Proactive Security in Multi-Cloud Environments: A Blockchain-Integrated Real-Time Anomaly Detection and Mitigation Framework

[1*]J Maha Lakshmi, [2] Krishna Prasad K ,[3] Viswanath G

*[*1]Post-Doctoral Fellow, Srinivas University, Mangaluru, Karnataka, Associate Professor, Information Technology, MLR institute of Technology, Dundigal, Hyderabad, Telangana,India*

[2]Professor, Institute of Engineering and Technology, Srinivas University, Mukka-574146, Karnataka, India, ORCID-ID: 0000-0001-5282-9038;

[3]Associate Professor, Sri Venkatesa Perumal College of Engineering and Technology, Puttur, India, ORCID-ID: 0009-0001-7822-4739;

*[1]*Corresponding author: Email Id: mahalakshmi1203@gmail.com*

Abstract : AegisHaven is a proactive security framework designed to safeguard Virtual Machine (VM) deployments in dynamic multi-cloud environments. Motivated by the challenges posed by post-deployment vulnerabilities, evolving threats, and the fragmented nature of traditional security approaches, this framework integrates blockchain verification with real-time anomaly detection and automated mitigation mechanisms. The system leverages advanced machine learning techniques, including Variational Autoencoders (VAEs) and Long Short-Term Memory (LSTM) networks, to monitor runtime behavior and detect anomalies, such as zero-day attacks, with high accuracy, precision, and recall. Upon detection, AegisHaven employs automated mitigation strategies, including isolation, rollback, and self-healing, to ensure rapid response and recovery, minimizing downtime and operational disruptions. Blockchain technology enhances the framework by providing immutable logs, ensuring auditability, regulatory compliance, and tamper-proof evidence for forensic analysis. The evaluation demonstrates AegisHaven's scalability, with the ability to process up to 1000 entries per second, while maintaining low CPU utilization (55%) and reduced memory overhead (1536 MB). Compared to traditional frameworks like BCALS and Logchain, AegisHaven consistently outperforms in anomaly detection accuracy (96.5%) and mitigation success rates (98.5%), validating its robustness and efficiency. Future enhancements will focus on optimizing machine learning models for faster detection, exploring lightweight blockchain technologies to minimize overhead, and expanding compatibility with additional cloud providers. These improvements position AegisHaven as a scalable, adaptable, and efficient security solution for modern multi-cloud infrastructures.

Keywords: Anomaly Detection, Blockchain Security, Multi-Cloud Environments, Machine Learning, Automated Mitigation, Scalability.

1. Introduction

The advent of cloud computing has revolutionized the delivery and scalability of digital services, with Virtual Machine Images (VMIs)[1] playing a central role in enabling rapid deployment and efficient resource utilization. These VMIs encapsulate operating systems, software configurations, and application states, allowing seamless provisioning of services across diverse cloud platforms. In multi-cloud environments where organizations leverage multiple cloud providers for redundancy, flexibility, and compliance, the strategic deployment of VMIs is critical to achieving operational efficiency. However, while much attention has been devoted to securing VMIs before deployment, the post-deployment phase remains a significant vulnerability, exposing cloud infrastructures to sophisticated and evolving threats[2]. Once deployed, VMIs operate in distributed and dynamic environments that are characterized by interconnected cloud services and varying operational demands. These post-deployment dynamics

[3] introduce a spectrum of challenges that traditional security frameworks are ill-equipped to address. The runtime behavior of virtual machines is influenced by fluctuating workloads, user interactions, and evolving cyber threats, such as zero-day vulnerabilities[4], unauthorized access attempts, and resource hijacking (e.g., cryptojacking). These threats exploit the inherent complexities and dynamism of multi-cloud systems, where a lack of cohesive oversight can lead to significant security gaps[5].

