback and learning loop within the framework allow predictive accuracy and remediation strategies to improve continuously which makes it adaptable to new failure modes.
- The framework establishes itself as an operational stability tool while serving as a strategic enabler for the development of fully autonomous intelligent DevOps ecosystems through its address of these four trends.

VI. CHALLENGES AND CONSIDERATIONS

The implementation of an LLM-powered self-healing framework presents substantial operational efficiency and resilience potential yet multiple obstacles need resolution for safe reliable sustainable deployment.

➢ *False Positives and Negatives in Anomaly Detection:*

- The combination of overly sensitive thresholds with insufficient training data leads to two major issues: false positives that initiate unnecessary remediation actions and false negatives that fail to detect actual incidents (Alam & El Saddik, 2024).
- The system's recommendations suffer from reduced trust while resources are wasted and unplanned downtime occurs because of this issue.
- The detection algorithm requires continuous tuning for risk mitigation through the combination of statistical and LLM reasoning methods (Alam & El Saddik, 2024).

➢ *Model Drift:*

- The LLM's predictions become less accurate because application behaviors together with infrastructure configurations and workload patterns naturally change throughout time.
- The system requires periodic retraining with new observability data and automated drift detection systems to preserve its accuracy (Singh, 2023).

➢ *Security Risks in Autonomous Remediation:*

- Security risks emerge when automated agents receive the power to modify infrastructure because it creates potential risks for privilege escalation and malicious exploitation.
- The exposure can be minimized through the implementation of role-based access control (RBAC) together with multi-step verification for high-impact changes and secure API handling (Algomox, 2025).

➢ *Explainability and Engineer Trust:*

- Operations teams will only accept AI-driven resolutions when engineers can understand the reasoning behind specific actions.
- The implementation of explainable AI (XAI) techniques which include human-readable RCA summaries and ranked confidence scores, and decision rationale enhances transparency and adoption rates (Algomox, 2025).

VII. FUTURE SCOPE

Large language models (LLMs) show great promise for optimizing application support operations with self-healing interfaces in the future because of their sector-wide advances and capabilities. The following list contains essential future directions alongside research areas for exploration:

➢ *Enhanced System Adaptability:*

The system adaptability will increase through LLM applications that develop self-healing systems which can automatically diagnose and solve software environment issues. The ability of these models to handle extensive data sets while extracting valuable information leads to improved system self-regulation and optimization (Rengers et al., 2024).

➢ *Integrating Self-Healing Mechanisms:*

Future research should develop self-healing workflows that automatically detect and repair application anomalies with LLM integration. Operational efficiency will greatly increase because LLMs reduce both downtime and maintenance expenses (Li et al., 2024).

➢ *Advanced Predictive Analytics:*

LLMs enable the development of predictive analytics tools for self-healing interfaces which forecast system failures before they occur. System resilience along with service reliability improves when organizations adopt proactive strategies (Tailor et al., 2024).

➢ *Ethical and Practical Considerations:*

The integration of LLMs into self-healing systems requires immediate attention to ethical issues which include data privacy protection and security measures and consent management. Widespread acceptance of critical application domains requires ethical systems designed according to established guidelines (Jiang et al., 2024).

➢ *Cross-Domain Innovations:*

The successful applications of LLMs in healthcare and bioinformatics offer opportunities to develop innovative software support operations which span multiple domains. Interdisciplinary approaches between different fields could produce new solutions that strengthen the capabilities of self-healing systems (Liu et al., 2024; Mbadjeu Hondjeu et al., 2025).

➢ *User Interface and Experience:*

Future research should investigate ways LLMs can improve both interfaces and user experiences of self-healing systems to create more intuitive user-friendly systems. The optimization of user interactions with self-healing systems through improvements leads to enhanced operational efficiency (Su et al., 2024).

➢ *Customization and Personalization:*

The development of personalized self-healing solutions requires significant attention to create frameworks for such customized systems. LLMs help organizations build customized interfaces along with solutions which match their specific organizational needs (Maity & Saikia, 2025).

