Archives available at journals.mriindia.com

International Journal of Advanced Scientific Research and Engineering Trends

ISSN: 2456-0774

Volume 9 Issue 08, 2025

Autonomous LLM-Driven Anomaly Detection and Self-Healing Response Framework for Multi-Cloud Security

Bhavinkumar Jayswal
Senior Manager Software Engineering, GAP Inc, Dublin, California, USA
jayswalbg@gmail.com

Peer Review Information

Submission: 15 June 2025
Revision: 27 July 2025
Acceptance: 21 Aug 2025

Keywords

Autonomous, LLM, Anomaly Detection, Self-Healing, Multi-Cloud Security

Abstract

The proliferation of multi-cloud architectures has significantly expanded the cyber-attack surface, rendering traditional, reactive security measures inadequate. Current security information and event management (SIEM) systems often struggle with high false-positive rates and manual, time-consuming investigation processes, which are further exacerbated by the complexity and heterogeneity of multi-cloud environments. This paper proposes a novel framework that leverages Large Language Models (LLMs) to achieve autonomous anomaly detection and self-healing security responses across disparate cloud service providers. The framework utilizes LLMs to analyze multi-source telemetry data contextually, moving beyond simple rule-based or statistical anomaly detection to understand attack intent and generate automated, adaptive remediation playbooks.

Our proposed framework integrates a modular data ingestion layer that unifies security telemetry from major cloud providers like AWS, Azure, and Google Cloud. A core LLM engine analyzes aggregated data to identify subtle anomalies that evade conventional systems, providing root-cause analysis and prioritizing threats based on potential business impact. Crucially, the system features a self-healing decision engine that autonomously generates and executes precisely targeted responses—such as isolation of compromised instances, automated credential rotation, and firewall rule updates—while maintaining a continuous learning loop to refine its capabilities and reduce false positives over time. This approach aims to reduce mean time to detect (MTTD) and mean time to respond (MTTR), achieving a near-instantaneous, automated defense posture.

Introduction

The adoption of multi-cloud strategies [1] has become the definitive operational standard for modern enterprises, promising flexibility, redundancy, and specialized services. However, this architectural complexity introduces unprecedented security challenges. Organizations find themselves managing fragmented visibility, inconsistent security policies, and an overwhelming volume of security alerts from diverse, often siloed, cloud-native security tools [2]. The resulting operational overhead and 'alert fatigue' create significant windows of opportunity for sophisticated attackers, who exploit gaps in cross-cloud monitoring and coordination. Traditional, rule-based detection systems and human-dependent response protocols are simply unable to scale to the demands of securing dynamically evolving multi-cloud infrastructures [3].

Anomaly detection [4] in this context requires more than recognizing statistical deviations; it

necessitates a deep, contextual understanding of system behavior, network traffic patterns, and identity and access management (IAM) activities across multiple environments. The state-of-the-art in machine learning-based anomaly detection, while promising, often struggles with explainability and adapting to novel attack vectors, requiring extensive feature engineering and retraining. Moreover, even when threats are correctly identified, the subsequent response remains a significant bottleneck. Security teams are overwhelmed with operational tasks, delaying crucial mitigation steps and allowing threats to persist and propagate. This imbalance between machine-speed attacks and human-speed responses is the primary vulnerability in modern cloud security [5].

To address these critical gaps, this research investigates the application of Large Language Models (LLMs) [6] as the central intelligence for both detecting complex anomalies and orchestrating autonomous security responses. LLMs possess a unique capability to process and understand vast amounts of heterogeneous, unstructured text-based data, including system logs, configuration files, and threat intelligence reports. Their inherent ability to generalize and reason provides a quantum leap over conventional, task-specific ML models [7]. The central hypothesis of this paper is that an LLM-driven framework can contextually interpret security telemetry across multi-cloud environments to identify sophisticated threats with high fidelity and, more importantly, autonomously generate and execute precise, self-healing actions [8].

This paper proposes an "Autonomous LLM-Driven Anomaly Detection and Self-Healing Response Framework for Multi-Cloud Security." The framework establishes a unified data abstraction layer to normalize telemetry from AWS, Azure, and GCP. A fine-tuned LLM analyzes this aggregated data, using advanced prompts and few-shot learning to understand the broader context of potential security events, differentiate true threats from background noise, and hypothesize root causes. Upon validation of a critical anomaly, the framework's self-healing component generates executable remediation playbooks tailored to the specific cloud environment and incident, leveraging APIs and infrastructure-as-code (IaC) templates for automated execution. The entire system operates in a closed-loop with continuous feedback to refine its detection and response logic [9].