The distributed architecture of multi-cloud environments further exacerbates these challenges. Organizations often deploy VMIs across multiple cloud providers, each with distinct configurations, policies, and security tools. This heterogeneity complicates the implementation of unified monitoring and response mechanisms, resulting in fragmented security oversight. Transitioning VMIs between cloud platforms, a common practice in multi-cloud operations, creates additional vulnerabilities by exposing sensitive data and configurations to varied and potentially less secure environments. Furthermore, the sheer scale and variability of multi-cloud systems make it difficult to maintain up-to-date security measures. The rapid state changes in deployed VMs, driven by the dynamic provisioning and de-provisioning of resources, can outpace the capabilities of traditional security policies, leaving organizations exposed to new and unexpected threats. In this context, ensuring continuous protection for VMIs becomes an increasingly complex and critical task.

Traditional approaches to securing VMIs during their post-deployment phase predominantly rely on reactive strategies, addressing vulnerabilities only after they have been exploited. These methods often involve static rule-based systems that are designed to detect pre-identified patterns of anomalous behavior. While effective against known threats, such systems are inherently limited in their ability to identify novel or sophisticated attacks, such as advanced persistent threats (APTs) and zero-day exploits[7]. This limitation arises from their dependence on predefined signatures and thresholds, which cannot adapt to the ever-evolving tactics of cyber adversaries. Moreover, reactive systems frequently suffer from high rates of false positives, generating excessive alerts that overwhelm security teams and delay response times. This inefficiency is particularly problematic in large-scale multi-cloud environments, where the volume of alerts can become unmanageable. The resulting delays in addressing genuine threats increase the risk of data breaches, service disruptions, and reputational damage.

The fragmented nature of security tools across different cloud platforms further compounds these issues [8]. Each provider offers proprietary solutions for monitoring and threat detection, leading to a lack of interoperability and standardization[8]. This fragmentation forces organizations to adopt a siloed approach to security management, making it difficult to implement cohesive anomaly detection and response strategies across multi-cloud deployments. Consequently, security gaps often emerge during transitions between platforms, providing opportunities for malicious exploitation. Another significant limitation of existing solutions is their reliance on manual interventions to address detected anomalies [9]. Security teams are tasked with interpreting alerts, diagnosing issues, and deploying mitigation strategies processes that are inherently time-consuming and prone to human error. In dynamic and distributed multi-cloud environments, where the window for effective response is often narrow, such delays can have catastrophic consequences. Finally, the absence of robust mechanisms for transparency and auditability in traditional frameworks undermines their effectiveness. Although logging and reporting are common practices, these logs are often stored in centralized systems that are susceptible to tampering or loss during a security incident. This lack of immutable and trustworthy records not only hinders regulatory compliance but also complicates post-incident forensic investigations, leaving organizations without a clear understanding of the breach's scope or origin [10].

The inherent limitations of reactive security measures underscore the need for a paradigm shift towards proactive and adaptive solutions. By addressing threats dynamically and reducing reliance on manual processes, next-generation frameworks can significantly enhance the resilience and security of VMIs in multi-cloud environments. Real-time anomaly detection and mitigation represent a fundamental departure from traditional reactive security strategies, offering a proactive approach to addressing runtime threats. This paradigm leverages advanced technologies, such as machine learning (ML) and artificial intelligence (AI), to continuously monitor the behavior of deployed VMs, identifying deviations from normal patterns as they occur. Unlike static rule-based systems, real-time detection mechanisms can adapt to evolving threats and operational contexts, enabling the identification of complex and previously unknown attacks. Machine learning models form the backbone of effective real-time anomaly detection

systems. By analyzing metrics such as CPU and memory utilization, network traffic, file access patterns, and system calls, these models can construct behavioral baselines for VMIs. Anomalies are detected when deviations from these baselines exceed acceptable thresholds, allowing the system to identify potential security incidents with high precision. Advanced techniques, such as time-series analysis and clustering algorithms, further enhance the system's ability to differentiate between legitimate variability and malicious activity. Equally critical to real-time security is the ability to respond to detected anomalies dynamically. Automated mitigation strategies, such as isolating compromised VMs, rolling back to verified snapshots, or terminating malicious processes, ensure that threats are neutralized before they can escalate. These automated responses eliminate the delays associated with manual interventions, significantly reducing the potential impact of security incidents. In addition, self-healing mechanisms—where the system automatically restores affected VMs to a known secure state—enhance operational resilience and reduce downtime. The integration of real-time detection and mitigation with blockchain technology provides an additional layer of security and transparency. By logging detected anomalies, response actions, and resulting system states in an immutable ledger, the system ensures that all events are securely recorded and tamper-proof. This transparency facilitates regulatory compliance and supports forensic investigations, providing organizations with the tools to demonstrate accountability and learn from past incidents. In summary, the adoption of real-time anomaly detection and mitigation represents a transformative approach to securing VMIs in multi-cloud environments. By addressing runtime threats dynamically and automating responses, this proactive framework not only mitigates security risks but also enhances operational efficiency, transparency, and user trust. This shift from reactive to proactive security marks a critical step forward in meeting the demands of modern cloud computing infrastructures.