VIII. CONCLUSION

LLMs integrated into application support operations through self-healing interfaces show great potential for optimization. Support operations will undergo a transformation through LLMs because they automate basic work and speed up responses while maximizing operational efficiency. The management and monitoring of applications receive enhanced capabilities through LLMs which surpass traditional methodologies by implementing a dynamic data-driven approach (Pahune and Akhtar, 2025). The main advantage of using LLMs in support operations stems from their capacity to process extensive data sets and extract new types of insights that standard systems cannot access. Organizations can develop interfaces through LLM natural language processing capabilities which automate process tasks and perform real-time issue detection and resolution to function as self-corrective systems (Pahune and Akhtar, 2025). The implementation of LLMs faces two major obstacles which include protecting data privacy and addressing biases that exist within AI systems (Feretzakis et al., 2025). The operational framework for LLMs requires a distinct LLMOps design that addresses the special needs and intricate aspects of LLMs beyond traditional MLOps strategies (Pahune and Akhtar, 2025). The successful implementation of LLMs for optimizing application support operations requires thorough examination of ethical issues and operational and regulatory challenges. Businesses can improve support operation efficiency and reliability through robust LLMOps practices and self- healing interfaces which promise better service quality and lower operational costs.

REFERENCES

[1]. Ajay Varma Indukuri. Cloud-native transformation: Architectural principles and organizational strategies for infrastructure modernization. World Journal of Advanced Research and Reviews, 2025, 26(01), 3914-3926. Article DOI: https://doi.org/10.30574/wjarr.2025.26.1.1467.

[2]. Kumar, P. (2024). Large language models (LLMs): Survey, technical frameworks, and future challenges. Artificial Intelligence Review, 57(260). https://doi.org/10.1007/s10462-024-10888-y.

[3]. Algomox. (n.d.). Reduce MTTR and alert fatigue with AIOps. Algomox. https://www.algomox.com/resources/blog/reduce_mttr_alert_fatigue_aiops/.

[4]. GitHub. (n.d.). LLM repositories written in JavaScript, sorted by forks. GitHub. Retrieved from https://github.com/topics/llm?l=javascript&o=asc&s=forks.

[5]. Evans, D. (2025, August 15). Platform engineering control mechanisms. Google Cloud Blog. https://cloud.google.com/blog/products/application- modernization/platform-engineering-control-mechanisms.

[6]. Ahmed, S., Rahman, A., & Ashrafuzzaman, M. (2023). A systematic review of AI and machine learning-driven IT support systems: Enhancing efficiency and automation in technical service management. American Journal of Scholarly Research and Innovation, 2(02), 75–101. https://doi.org/10.63125/fd34sr03.

[7]. Mireles, Y. (2024, November 26). What is APM (Application Performance Monitoring). New Relic. https://newrelic.com/blog/best-practices/what-is-apm.

[8]. Khan, A. F., Khan, A. A., Mohamed, A., Ali, H., Moolinti, S., Haroon, S., Tahir, U., Fazzini, M., Butt, A. R., & Anwar, A. (2025). LADs: Leveraging LLMs for AI-Driven DevOps [Preprint]. arXiv. https://arxiv.org/abs/2502.20825.

[9]. Yang, Z., Jin, Y., Liu, J., & Xu, X. (2025). An intelligent fault self-healing mechanism for cloud AI systems via integration of large language models and deep reinforcement learning [Preprint]. arXiv. https://arxiv.org/abs/2506.07411.

[10]. Livens, J. (2024, June 26). What is observability? Not just logs, metrics, and traces. Dynatrace. https://www.dynatrace.com/news/blog/what-is-observability-2/.

[11]. Atlan. (2024, December 26). Data quality and observability: Key differences & relationships! https://atlan.com/data-quality-and-observability.

[12]. Data Galaxy. (2025, February 11). Using a semantic layer to bring context to AI. https://www.datagalaxy.com/en/blog/data-context-meets-ai/.

[13]. Alam, S., & El Saddik, A. (2024). Autonomous artificial intelligence agents for fault detection and self-healing in smart manufacturing systems. ResearchGate. https://www.researchgate.net/publication/393520706_Autonomous_Artificial_Intelligence_Agents_for_Fault_Detection_and_SelfHealing_in_Smart_Manufacturing_Syste ms.

[14]. Kumar, A., & Zhang, Y. (2024). Scaling generative AI for self-healing DevOps pipelines: Technical analysis. Research Gate. https://www.researchgate.net/publication/392768295_Scaling_Generative_AI_for_Se lf-Healing_DevOps_Pipelines_Technical_Analysis.

