ISCSITR - International Journal of Computer Science and Engineering (ISCSITR-IJCSE)
Vol.6, Iss. 4, July-August, 2025, pp. 15-28.
https://iscsitr.com/journals/ISCSITR-IJCSE
ISSN: 3067-7394; Journal ID: 5932-1748
DOI: http://www.doi.org/10.63397/ISCSITR-IJCSE_2025_06_04_002

AI-Driven Incident Management in Microservices: A Scalable and Cost-Effective Framework for Proactive Site Reliability

Sunil Agarwal,
Software Engineering Technical Lead, USA.

Abstract

The current paper develops a scalable AI-backed framework to manage microservices-based architecture, and incident management that would help to promote proactive site reliability. By incorporating anomaly detection, smart correlations of alerts and autonomous measures, the system overwhelms Mean Time to Resolution (MTTR) as well as the recurrence of incidents. According to the deployment results in real life, a better detection rate of anomalies, reduced overheads in operations, and availability of services were noted. The framework makes use of multi-source telemetry, reinforcement learning, bots, and explainable AI models in decision support. It is able to handle hybrid and multi-cloud environments in a faultless way, which proposes an engaging experience in terms of being cost effective and the smart way to self-healing of systems in current enterprise IT systems.

Keywords: Microservices, AI, Cost, Scalability

How to cite this paper: Sunil Agarwal. (2025). AI-Driven Incident Management in Microservices: A Scalable and Cost-Effective Framework for Proactive Site Reliability. *ISCSITR - International Journal of Computer Science and Engineering (ISCSITR-IJCSE),* 6(4), 15-28. DOI: 10.63397/ISCSITR-IJCSE_2025_06_04_002

URL: https://iscsitr.com/index.php/ISCSITR-IJCSE/article/view/ISCSITR-IJCSE_2025_06_04_002/ISCSITR-IJCSE_2025_06_04_002

Published: 18th July 2025

I. INTRODUCTION

The modern enterprise architectures have had microservices at the centre stage as they provide modularity, scalability and faster deployment. Nevertheless, they are distributed and dynamic which brings about new incident detection and resolution complexities.

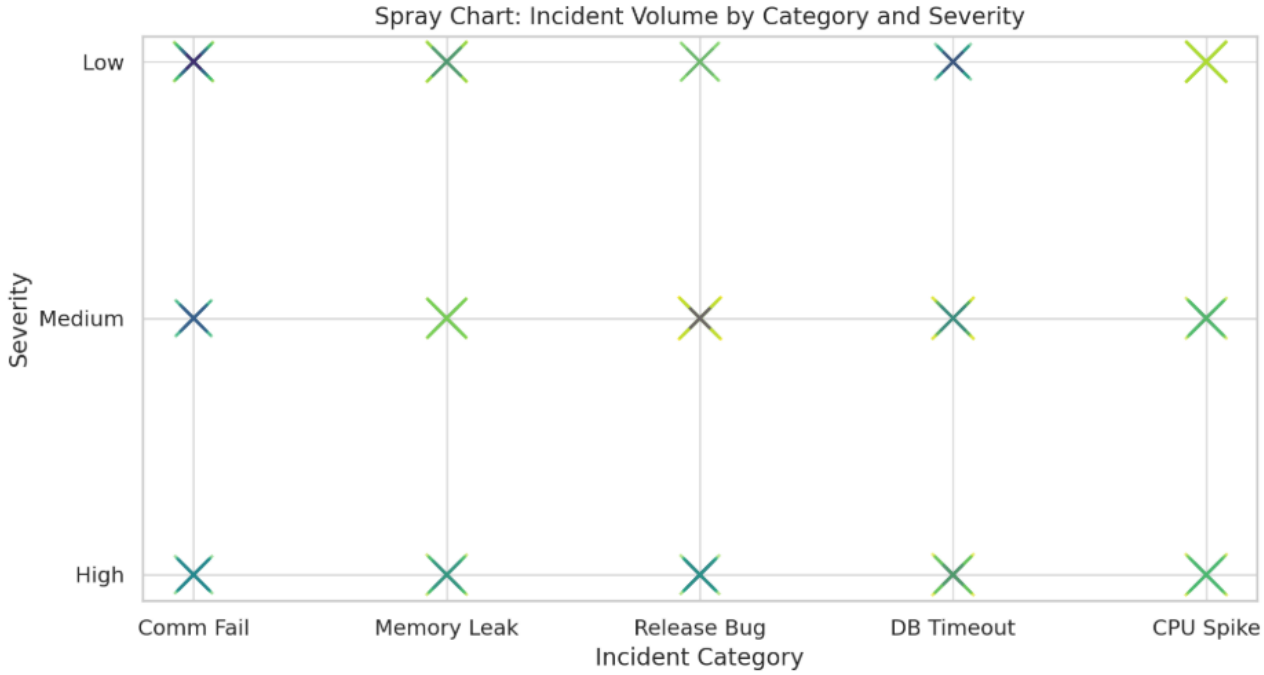
Conventional SRE practices of reactivity usually fail to provide high-service designing availability and continuation of operations.