The primary objective of this research is to develop a robust security framework that integrates real-time anomaly detection and automated response mechanisms with blockchain-based Virtual Machine Image (VMI) verification. The proposed system addresses runtime security gaps by proactively identifying and mitigating threats during the post-deployment phase in multi-cloud environments. Leveraging blockchain's immutable integrity and machine learning's dynamic adaptability, this framework aims to establish a comprehensive and resilient security paradigm.

- **Enhancing Post-Deployment Security**: Augmenting the pre-deployment integrity verification of VMIs with continuous runtime monitoring to detect deviations from expected behavior in real time.
- **Automating Threat Mitigation**: Implementing intelligent response mechanisms capable of isolating, neutralizing, and recovering from detected anomalies without manual intervention, thereby reducing response time and minimizing operational impact.
- **Ensuring Transparency and Trust**: Leveraging blockchain's tamper-proof logging capabilities to document anomalies, responses, and state transitions, creating an auditable and trustworthy record of security events.
- **Supporting Multi-Cloud Interoperability**: Developing a system that integrates seamlessly across diverse cloud platforms, enabling consistent security measures regardless of provider-specific configurations.

This objective not only addresses the reactive limitations of traditional security solutions but also pioneers a proactive approach to safeguarding VMIs, ensuring resilience and trustworthiness in increasingly complex cloud ecosystems.

This study addresses the following key research questions to guide the development and evaluation of the proposed framework:

1. How can real-time anomaly detection be effectively integrated with blockchain technology to enhance the security of VMIs in multi-cloud environments?
2. What machine learning techniques are most effective for identifying and classifying anomalies in the dynamic and heterogeneous context of multi-cloud systems?
3. How can automated mitigation strategies be designed to promptly neutralize threats while maintaining operational efficiency?
4. How can cross-platform compatibility be achieved to provide a unified security framework across multiple cloud service providers?

This research makes the following significant contributions to the field of cloud security:

- **A Novel Anomaly Detection System Powered by Machine Learning:** Introduces a real-time anomaly detection system leveraging machine learning to identify emerging threats.
- **Automated, Blockchain-Logged Mitigation Strategies :** Develops a framework for automated threat mitigation with blockchain-based immutable logging.
- **Cross-Platform Compatibility in Multi-Cloud Setups:** Designs a platform-agnostic system ensuring seamless security across diverse cloud providers.

2. Related Work

2.1 **Blockchain in Cloud Security**

Blockchain technology has been increasingly utilized in cloud security to enhance verification, logging, and overall data integrity. Several studies have explored the integration of blockchain with cloud computing to address security challenges[11][12]. Blockchain's distributed ledger provides a tamper-proof system for maintaining cloud data provenance, allowing for the tracking and recording of data object operations while preserving user privacy[13]. One of the key applications of blockchain in cloud security is the implementation of authentication and access control mechanisms. Blockchain and smart contracts offer improved data and rule confidentiality, integrity, and system availability by eliminating single points of failure [14]. Additionally, blockchain-based public key cryptosystems have been proposed to protect data in cloud storage, automatically encrypting data using homomorphic techniques and storing ciphertext indexes in the blockchain [15]. However, there are still gaps in addressing post-deployment threats in blockchain-based cloud security solutions. While blockchain provides robust security features, it also introduces its own vulnerabilities, such as mining attacks and key management issues, which need to be addressed [16]. Furthermore, the emergence of quantum computing poses a significant threat to blockchain security, particularly in the areas of public-key cryptography and hash functions, necessitating the development of quantum-resistant cryptosystems for blockchain architectures [17]. In conclusion, while blockchain has shown promise in enhancing cloud security through improved verification and logging mechanisms, there is a need for further research to address post-deployment threats and emerging challenges such as quantum computing. Future work should focus on developing more resilient blockchain-based security solutions that can withstand both traditional and quantum-based attacks in cloud environments.