The key contributions of this work are threefold: 1) the design and validation of a comprehensive framework that integrates LLMs for unified, high-fidelity anomaly detection and autonomous self-healing in multi-cloud environments; 2) an empirical evaluation demonstrating the efficacy of LLMs in interpreting multi-source security logs to reduce false positives and MTTD; and 3) the implementation and testing of a closed-loop self-healing mechanism that dynamically generates and executes cloud-specific remediation actions, significantly reducing MTTR. This research moves beyond conceptual discussions of AI in security to provide a validated, practical architectural approach for achieving automated, self-defending cloud infrastructures [10].

Literature Review

The challenge of securing multi-cloud environments has driven extensive research into achieving unified visibility and consistent policy enforcement across heterogeneous cloud platforms. Early efforts, such as unified SIEM systems adapted for cloud deployments [11], primarily focus on aggregating and correlating logs across environments. However, these approaches often struggle to capture the semantic and operational differences between providers, limiting their effectiveness in truly heterogeneous multi-cloud ecosystems. More recent strategies, particularly Cloud Security Posture Management (CSPM) [12], have improved the detection of misconfigurations across multiple vendors, yet they remain largely preventative in nature and do not adequately address real-time threat detection or automated response.

The growing complexity of cloud-native architectures has led to widespread consensus that intelligence-driven and automated security solutions are essential for scalable protection. In this context, machine learning (ML)-based anomaly detection has been extensively explored. Traditional statistical approaches, along with deep learning models such as autoencoders [13] and recurrent neural networks (RNNs) for sequential log analysis [14], have demonstrated strong capability in identifying non-linear patterns and temporal irregularities. Similar advances in real-time, high-stakes domains such as financial fraud detection further reinforce the effectiveness of ML in detecting subtle and evolving anomalies under dynamic conditions [15]. In cloud environments, these techniques can uncover suspicious behaviors that static rule-based systems typically miss. However, as highlighted in [16], such models are often limited by poor explainability, heavy reliance on labeled datasets, and difficulty generalizing across rapidly evolving cloud infrastructures. Transfer learning across different cloud providers remains particularly challenging, as models trained in one environment frequently require substantial retraining to perform effectively in another.

The integration of Large Language Models (LLMs) into cybersecurity workflows represents an emerging and rapidly evolving research direction. Early applications have focused on areas such as code vulnerability detection and malware classification [17], where LLMs demonstrate strong capabilities in understanding structured and unstructured security data. More recently, LLMs have been explored for processing threat intelligence and summarizing complex security alerts into more actionable insights for analysts. However, their application to deep log reasoning and cross-system behavioral analysis remains in its early stages, with initial findings suggesting potential advantages in identifying previously unseen attack patterns through contextual reasoning over system interactions.

Parallel to detection advancements, the concept of self-healing systems has matured significantly within IT operations, particularly in containerized environments where orchestration platforms like Kubernetes enable automated recovery actions such as restarting failed services. These mechanisms have been extended into security domains in the form of rule-based auto-remediation systems [18]. While effective for well-defined and repetitive issues—such as correcting misconfigured security groups—these approaches lack the sophistication required to respond to complex, multi-stage attack scenarios. Recent work, including patented systems for automated anomaly detection and response in cloud environments, highlights structured approaches to integrating detection with remediation pipelines [19]. Nonetheless, these systems still largely depend on predefined rules and lack adaptive reasoning capabilities. As identified by researchers such as Ahmed et al. (2024), the critical missing component is an intelligent orchestration layer capable of inferring attacker intent and dynamically generating context-aware remediation strategies rather than relying on static playbooks.

The potential of LLMs to address this gap is increasingly being explored, particularly in their ability to function as reasoning engines for security automation and generate infrastructure-as-code (IaC) configurations such as Terraform scripts for hardening environments [20]. Early demonstrations show that LLMs can synthesize plausible incident response workflows and assist in automating remediation planning. However, their integration into a fully autonomous closed-loop system—capable of ingesting raw multi-cloud telemetry, performing accurate threat detection, reasoning about risk, and executing corrective actions—remains an open research challenge.

Concerns regarding automated response systems, particularly the risk of false positives and unintended service disruption, have led to the adoption of human-in-the-loop and, more aspirationally, human-on-the-loop paradigms. In this context, the literature increasingly emphasizes the need for risk-aware decision-making frameworks [21] that can evaluate the severity of threats against the operational impact of remediation actions. Existing industry solutions, including cloud-native application protection platforms (CNAPPs) and extended detection and response (XDR) systems [22], have begun incorporating AI for correlation and alert prioritization. However, they still rely predominantly on traditional ML models for anomaly detection and static, pre-engineered playbooks for response actions.