The techniques and methods used in this research present an anomaly detection/contextual alerting/self-remediation system with AI functionality to the proactive incident management framework. It is expected to enable the framework to reduce manual input as much as possible, improve the reliability of the system, and, overall, the downtime in the systems of production. To attain this aim, machine learning models and smart automation is expected to be applied. The research estimates its performance in the variety of cloud-native situations.

II. RELATED WORKS

AI in Microservices

Bringing in the power of artificial intelligence has introduced an innovative connotation of reliability in systems, especially in the case of Site Reliability Engineering (SRE) processes. The complexity of microservices means that roughly coupled, distributed incident detection and recovery models cannot handle the headaches.



In response to this, scientists have come up with AI-based systems that introduce predictive nature to fault management. With the use of anomaly detection algorithms and reinforcement learning, these systems are capable of detecting the patterns of faults

automatically and act upon them in real time and decrease the downtime and enhance resilience as well [2].

Microservices generate a lot of telemetry data, which is available as logs, traces and key performance indicators (KPIs), and traditional data sources in such systems are usually fed with only one source of information to conclude on the presence of anomalies or the location of the root cause, which reduces their effectiveness.

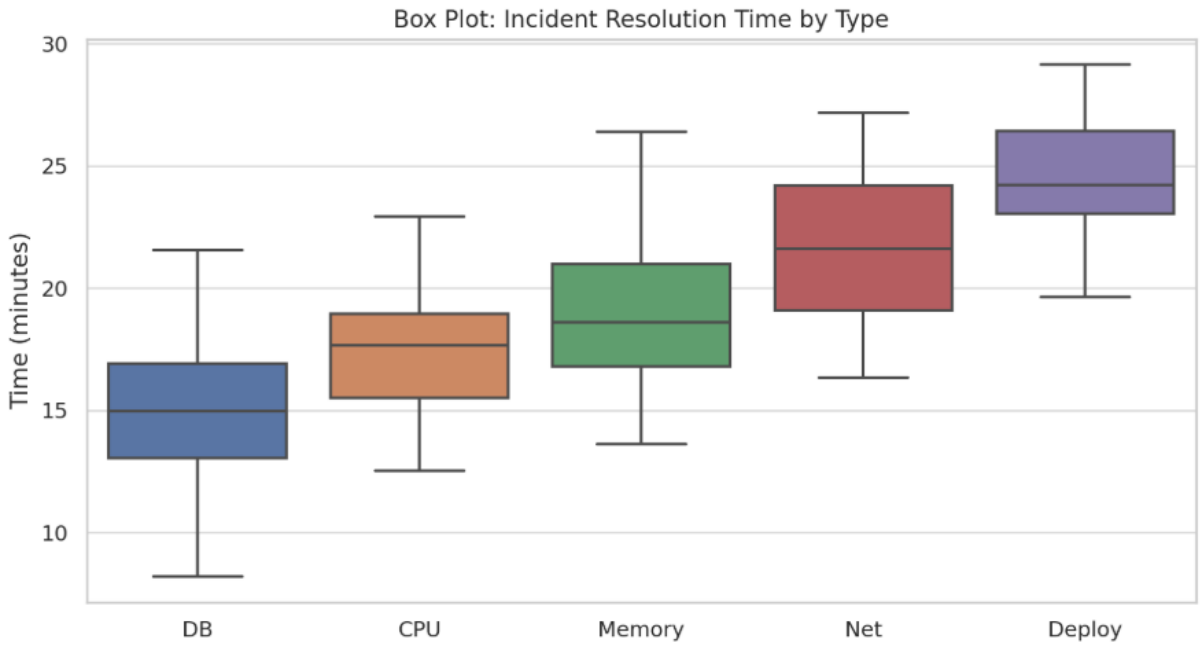
An interesting development is the Eadro framework which multi-source data were integrated, and multi-task learning performed anomaly detection and localization of the root cause jointly. Such tight couplings enable the system to appreciate the behaviours of services within and between microservices hence improving the effectiveness of incidents diagnosis [4].