[15]. Rohith Samudrala, L. N. (2024). The future of AI in incident management. International Journal of Science and Advanced Technology, 14(3), 45–52. https://www.ijsat.org/papers/2024/3/2840.pdf.

[16]. Patel, R., & Nguyen, T. (2024). AIOps in action: Streamlining IT operations through artificial intelligence. ResearchGate.https://www.researchgate.net/public

ation/391850093_AIOps_in_Action_Streamlining_IT_Operations_Through_Artificial_Intelligence_1.
[17]. Indium Software. (2024, February 20). Gen AI for application support. Indium Software. https://www.indium.tech/blog/gen-ai-for-application-support/.
[18]. Booz Allen Hamilton. (n.d.). Accelerating incident response with AIOps. Booz Allen Hamilton. https://www.boozallen.com/insights/ai-research/accelerating-incident- response-with-aiops.html.
[19]. Singh, M., & Chen, L. (2023). AIOps for IT operations: Trends, challenges, and opportunities. ResearchGate. https://www.researchgate.net/publication/376545899_AIOps_for_IT_operations.
[20]. Lee, H., & Martinez, J. (2023). Shift-left reliability: Embedding resilience earlier in the DevOps lifecycle. ResearchGate. https://www.researchgate.net/publication/376552421_Shift-Left_Reliability.
[21]. Algomox. (2025, July 22). Self-healing infrastructure with agentic AI. Algomox. https://www.algomox.com/resources/blog/self_healing_infrastructure_with_agentic_a i/.
[22]. Jiang, X., Yan, L., Vavekanand, R., & Hu, M. (2024). Large Language Models in Healthcare Current Development and Future Directions. Mdpi Ag. https://doi.org/10.20944/preprints202407.0923.v1.
[23]. Li, J., Jin, Z., Zhang, M., Weyns, D., Tei, K., & Li, N. (2024). Exploring the Potential of Large Language Models in Self-adaptive Systems. 77–83. https://doi.org/10.1145/3643915.3644088.
[24]. Liu, J., Yang, M., Yu, Y., Xu, H., Wang, T., Li, K., & Zhou, X. (2024). Advancing bioinformatics with large language models: components, applications and perspectives. https://doi.org/10.48550/arxiv.2401.04155.
[25]. Maity, S., & Saikia, M. J. (2025). Large Language Models in Healthcare and Medical Applications: A Review. Bio-Engineering (Basel, Switzerland), 12(6). https://doi.org/10.3390/bioengineering12060631.
[26]. Mbadjeu Hondjeu, A. R., Zhao, Z. Y., Newton, L., Ajenkar, A., Hladkowicz, E., Ladha, K., Wijeysundera, D. N., & Mcisaac, D. I. (2025). Large language models in perioperative medicine—applications and future prospects: a narrative review. Canadian Journal of Anesthesia/Journal Canadien d'anesthésie, 72(6), 1000–1014. https://doi.org/10.1007/s12630-025-02980-w.
[27]. Rengers, T. A., Thiels, C. A., & Salehinejad, H. (2024). Academic Surgery in the Era of Large Language Models. JAMA Surgery, 159(4), 445. https://doi.org/10.1001/jamasurg.2023.6496.
[28]. Su, Z., Tang, G., Huang, R., Qiao, Y., Zhang, Z., & Dai, X. (2024). Based on Medicine, The Now and Future of Large Language Models. Cellular and Molecular Bioengineering, 17(4), 263–277. https://doi.org/10.1007/s12195-024-00820-3.
[29]. Tailor, P. D., D'Souza, H. S., Li, H., & Starr, M. R. (2024). Vision of the future: large language models in ophthalmology. Current Opinion in Ophthalmology, 35(5), 391–402. https://doi.org/10.1097/icu.0000000000001062.
[30]. Feretzakis, G., Vagena, E., Peristera, P., Kalles, D., Anastasiou, A., & Kalodanis, K. (2025). GDPR and Large Language Models: Technical and Legal Obstacles. Future Internet, 17(4), 151. https://doi.org/10.3390/fi17040151.
[31]. Pahune, S., & Akhtar, Z. (2025). Transitioning from MLOps to LLMOps: Navigating the Unique Challenges of Large Language Models. Information, 16(2), 87. https://doi.org/10.3390/info16020087.