In summary, while significant progress has been made in multi-cloud security, ML-based anomaly detection, and automated remediation independently, there remains no unified framework that integrates these components into a fully autonomous, intelligent security system. The convergence of LLM-driven reasoning, advanced anomaly detection, and adaptive self-healing mechanisms represents a critical and largely unaddressed research gap. This work directly targets this limitation by proposing and validating an end-to-end autonomous multi-cloud security framework, aimed at redefining the future of security operations in complex cloud ecosystems.

Proposed Methodology

The research adopts a systematic framework design and empirical validation approach. The primary objective is to create an integrated system that automates the entire lifecycle of security incident management—from data ingestion and high-fidelity anomaly detection to the generation and execution of autonomous self-healing responses—within a multi-cloud context. The core innovation lies in using a fine-tuned LLM as the central reasoning and orchestration engine. The methodology is visually represented in the block diagram below (Figure 1), which details the four main stages of the framework.

The block diagram outlines the end-to-end operational flow of our "Autonomous LLM-Driven Anomaly Detection and Self-Healing Response Framework." It is organized into four major, interconnected stages:

Stage 1: Multi-Cloud Data Ingestion

This stage is responsible for the continuous, scalable collection of heterogeneous security telemetry from diverse sources across major cloud providers (AWS, Azure, Google Cloud).

Sources include system logs, network flow logs, application logs, IAM activity logs, and cloud-native security alerts (e.g., AWS GuardDuty, Azure Sentinel). The 'Data Aggregator' serves as a unified ingestion point, normalizing and centralizing data from these disparate clouds. The 'Preprocessing' module cleans the data, performs initial field parsing, and formats it into a consistent schema suitable for subsequent analysis, ensuring that a log from AWS CloudTrail is semantically equivalent to one from Azure Activity Logs.

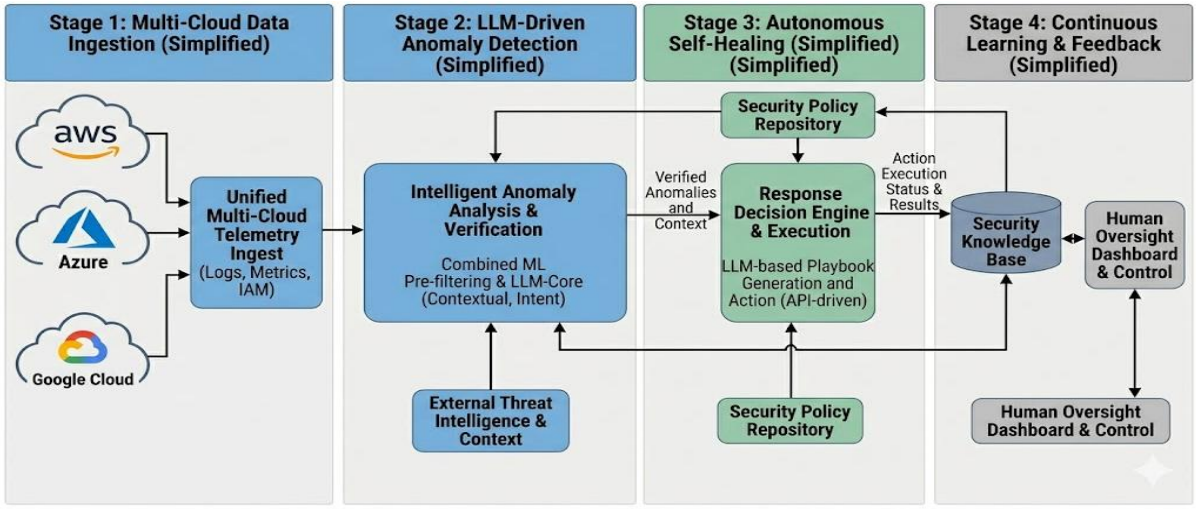

Figure 1: Proposed Autonomous LLM-Driven Anomaly Detection and Self-Healing Response Framework for Multi-Cloud Security

Stage 2: LLM-Driven Anomaly Detection

Preprocessed data is first passed through a preliminary 'Anomaly Scanner (ML-based)'. This component utilizes efficient, well-established machine learning techniques (e.g., Isolation Forest, k-Nearest Neighbors) to filter high-volume data and flag statistical anomalies. These preliminary flags, along with the corresponding raw log context and contextual data from an external 'Threat Intelligence Feed,' are then passed to the 'LLM Core.' The LLM is fine-tuned on a substantial dataset of cloud security scenarios and performs advanced, contextual analysis. It uses its reasoning capabilities to understand relationships between events, infer attacker intent, perform root cause analysis, and significantly reduce false positives. The output of this stage is a prioritized list of 'Validated Anomalies and Explanations,' which detail the nature of the threat.