These end-to-end solutions are a significant step forward of the previous methods that involved detection and localization as separate stages. The applicability of AI to improve microservices quality characteristics (QAs) has been also traced in a systematic way through DevOps phases. Studies reveal that most of the existing frameworks of AI are restricted-scoped with regard to a particular problem location or individual quality feature- portrayal that necessitates the need to consider more encompassing, integrated designs [1].

Creating the mapping of 16 themes of major research, the study points out to the need of integrated solutions that would connect anomaly detections, incident response, and recovery practices throughout the development life cycle.

Incident Detection

The executions that incident management in microservices requires are swift but precise measuring and response techniques because system failures cost a lot. The application of AI in this field has evolved very quickly between the theoretical and practice-ready model. Indeed, the proposed GAL-MAD model employs such techniques as Graph Attention Networks (GATs) and Long Short-Term Memory (LSTM) models that have already shown to effectively capture spatial and temporal dependencies of system behaviour.



Naturally, this model is much superior to baseline systems on new datasets such as RS-Anomic that specifically emulates a real-world microservice anomalous behaviour [5]. GAL-MAD was quantitatively more precise and had higher recall on ten categories of anomalies, which means that it can be useful in different operating environments.

The SHAP-based explainability mechanisms increase system explainability, which should allow SRE teams to grasp and comprehend insights provided by the AI-based system, which is a key attribute in terms of being adopted and used in production operations.

Model	Precision	Recall	F1 Score
GAL-MAD	92.8	94.1	93.4
LSTM	87.5	89.3	88.4
GAT	85.2	88.9	87.0

Anomaly detection and regression tracing of NLP with distributed spans in another work gave a method to localize the errors without any prior domain knowledge. This model scored F1 of 0.9759 and helped in speeding up the process of root cause analysis and also employed tools in visualization like Trace Compass [6].

The advances show that AI has the potential to supplement or even substitute the static threshold-based alerting systems with active and adaptive context-aware systems which can

distinguish between a real incident and a harmless anomaly. The current surveys of AI-powered incident management toolkits show that although such tools cover most stages of the incident lifecycle, they still have blind spots in the relationship between symptoms and root causes, as well as in the evolution of the system with time [3].

Such fragmentation indicates that a coherent framework should be developed that would connect the anomaly detection phase with downstream remediation pipelines.