2.2 **Real-Time Anomaly Detection in Cloud Systems**

Real-time anomaly detection in cloud systems has become increasingly important with the growth of Industry 4.0 and the Internet of Things. Machine learning models and frameworks play a crucial role in detecting anomalies efficiently and accurately in these complex environments. Several ML models have been proposed for real-time anomaly detection in cloud systems. These include Hierarchical Temporal Memory (HTM) combined with Bayesian Network (BN)[18] ensemble models combining Local Outlier Factor, One-Class Support Vector Machine, and Autoencoder [19], and Principal Component Analysis (PCA), Fully Connected Autoencoder (FC-AE), and Convolutional AutoEncoder (C-AE) [20] . Additionally, Federated Learning-based approaches have been explored to address privacy concerns in smart grid anomaly detection [21]. Key challenges in implementing real-time anomaly detection in cloud systems include resource efficiency, accuracy, and scalability. To address these challenges, researchers have proposed various solutions. For instance, edge computing and fog computing architectures have been utilized to reduce network traffic and cloud resource utilization .The RAMP (Real-Time Aggregated Matrix Profile) system has been developed to detect anomalies in scientific workflow systems in real-time, incorporating user feedback to improve accuracy [22] in conclusion, real-time anomaly detection in cloud systems requires a balance between accuracy, resource efficiency, and scalability. Hybrid approaches combining multiple ML models, edge computing, and privacy-preserving techniques show promise in addressing these challenges. As the field evolves, further research is needed to develop more robust and adaptable solutions for diverse cloud environments.

2.3 **Automated Mitigation in Cloud Security**

Automated mitigation techniques in cloud security have become increasingly important as organizations seek to rapidly respond to threats and vulnerabilities. Existing approaches include isolation of compromised resources and rollback to known-good states [23]. These techniques aim to minimize the impact of security incidents and restore normal operations quickly. However, integrating blockchain and Mobile Cloud Computing (MCS) for automated mitigation presents some challenges. The highly dynamic and distributed nature of mobile ad-hoc networks makes it difficult to implement traditional consensus protocols used in blockchain systems[24]. Additionally, the computational requirements of blockchain consensus mechanisms like proof-of-work can be prohibitive for resource-constrained mobile devices [25]. To address these limitations, researchers have proposed novel approaches. For example, the AdChain cloud architecture uses a stability-aware consensus protocol designed specifically for mobile ad-hoc environments. Other work has been explored using proof-of-stake consensus to reduce computational overhead. Integrating Software Defined Networking (SDN) with blockchain can also improve the durability and load balancing of cloud infrastructure for Industrial IoT applications [26] . While challenges remain, these innovations show promise for enhancing automated mitigation capabilities through blockchain integration in mobile and cloud environments.

2.4 Research Gaps and Challenges

While significant advancements have been made in blockchain, anomaly detection, and automated mitigation, several critical gaps remain:

1. **Post-Deployment Security**: Existing solutions focus on pre-deployment verification but fail to address runtime threats such as zero-day attacks and insider breaches.
2. **Fragmentation**: Current systems lack integration, with blockchain, anomaly detection, and mitigation operating in silos, reducing efficiency and responsiveness.
3. **Scalability**: Security frameworks often struggle to scale across large, distributed, and multi-cloud environments without excessive resources overhead.
4. **Interoperability**: Tools designed for specific cloud providers lack standardization, complicating their deployment in heterogeneous multi-cloud systems.
5. **Automation**: Limited integration between anomaly detection and automated mitigation delays threat responses and reduces effectiveness.
6. **Emerging Threats**: Quantum computing poses risks to blockchain cryptography, necessitating quantum-resistant security mechanisms.