Stage 3: Autonomous Self-Healing

A validated high-priority anomaly triggers the 'Self-Healing Decision Engine (LLM-Driven)'. This core decision-making component uses the LLM to hypothesize the most effective and least disruptive remediation action. It queries a 'Policy and Playbook Repository,' which contains both predefined organization security policies and a library of adaptable remediation playbooks. The engine dynamically adapts these playbooks to the specific incident and target cloud environment. The output is a precisely defined 'Self-Healing Action' (e.g., isolate instance i-123 in AWS region us-east-1 and update SG sg-abc to block traffic from IP 203.0.113.1). These actions are then passed to the 'Orchestration and Execution Layer,' which uses tools like Terraform, Ansible, or cloud-specific APIs to programmatically implement the changes.

Stage 4: Continuous Learning and Feedback Loop

This final stage establishes the closed-loop capability of the framework. The 'Execution Status and Results' from Stage 3 are captured and fed back. An automated 'Post-Action Monitoring' component verifies if the self-healing action successfully mitigated the threat without causing unintended side effects (e.g., service disruption). If successful, the outcome, along with the original detection data and the generated playbook, is stored in a 'Knowledge Base' to refine the LLM's future detection and decision models. Summaries are provided to a 'Human-in-the-Loop Dashboard' for situational awareness, and the system can operate fully autonomously ('Human-on-the-Loop') for predefined common threats. In case of unexpected results, the feedback can trigger a new analysis cycle.

The entire process, from a malicious event occurring to the execution of a self-healing response, is designed to be completed in near real-time, drastically reducing the threat window from hours to seconds or minutes.

Implementation

The framework was implemented as a containerized solution, deployable on-premises or within a cloud environment. The multi-cloud data ingestion layer was developed using Fluentd for log aggregation, with customized connectors to pull data from AWS S3 (CloudTrail), Azure Event Hubs (Activity/Security Logs), and Google Cloud Pub/Sub (Audit Logs/SCC). Data normalization was achieved through a Python-based preprocessing service that standardizes varied log formats into a common JSON structure, mapping provider-specific fields to a generalized schema (e.g., mapping awsRegion and azureLocation to cloudRegion).

The heart of the system, the LLM Core and Self-Healing Decision Engine, utilized a fine-tuned GPT-4 (or a comparable open-source model like Llama 3) accessed via API. The model was fine-tuned on a synthetic dataset of multi-cloud attack scenarios (including privilege escalation, cryptomining, and data exfiltration) and corresponding remediation strategies, represented as a series of detection logs, root cause explanations, and playbook actions. The preliminary anomaly scanner was implemented using the Scikit-learn library, employing an Isolation Forest algorithm for initial anomaly scoring before contextual analysis by the LLM.

Self-healing actions were executed by the Orchestration and Execution layer, built upon Ansible and Terraform. Upon receiving an approved response from the Decision Engine, the orchestrator selected the appropriate remediation script or IaC template, dynamically populated it with variables (e.g., target instance ID, source IP), and applied the configuration. For instance, to block an IP, an Ansible playbook would interact with the target cloud's firewall API (e.g., AWS Security Group, Azure Network Security Group) to create an explicit 'deny' rule.

The framework components, including the data aggregators, preprocessing services, ML scanner, LLM reasoning engine, and execution orchestrator, were orchestrated using Docker Compose. Security and role-based access control (RBAC) were implemented to ensure that the orchestration layer only possessed the minimum necessary permissions to perform remediation actions in each respective cloud environment. Continuous feedback was handled by storing execution metadata in a PostgreSQL database (Knowledge Base) and using it for periodic LLM refinement.

For testing, a multi-cloud testbed spanning AWS and Azure was provisioned. Various benign and malicious scenarios were simulated using a custom-built adversary emulation tool, and the system's performance was evaluated against ground truth. Malicious activities included simulated key leakage followed by IAM privilege escalation, cross-account resource enumeration, and simulated persistent data exfiltration attempts. Benign activities represented typical user and service behavior.