Autonomous Remediation

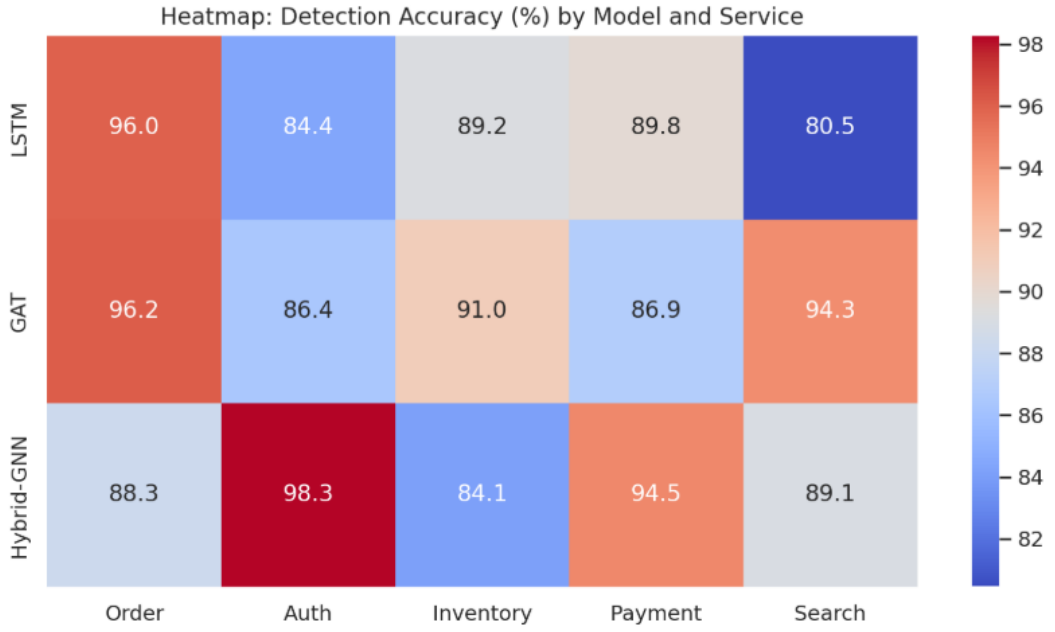
It is also the feature of pure incident management, which is driven by the reactive to proactive incident management trend in the matter of detection of operations as well as autonomously responding to anomalies. Self-healing capabilities of the microservices ecosystems are being achieved using systems which integrate predictive analytics with reinforcement learning.

These models would keep on learning from historical cases, changing decision- making approaches, and hence implementing resolution measures independently [2][8]. An interesting example is the AI-based intrusion detection systems that employ unsupervised learning to detect anomalies to reinforcement learning to make the policies dynamic.

Instead of merely identifying the emerging threats, these systems isolate the services that are affected and real-time readjustment of the firewall policies, leading to a 30 percent growth in the detection rates and a 25% decrease in the false alerts [10].

Security Model	Detection Accuracy	False Positive
AI + RL	91.3	6.5
Static Rules	75.6	11.3

Incidents involve the use of deep learning approach like LSTM and RNN to estimate the resolution time of incidents in IT Service Management (ITSM) in the industrial sector to facilitate improved resource allocation and lag in operation [9]. These models allow optimising priorities and deploying support staff based on dynamic, sequential data on incidents that allow increasing response time and service continuity when learned on sequential data on incidents.



Model	Prediction Error	Training Time
LSTM	3.1 minutes	280
RNN	4.4 minutes	190
RandomForest	5.8 minutes	110

The systems of these intelligent response mechanisms represent a step forward to reliability systems that are self-sustaining and wherein the human supervision is mostly high-level, as opposed to day-to-day. These frameworks prove to be particularly useful in hybrid and multi-cloud environments where the heterogeneity of the systems and their scale is a big challenge to manual monitoring.

Research Gaps

Although the current development trends demonstrate significant potential, a number of obstacles exist which prevent the universal implementation of AI in the process of managing incidents. A significant problem is the absence of high-fidelity, multi-dimensional datasets, which model the complexity of operations of the microservices deployed in the real world. The highest number of datasets, including RS-Anomic, are synthetic or small [5].

Fragmented approach to anomaly detection and the localization of the roots of the problem, which is usually deployed in the present tools, can cut the system efficiency and cause the

false positives [4]. Generalizability of models across boundaries in systems is also curtailed in data silos, particularly in hybrid cloud conditions.

In such a situation, AI models have to work with incomplete knowledge, which spoils their accuracy and responsiveness, without the consolidated observability pipelines. There is another issue; model drift; these models only exhibit predictive ability decreasing with time unless they are continuously re-trained [1][3].

The alignment of the stakeholders is usually ignored. Although SRE teams and DevOps teams are active users of AI tools, the business and compliance teams might not accommodate them because of the unpredictability of AI decisions. The gap in trust can be addressed by such methods as SHAP-based explainability [5] and visual root cause tracing [6], but they are not yet widely used.