A unified framework integrating blockchain, real-time anomaly detection, and automated mitigation is essential to addressing these gaps, ensuring comprehensive, scalable, and adaptive cloud security.

3. Methodology

3.1 **Proposed Framework Architecture**

The proposed security framework, named "Aegis Haven Framework", embodies the concept of an impenetrable shield (Aegis) and a secure refuge (Haven). This framework addresses critical post-deployment challenges in multi-cloud environments by integrating blockchain technology, real-time anomaly detection, and automated mitigation mechanisms. It ensures robust integrity verification, precise anomaly detection, and seamless recovery for Virtual Machine Images (VMIs), all while maintaining adaptability and scalability across diverse cloud platforms. Following are the core components of the proposed framework i.e Block chain Layer(Blockchain Vault), Anomaly Detection layer and Mitigation Layer.

1. The **Blockchain Vault secures** VMI integrity and authenticity by storing cryptographic hashes and providing an immutable record of security events. Smart contracts automate hash verification processes, while a lightweight proof-

of-stake (PoS) consensus mechanism ensures efficiency in multi-cloud settings. Here the Unique Implementation is performed in following manner i.e Dynamic snapshot verification via smart contracts guarantees runtime integrity during restoration

2. Anomaly Detection Engine Using ML/AI Models : The anomaly detection layer employs advanced machine learning algorithms to identify runtime deviations from normal VMI behavior. It adapts dynamically to evolving threats using both spatial and temporal anomaly detection techniques.

Novel Algorithms:

- **Variational Autoencoders (VAEs)**: Capture latent patterns of normal behavior and detect anomalies through reconstruction errors.
- **Long Short-Term Memory Networks (LSTMs)**: Model temporal dependencies to identify anomalies in time-series data, such as resource usage spikes.
- **Federated Learning with Adaptive Thresholding**: Enable decentralized training while preserving data privacy across cloud nodes.
- **Hybrid Clustering + Reinforcement Learning (RL)**: Combine clustering of high-dimensional data with RL to optimize anomaly thresholds dynamically.

3. **Mitigation Mechanism for Isolation and Recovery :** The mitigation layer automates responses to anomalies, minimizing downtime and manual intervention. It intelligently neutralizes threats and restores operations through a combination of isolation, rollback, termination, and self-healing. Here major implementation is Deep Q-Networks (DQNs)[27] which Optimize decision-making for mitigation actions based on severity and operational impact.