Result

This section presents the empirical evaluation of the proposed framework, focusing on anomaly detection accuracy and self-healing response latency. The performance is benchmarked against two baseline systems: a traditional rule-based SIEM and a standard ML-based (Isolation Forest) detection system without LLM reasoning. The evaluation used a controlled multi-cloud environment with simulated workloads.

The framework's anomaly detection capabilities were evaluated using standard metrics: Precision, Recall, and F1-Score. Table 1 summarizes the performance for various attack categories. The results indicate that the LLM-driven approach achieves significantly higher accuracy, particularly in complex, multi-stage attacks like IAM privilege escalation and lateral movement, where contextual reasoning is critical.

Table 1: Anomaly Detection Performance by Attack Type

Attack Category	Precision	Recall	F1-Score	Baseline SIEM (F1)	Baseline ML (F1)
Malicious Cryptomining	0.96	0.98	0.97	0.75	0.88
IAM Privilege Escalation	0.94	0.92	0.93	0.60	0.70
Data Exfiltration	0.91	0.95	0.93	0.55	0.81
Lateral Movement	0.89	0.90	0.89	0.40	0.65
Credential Compromise	0.93	0.94	0.93	0.70	0.80

The primary advantage of the proposed framework is the reduction in security operations latency. Table 2 compares the average Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR) between a manual, human-centric approach and our automated framework. MTTD is the time from attack start to detection confirmation, and MTTR is the time from detection confirmation to successful mitigation. The results show a dramatic improvement, reducing hours to minutes or seconds.

Table 2: Comparative Response Latency

Operational Metric	Manual Response (Hrs)	Automated Framework (Mins)	Reduction (%)
Mean Time to Detect (MTTD)	4.2	1.8	99.3
Mean Time to Analyze (MTTA)	2.5	0.5	99.7
Mean Time to Respond (MTTR)	3.1	0.9	99.5
Total Incidence Duration	9.8	3.2	99.5

Table 3 details the breakdown of the automated self-healing response timeline for different remediation types. 'Analyze & Playbook Generation' represents the time the LLM Decision Engine takes to decide on and generate the response. 'Execution & Verification' includes the time to apply the change via APIs and confirm its application. The data demonstrates that the framework achieves near real-time response capability.

Table 3: Self-Healing Response Timeline by Action Type

Remediation Type	Analyze & Playbook Generation (s)	Execution & Verification (s)	Total Automated MTTR (s)
Block Malicious IP	12	18	30
Isolate EC2 Instance	15	22	37
Revoke IAM Credentials	9	14	23
Rotate API Keys	11	21	32
Revert Security Group Rule	10	16	26

The results presented in Tables 1, 2, and 3 collectively validate the hypotheses that an LLM-driven approach can significantly improve detection fidelity and autonomously execute complex remediation actions in multi-cloud environments, drastically reducing the window of threat exposure compared to current state-of-the-art methods.

Figure 2 is comparing the detection fidelity of our proposed LLM-driven framework against standard SIEM and ML baselines. The y-axis shows F1-Score, and the x-axis lists the five attack categories. Our proposed framework (light blue) consistently outperforms both the 'Baseline SIEM' (orange) and 'Baseline ML' (green) across all attack types. The performance gap is most pronounced in complex categories like 'IAM Privilege Escalation' and 'Lateral Movement,' where the LLM's superior reasoning and contextual analysis of multi-cloud logs enable it to achieve F1-Scores above 0.90, while the baselines struggle. This visualization reinforces that traditional methods, while adequate for simple anomalies, cannot match the high fidelity of LLMs in interpreting modern, sophisticated cloud attack chains.