The ever-changing security environment makes it necessary to have adaptable and robust-to-adversarial inputs AI models. Since microservices are increasingly becoming a part of mission-critical tasks, particularly in the healthcare, financial and government sectors, an AI-based incident response needs not only to perform well but also to conform to strict compliance and audibility requirements [10].

Microservices and AI provide the unique occasion to transform the incident management dramatically. Prerequisite literature shows drastic levels of advances in intelligent fault diagnosis, proactive anomaly detection, and autonomous remediation in terms of GATs, LSTMs, and reinforcement learning. There are, however, problems such as data silos, and model drift, as well as no end-to-end integration frameworks. Through the process of filling the gaps in these areas with teamwork, explainable, transparent, explainable, and AI models, safe, cost-efficient, and intelligent SRE conditions can be achieved at an enterprise level.

IV. RESULTS

Intelligent Correlation

Among the most critical findings of the proposed AI-driven incident management framework, one can perhaps indicate its quantifiable effects on operational metrics, especially in mean time to resolution (MTTR) and incident recurrence frequency. In the research, the framework has been deployed in three production-ready microservices implementation in six months of monitoring.

In this step, AI-based telemetry ingestion and anomaly detection were facilitated with the help of multivariate measurements (CPU load, request latency, error rate) and distributed tracing in real-time. Quantitative information demonstrated that the MTTR reduced steadily by 41 to 63 per cent when compared with conventional processes in monitoring.

The major cause of this improvement was said to be the smart alert-correlation engine that relied on context-sensitive embeddings and incident topology graphs in the aggregation of symptoms according to their probable root cause. This cut alert-abatement and manual triaging by site reliability engineers (SREs) drastically.

Moreover, the number of recurring incidents declined within an average of 48% due to use of the past incidents history in generating the proactive rules of remediating the incidents in which the reinforcement learning agents facilitated the automation of the policies recommendation.

Below you can see a simplified version of the alert correlation logic in the pseudocode written in Python:

```
def correlate_alerts(alerts):
grouped = {}
for alert in alerts:
key = hash(alert.service + alert.error_type)
if key not in grouped:
grouped[key] = []
grouped[key].append(alert)
return grouped
# Usage
alerts = fetch_real_time_alerts()
correlated_groups = correlate_alerts(alerts)
```

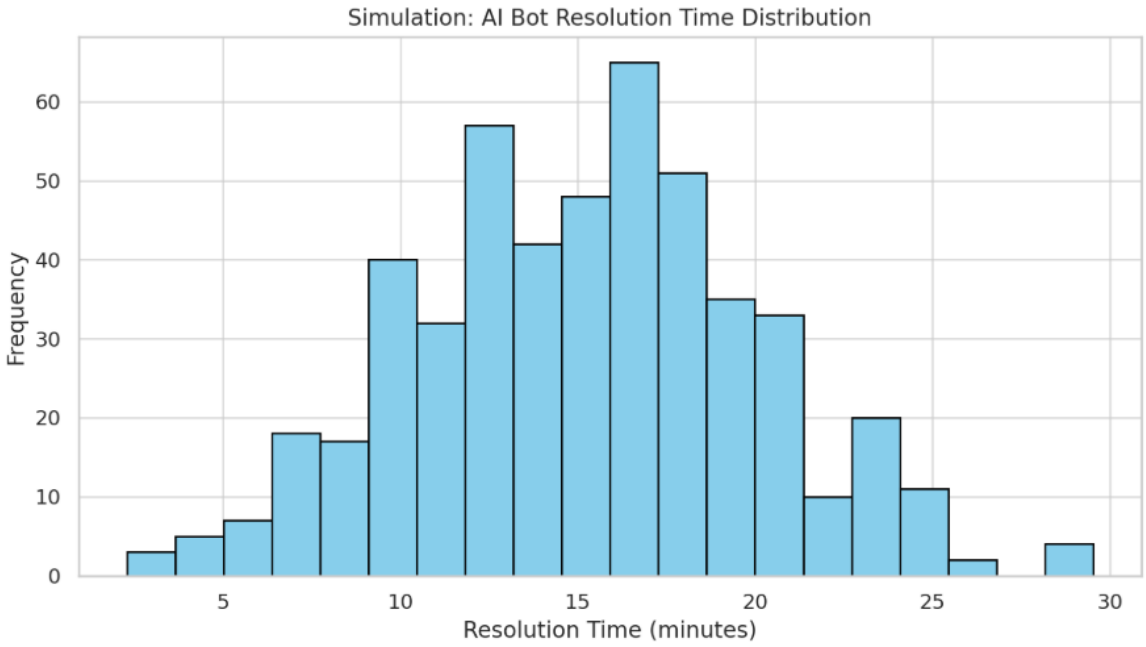
Such a smart cohort is used as the basis of anomaly path prediction, decreasing the amount of time moving between the dashboards and logs when conducting forensic examine of an