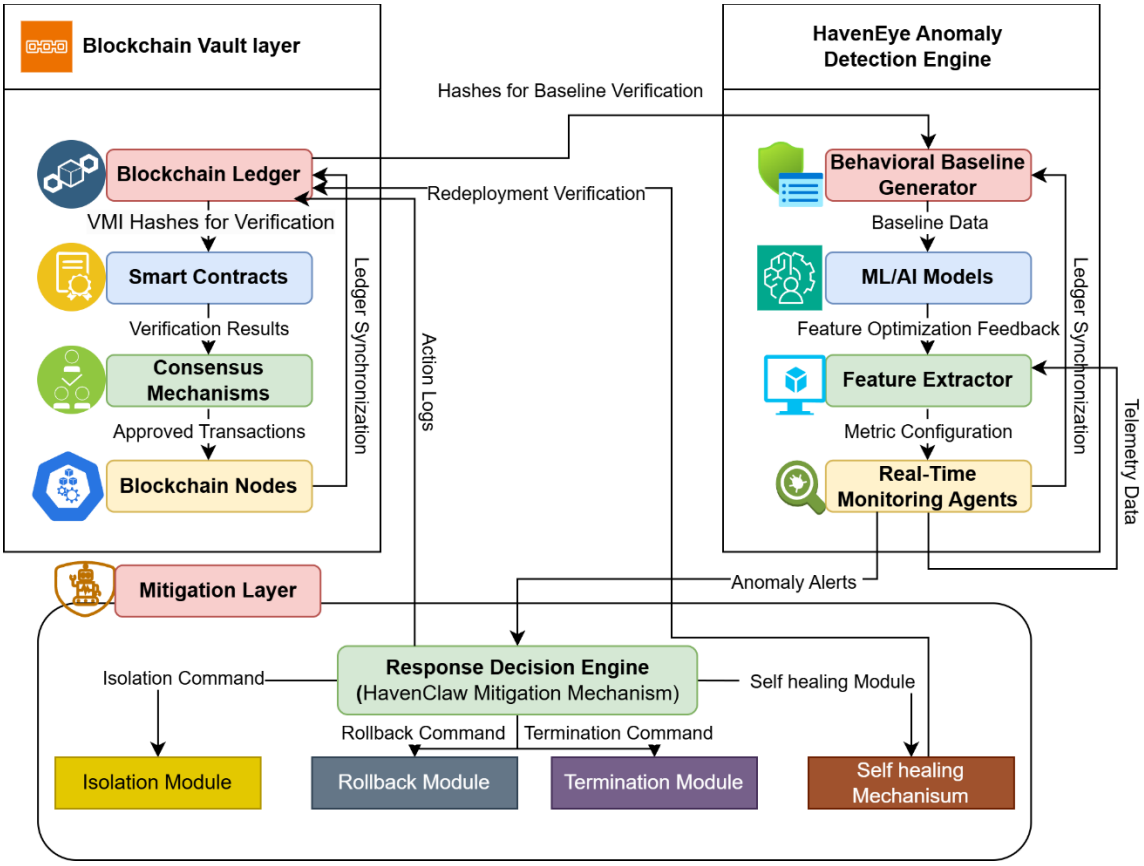

Figure 1. Proposed AegisHaven Framework Architecture

Figure 1 framework integrates a robust multi-layered approach to securing Virtual Machine Images (VMIs) in multi-cloud environments. The **Blockchain Vault Layer** ensures tamper-proof integrity through its Blockchain Ledger, Smart Contracts for automated verification, and distributed Consensus Mechanisms that propagate approved transactions across Blockchain Nodes. The **HavenEye Anomaly Detection Engine** employs Behavioral Baseline Generators, advanced ML/AI models, Feature Extractors, and Real-Time Monitoring Agents to detect and analyze anomalies in runtime telemetry data. The **HavenClaw Mitigation Layer** dynamically responds to threats using its Response Decision Engine, which orchestrates actions such as VM isolation, state rollback, malicious termination, and self-healing mechanisms while maintaining transparency through blockchain-logged action records. This integrated architecture exemplifies a proactive and adaptive security paradigm, leveraging blockchain's immutability, ML/AI's predictive capabilities, and automated mitigation strategies to provide comprehensive protection and recovery in modern cloud infrastructures.

3.2 Real-Time Anomaly Detection System

The real-time anomaly detection system is a critical component of the proposed framework, designed to dynamically identify deviations in Virtual Machine Image (VMI) behavior[28]. It leverages advanced machine learning and artificial intelligence algorithms to detect spatial and temporal anomalies in distributed, multi-cloud environments. The system integrates theoretical underpinnings with mathematical models to enhance detection precision, scalability, and adaptability[29].

Feature Selection: Feature selection is essential for defining the operational characteristics of VMIs that capture both normal and anomalous behaviors. Let $X = \{x_1, x_2, \dots, x_n\}$ represent the set of telemetry data features for a VMI. These features include:

- Resource Utilization: CPU usage (x_{cp_u}), memory consumption (x_{mem}), and disk I/O activity (x_{disk}).
- Network Traffic Metrics: Packet rate (x_{pkt}), bandwidth consumption (x_{bw}), and anomalies in connection behavior (x_{comn}).
- System-Level Operations: Features derived from system call frequencies (x_{sys}), process execution patterns (x_{proc}), and file access behaviors (x_{file}).

The feature vector X is preprocessed and normalized to ensure uniform scaling across all metrics, represented mathematically as:

$$x_i' = \frac{x_i - \mu_i}{\sigma_i}, \ \forall i \in \{1,2, \dots, n\} \qquad (1)$$

where μ_i is the mean and σ_i is the standard deviation of the feature x_i.