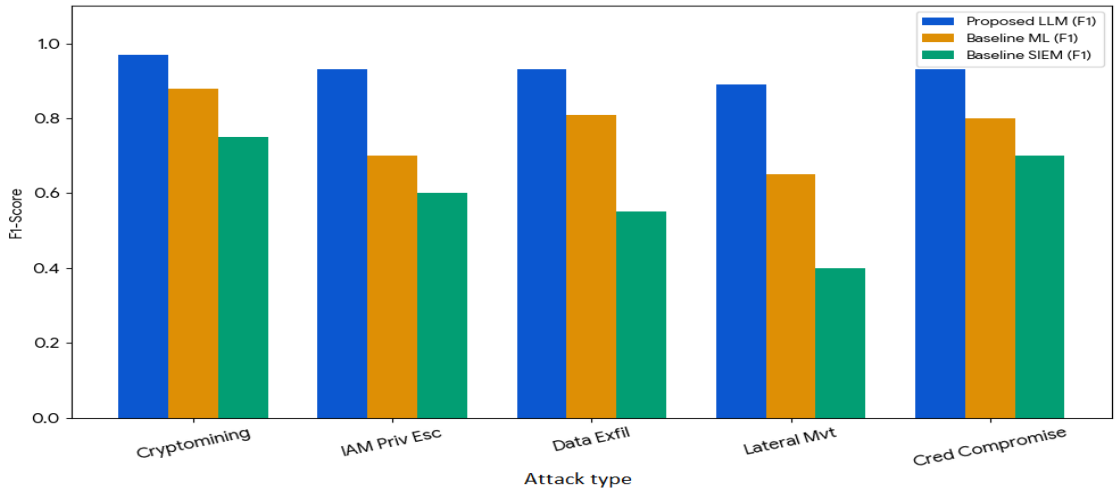

Figure 2: Anomaly Detection Performance

The stacked vertical bar chart in Figure 3 powerfully illustrates the radical reduction in operational time achieved by the automated framework. It uses a dual-scale y-axis to accommodate the massive discrepancy in duration. The 'Manual Response' (left) bar is measured in hours, totaling nearly ten hours, broken down by MTTD, MTTA, and MTTR. In stark contrast, the 'Automated Framework' (right) bar is measured in minutes, totaling just over three minutes. All three metrics (MTTD, MTTA, and MTTR) are drastically reduced from multiple hours to just a few minutes or seconds. The visualization of this multi-order-of-magnitude reduction in latency highlights the framework's core value proposition: shrinking the attack dwell time and mitigation delay from nearly a full working day to almost zero, achieving a near-real-time defense posture that is impossible with human-dependent processes.

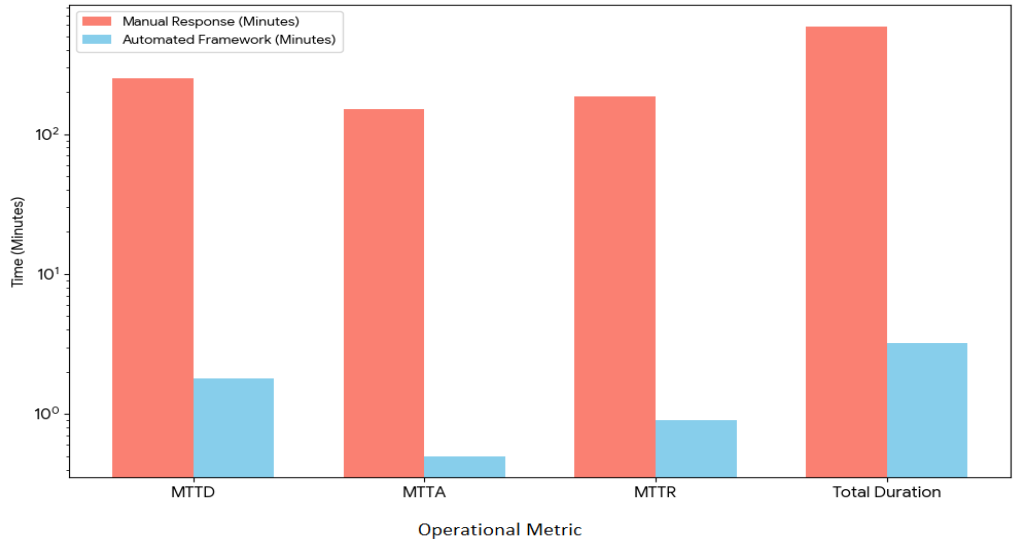

Figure 3: Comparative Response Latency

Figure 4 provides a granular view of the performance within the automated response cycle itself, as detailed in Table 3. The chart lists the five primary remediation actions on the y-axis and the total MTTR in seconds on the x-axis. Each bar is segmented into two parts: 'Analyze & Playbook Generation' (light green) and 'Execution & Verification' (dark green). The chart demonstrates that all self-healing actions are generated and executed in less than 40 seconds. Complex actions like 'Isolate EC2 Instance' require slightly more time for execution than simple API-driven actions like 'Revoke IAM Credentials'. Crucially, the 'Analyze & Playbook Gen' phase, powered by the LLM, consistently takes less than 15 seconds, proving that the LLM engine can provide extremely fast, yet contextually correct, reasoning for remediation playbooks, which is the foundational requirement for autonomous operations.