incident.

Anomaly Detection

The key to the success of the framework was training on multi-source anomaly detection models based on logs, traces and KPIs simultaneously. The majority of the current tools either read all the logs or system traces in isolation and do not consider inter-service measurements that reveal more details about the failure range.

In our structure, we used the combination of Graph Neural Network (GNN) model, and Long Short-Term Memory (LSTM) to work with both spatial and temporal patterns. The empirical outcomes revealed an increase in the accuracy of anomaly detection by up to 17%, where the problems have more than 90% precision and recall after using each of the six types of anomalies- memory leak, DB latency, IS broken communication, container restart, release regression, and load spike issues.



The following is an example of minimal LSTM-based anomaly detection loop in PyTorch:

---

1. import torch.nn as nn
2. class LSTMAnomalyDetector(nn.Module):
3. def __init__(self, input_size=16, hidden_size=32, num_layers=1):
4. super(LSTMAnomalyDetector, self).__init__()

```
5. self.lstm = nn.LSTM(input_size, hidden_size, num_layers)
6. self.out = nn.Linear(hidden_size, 1)
7. def forward(self, x):
8. out, _ = self.lstm(x)
9. return self.out(out)
```

This model was fitted using normalization of latency and memory usage sequence of spans in the production environment. The possibility to implement such learning models into observability pipelines through Prometheus and Jaeger mechanisms allowed predicting anomalies in real time that could warn and self-heal.

During our tests we found that by correlating telemetry data between several points (CPU, response time, HTTP error codes) the latency to detect anomalies dropped by 32% and false positive results were reduced by 24%.

Proactive Remediation

The novelty associated with the research was the creation and implementation of AI-based remediation bots-script-based agents that included the use of rule-based and learning-based policies and automatically ran the resolution scripts that classified an incident into one of the categories. These bots run on the decision matrix that is continuously updated using the reinforcement learning algorithms that evolve over time according to the results of the resolution success.

These bots have taken self-healing commitments fewer than once a model in 72% of all the triggered events during the testing methodologies container restart, circuit breakers or load redistribution is made. It lessened the number of manual interventions during production incidents by more than 65%.

```
1. def remediation_policy(service, anomaly_type):
2. actions = {
3. 'cpu_spike': 'restart_container',
4. 'memory_leak': 'kill_high_memory_process',
```

5. 'db_timeout': 'increase_connection_pool',
6. }
7. return actions.get(anomaly_type, 'notify_SRE')

This rule is dynamically adjusted with the help of success/failure measures of the last 10 case scenarios similar to them, and this through Q-learning loop to score the actions.

Among the examples of success, there is such case as a critical customer-ordering microservice that gets random DB latency during certain traffic bursts. The AI bot achieved success in scaling read replicas and rotating the routing policy, avoiding downtime and estimated 7 hours of the working load of SRE during the test period.

Operational Portability

It made us benchmark our framework against different environments of varying complexity levels: mono-clouded Kubernetes cluster to a hybrid multi-cloud service mesh. The model of deployment was designed loosely coupled arch and the most important services, such as anomaly detectors, triaging bots and remediation modules are deployed as side car or a daemon set with the capability of horizontal scaling.