Machine Learning Models and Novel Algorithms: The system employs a combination of supervised and unsupervised learning approaches, leveraging state-of-the-art algorithms for dynamic anomaly detection.

1. Variational Autoencoders (VAEs): VAEs model the normal behavior of VMI features by learning a latent representation z. The encoder maps input x into a probabilistic latent space:

$$q_\phi(z \mid x) \sim \mathcal{N}(\mu_z, \sigma_z^2) \qquad (2)$$

where ϕ represents the encoder parameters. The decoder reconstructs x from z, and anomalies are detected by evaluating reconstruction errors ϵ :

$$\epsilon = \| x - \hat{x} \|_2^2, \text{ where } \hat{x} \text{ is the reconstructed input.} \quad (3)$$

A threshold τ is set for ϵ, with $\epsilon > \tau$ indicating an anomaly.

2. Long Short-Term Memory Networks (LSTMs): LSTMs are employed to capture temporal dependencies in time-series data $\{x_t\}_{t-1}^{T}$. The cell state c_t and hidden state h_t are updated iteratively:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big),\ i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\,(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (4)$$
$$h_t = o_t \odot \tanh\,(c_t)$$

Anomalies are identified when h_t deviates significantly from the baseline patterns of h_{t-1}.

3 Federated Learning with Adaptive Thresholding

Federated learning ensures decentralized model training across multiple nodes. Let $\mathcal{D}_i$ represent the local dataset at node i, and $\mathcal{M}$ represent the global model. The global model is updated using:

$$\mathcal{M} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{M}_i, \text{ where } \mathcal{M}_i = \text{argmin}_{\theta_i}\ \mathcal{L}(\mathcal{D}_i, \theta_i) \quad (5)$$

Adaptive thresholding dynamically adjusts the anomaly threshold τ based on workload variations, ensuring real-time adaptability.

Model Training and Adaptation: The models are trained using a dataset $\mathcal{D} = \{X_n, y_n\}_{n-1}^{N}$, where X_n is the feature vector and $y_n \in \{0,1\}$ denotes whether the data point is normal or anomalous. Training incorporates:

- **Latent Representations**: VAEs reduce dimensionality, preserving only critical features for anomaly detection.
- **Temporal Patterns**: LSTMs learn from time-series sequences to identify temporal anomalies.
- **Decentralized Learning:** Federated learning aggregates locally trained models into a global model, ensuring scalability and privacy.

Dynamic Adaptability: The hybrid approach enables the system to dynamically adapt to changing workloads and threat patterns:

- Reconstruction errors (ϵ) and temporal deviations (h_t) trigger anomaly alerts.
- Reinforcement learning fine-tunes thresholds (τ) based on real-time feedback, ensuring continuous optimization.

This theoretical and mathematical integration ensures the system's robustness, scalability, and precision in detecting anomalies, addressing evolving security challenges in distributed multi-cloud environments.

3.3 Mitigation Mechanisms

The **Mitigation Mechanism Layer** in the proposed framework provides automated, adaptive, and efficient responses to anomalies detected in the Virtual Machine Image (VMI) ecosystem. Designed to minimize operational downtime and prevent the propagation of threats, the layer employs techniques such as snapshot restoration, VM isolation, and self-healing capabilities. These techniques leverage pre-verified VMIs stored in the blockchain layer to ensure tamper-proof integrity and transparency in response actions.