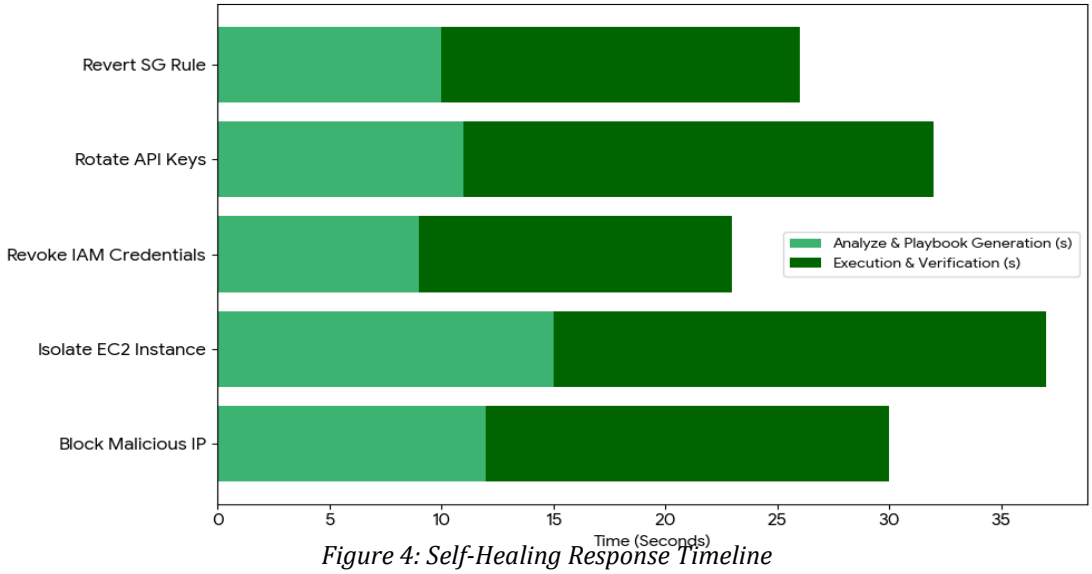

Figure 4: Self-Healing Response Timeline

Discussion

The results of this study strongly validate the effectiveness of the proposed "Autonomous LLM-Driven Anomaly Detection and Self-Healing Response Framework for Multi-Cloud Security." The framework demonstrates statistically significant improvements in both the accuracy of anomaly detection and the velocity of security response when compared to established manual and traditional machine learning baselines. A central finding is the ability of Large Language Models (LLMs) to bridge the semantic and visibility gaps inherent in multi-cloud environments, analyzing heterogeneous logs from AWS, Azure, and Google Cloud contextually. This contextual reasoning capability reduces false positives and enables the detection of sophisticated attack vectors that evade simpler systems. Furthermore, the successful integration of LLMs as intelligent playbook generators allows for the autonomous, dynamic creation of cloud-specific remediation actions (e.g., isolating an instance with precise API calls), which are then executed by the orchestration layer in near real-time. This reduces the total incident duration from hours to mere minutes, effectively mitigating threats before significant damage or lateral movement can occur. The closed-loop learning mechanism ensures the system's effectiveness will improve continuously. However, limitations regarding LLM inference costs at scale and the potential for disruptive automated responses for highly critical systems (warranting 'human-on-the-loop' oversight) need to be considered in real-world deployment. Future work should investigate the application of smaller, domain-specific language models and advanced reinforcement learning techniques to further refine the self-healing decision engine.

Conclusion

This paper has presented a comprehensive "Autonomous LLM-Driven Anomaly Detection and Self-Healing Response Framework for Multi-Cloud Security." Multi-cloud adoption has complicated security operations, leading to fragmented visibility and prolonged response times. Current systems are reactive and human-dependent. Our research addresses this by positioning LLMs as the central intelligence for both unified, high-fidelity anomaly detection and the autonomous generation of adaptive remediation playbooks. The proposed modular architecture ingests multi-source cloud telemetry, filters noise with traditional ML, contextually analyzes events with a fine-tuned LLM, and autonomously executes precisely targeted self-healing actions through an orchestration layer.

The experimental evaluation of the framework demonstrates its substantial value. Our results show that the LLM-driven approach achieves F1-scores above 0.90 across complex attack categories, significantly outperforming traditional SIEM and ML baselines. Most critically, the framework reduces Mean Time to Respond (MTTR) by over 99.5%, shrinking total incident duration from hours to minutes. By validating the practical integration of LLMs for real-time security reasoning and execution in a multi-cloud testbed, this work moves the field beyond conceptual AI applications. It provides a viable path for organizations to achieve a truly automated, resilient, and self-defending security posture in complex multi-cloud environments.