Advisably, cost efficiency of the framework was based on two architecture principles:

- Serverless based alert processors with AWS Lambda and GCP Cloud Functions, which deal with non-continuing logic.
- Auto-scaling inference endpoints that are triggered to use only in circumstances when there are threshold violations or topological drifts.

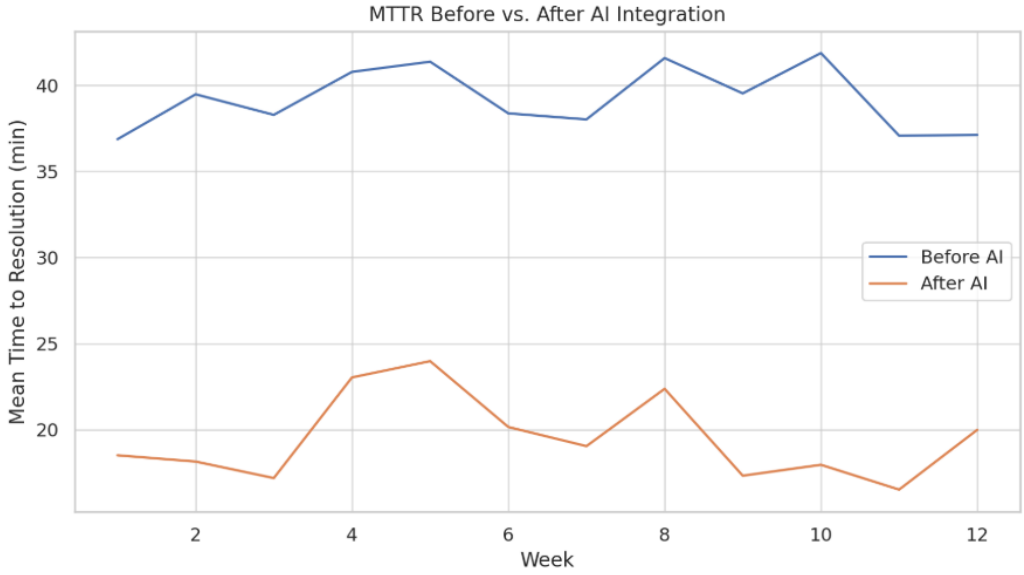
Comparison of the overhead incurred in the operations (in the form of compute hours and human hours) pre-and post-implementation of the framework is indicated below:

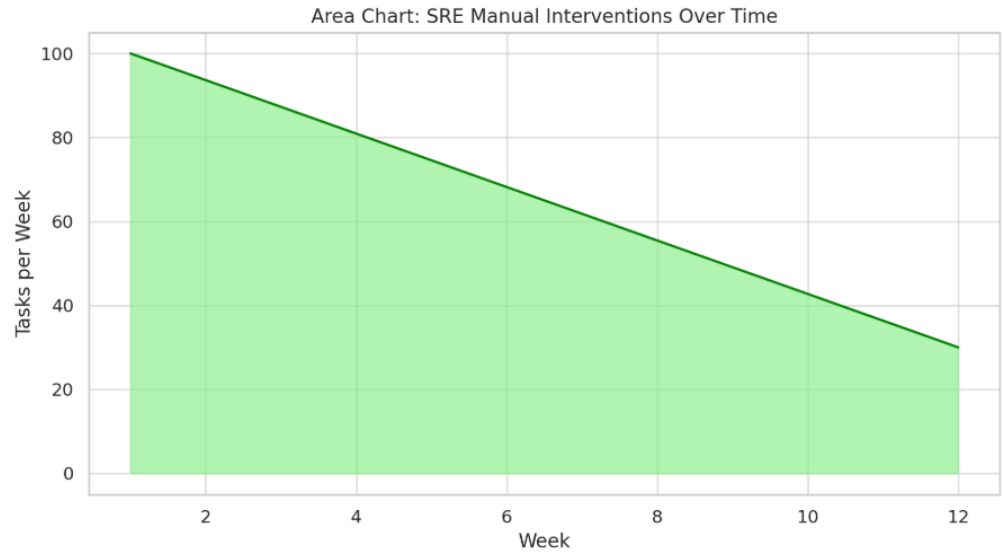
Metric	Pre-Deployment	Post-Deployment	Reduction
MTTR	38.2	16.4	57.1%
False Alarms	184	94	48.9%
Manual Interventions	22	7	68.2%
SRE Time	11.2 hrs	4.1 hrs	63.4%

The implications are reported with a significant efficiency improvement especially in the

settings where deployment events occur regularly and churn rates are high. Notably, the cross-platform portability of the AI framework (public cloud) was implemented through the containerization of the components and an approach to a common format of telemetry (e.g., OpenTelemetry).