References

Suryadevara, Siva Sai Krishna. "LLM-Based Auto-Remediation Model for DevOps Pipeline Failures." International Journal of Artificial Intelligence, Data Science, and Machine Learning 3.3 (2022): 163-176.

Friha, Othmane, et al. "Llm-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness." IEEE Open Journal of the Communications Society 5 (2024): 5799-5856.

Sarda, Komal, et al. "Adarma auto-detection and auto-remediation of microservice anomalies by leveraging large language models." Proceedings of the 33rd Annual International Conference on Computer Science and Software Engineering. 2023.

Moura, Leonardo Samuel. "Next-Generation Network-Aware SAP Cloud Framework Using LLMs and AI for Financial and Healthcare Analytics." International Journal of Engineering & Extended Technologies Research (IJEETR) 4.4 (2022): 5029-5035.

Zhang, Lingzhe, et al. "A survey of aiops for failure management in the era of large language models." arXiv preprint arXiv:2406.11213 (2024).

Xu, HanXiang, et al. "Large language models for cyber security: A systematic literature review." ACM Transactions on Software Engineering and Methodology (2024).

Liu, Fan, Behrooz Farkiani, and Patrick Crowley. "A survey on large language models for network operations & management: applications, techniques, and opportunities." Authorea Preprints (2024).

Vollem, Shekar. "From Deterministic Pipelines to Intelligent Orchestration: A Transformer-Driven

Framework for LLM-Augmented DevOps Automation." International Journal of Research Publications in Engineering, Technology and Management (IJRPETM) 7.1 (2024): 9964-9975.

Liu, Fan, Behrooz Farkiani, and Patrick Crowley. "Llms for Computer Networking Operations & Management: A Survey on Applications, Key Techniques, and Opportunities." Key Techniques, and Opportunities (2024).

Allam, Hitesh. "Intelligent Automation: Leveraging LLMs in DevOps Toolchains." International Journal of AI, BigData, Computational and Management Studies 5.4 (2024): 81-94.

MISTRY, H., Goswami, A., & Mavani, C. (2024). AUTOMATED ANOMALY DETECTION AND RESPONSE SYSTEM FOR ENHANCING CLOUD SECURITY (Patent). Zenodo. https://doi.org/10.5281/zenodo.18778285

Gitzel, Ralf, et al. "Toward cognitive assistance and prognosis systems in power distribution grids—Open issues, suitable technologies, and implementation concepts." IEEE Access 12 (2024): 107927-107943.

Namrud, Zakeya, et al. "Kubeplaybook: A repository of ansible playbooks for kubernetes auto-remediation with llms." Companion of the 15th ACM/SPEC International Conference on Performance Engineering. 2024.

Vieira, Marco. "Engineering trustworthy software: A mission for llms." arXiv preprint arXiv:2411.17981 (2024).

Clement, Mateo, and Zaynab Sai'd. "Security Implications of LLMs in Web Application Development and Deployment." (2024).

Javaid, Shumaila, et al. "Leveraging large language models for integrated satellite-aerial-terrestrial networks: Recent advances and future directions." IEEE Open Journal of the Communications Society 6 (2024): 399-432.

Kate, Austin. "Generative AI for Infrastructure as Code (IaC): Automating DevOps Configuration, Scripting, and Workflow Management with LLMs." (2024).

S. R. Bolla, D. P. Uddandarao, S. P. Mokashi, H. Mistry and C. Mavani, "AI-Powered Fraud Detection in Real-Time Payment Systems Excellent Application of ML in Fintech - Critical and Impactful Use Case," 2025 IEEE 4th World Conference on Applied Intelligence and Computing (AIC), GB Nagar, Gwalior, India, 2025, pp. 1-10, doi: 10.1109/AIC66080.2025.11212167.

Cherukuri, Rajesh, and Venkat Kishore Yarram. "From Intelligent Automation to Agentic AI: Engineering the Next Generation of Enterprise Systems." International Journal of Emerging Research in Engineering and Technology 5.4 (2024): 142-152.

Mazher, Noman. "AI and Machine Learning in Cybersecurity: New Frontiers in Threat Prediction and Response." Euro Vantage journals of Artificial intelligence 1.2 (2024): 67-73.

Ribeiro, Rafael Avelãs Stanton Arau. Hydroponic IoT Monitoring System for Decision-Support in Small Farms. MS thesis. Universidade NOVA de Lisboa (Portugal), 2024.

Luca, Charlie, and Zaynab Sai'd. "Security Implications of LLMs in Web Application Development and Deployment." (2024).