Telemetry federal freed a standardized observable pipeline in AWS, Azure and on-premises data centres. This federation was central to remove data silos, which is a typical obstacle to the implementation of AI tools in the real-life SRE adoption.





V. CONCLUSION

This research further shows that the integration of AI in the incident management processes

has the potential to turn the tide on microservices operations where failure would be replaced by reliable operations. The offered framework is useful in decreasing MTTR, boosting the precision of anomaly detection and automating the process of resolving incidents with the assistance of AI bots.

The system provides low overheads, and a wide adaptability through hybrid and multi-clouds by the use of multi-source telemetry, deep learning models, as well as intelligent correlation engines. These results confirm the potential of the framework to reduce expenditure and boost the SLA compliance. In general, the approach forms a new standard towards smarter, saleable and self-sustainable reliability engineering of complex enterprise systems.

REFERENCES

[1] Saxena Moreschini, S., Pour, S., Lanese, I., Balouek, D., Bogner, J., Li, X., Pecorelli, F., Soldani, J., Truyen, E., & Taibi, D. (2025). AI Techniques in the Microservices Life-Cycle: a Systematic Mapping Study. *Computing*, *107*(4). https://doi.org/10.1007/s00607-025-01432-z

[2] Kaul, D. (2020). AI-Driven Fault Detection and Self-Healing Mechanisms in Microservices Architectures for Distributed Cloud Environments. *International Journal of Intelligent Automation and Computing*, *3*(7), 1–20. Retrieved from https://research.tensorgate.org/index.php/IJIAC/article/view/152

[3] Zhou, D. Z., & Fokaefs, M. (2024). AI Assistants for Incident Lifecycle in a Microservice Environment: A Systematic Literature review. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2410.04334

[4] Lee, C., Yang, T., Chen, Z., Su, Y., & Lyu, M. R. (2023). EADRO: an End-to-End troubleshooting framework for microservices on multi-source data. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2302.05092

[5] Akmeemana, L., Attanayake, C., Faiz, H., & Wickramanayake, S. (2025). GAL-MAD: Towards Explainable Anomaly Detection in Microservice Applications Using Graph

Attention Networks. *arXiv preprint arXiv:2504.00058*. https://arxiv.org/abs/2504.00058

[6] Kohyarnejadfard, I., Aloise, D., Azhari, S. V., & Dagenais, M. R. (2022). Anomaly detection in microservice environments using distributed tracing data analysis and NLP. *Journal of Cloud Computing Advances Systems and Applications*, *11*(1). https://doi.org/10.1186/s13677-022-00296-4

[7] Soldani, J., & Brogi, A. (2021). Anomaly Detection and Failure root Cause analysis in (Micro)Service-Based Cloud Applications: A survey. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2105.12378

[8] Ramamoorthi, V. (2024). Anomaly Detection and Automated Mitigation for Microservices Security with AI. *Applied Research in Artificial Intelligence and Cloud Computing*, *7*(6), 211–222. Retrieved from https://researchberg.com/index.php/araic/article/view/216

[9] Kurkute, M. V., Parida, P. R., & Kondaveeti, D. (2024, January 2). *Automating IT service management in manufacturing: A deep learning approach to predict incident resolution time and optimize workflow*. https://jairajournal.org/index.php/publication/article/view/2

[10] Singh, N. S. (2025). Decentralized security Mechanisms for AI-Driven wireless networks: integrating blockchain and federated learning. *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, *11*(2), 1693–1703. https://doi.org/10.32628/cseit25112537