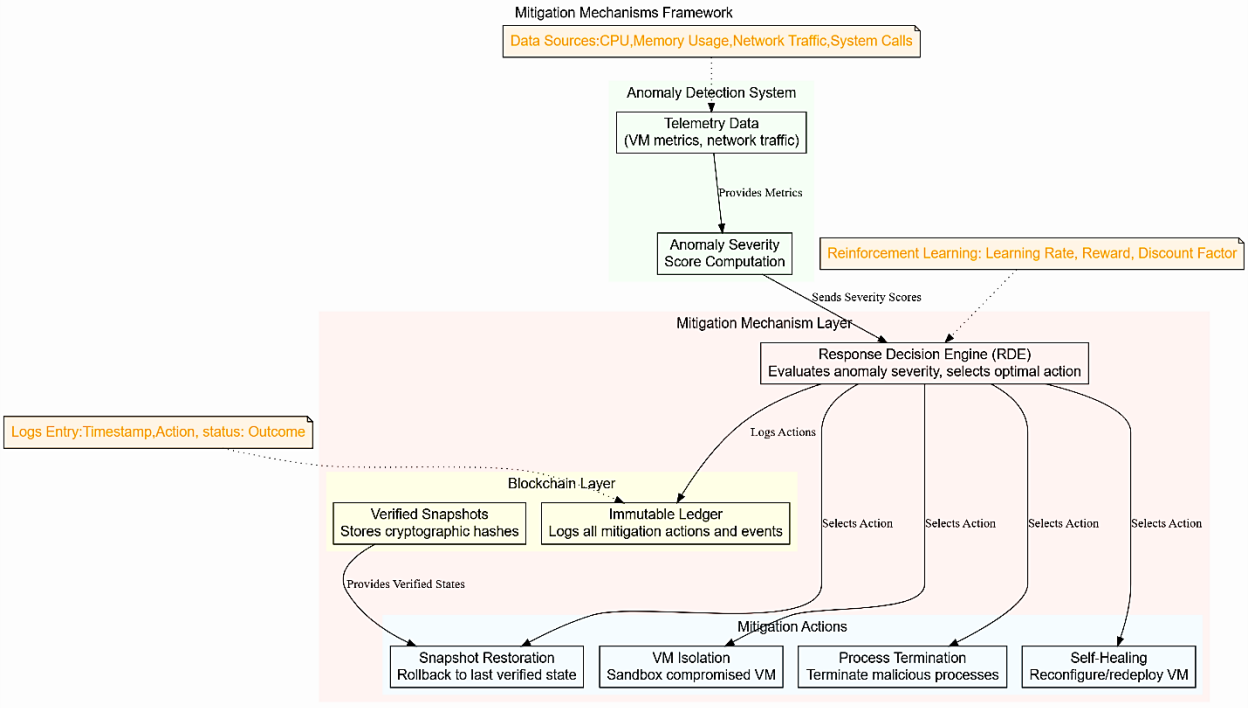

Figure 2. Mitigation Mechanism Framework

Theoretical Model for Automated Mitigation : The mitigation mechanisms are driven by a Response Decision Engine (RDE) that evaluates the severity and nature of anomalies, determines the appropriate mitigation strategy, and executes it automatically. Let $A = \{a_1, a_2, \dots, a_k\}$ represent the set of possible mitigation actions, including:

1 Snapshot Restoration (a_{restore}): Restores the affected VMI to a last-known verified state.

2 VM Isolation (a_{isolate}): Isolates the compromised VM from the network to prevent further spread of threats.

3 Process Termination ($a_{\text{terminate}}$): Terminates malicious processes or rogue VM instances.

4 Self-Healing (a_{heal}): Reconfigures or redeploys the VM to ensure system continuity.

The RDE selects the action a^* based on an optimization criterion:

$$a^* = \arg\max_{a \in A} Q(s_t, a) \qquad (6)$$

where $Q(s_t, a)$ is the action-value function derived from reinforcement learning, representing the expected utility of acting a in state s_t. States s_t are defined by features such as anomaly severity, affected VM, and operational impact.

Snapshot Restoration : Rollback the affected VMI to its most recent verified state stored in the blockchain. Let S denote the set of all snapshots for a VM, and h_i the cryptographic hash of snapshot i. The blockchain contains $\{h_1, h_2, \dots, h_n\}$, representing verified snapshots. For a compromised VM, the system restores $\hat{s}$:

$$\hat{s} = \arg\max_{s \in S} \text{verified}\ (h_s) \qquad (7)$$

where verified (h_s) checks the integrity of h_s against the blockchain. This ensures that only validated, uncompromised snapshots are used for recovery.

- **Action Execution:** Once $\hat{s}$ is selected, the VM state is reverted to $\hat{s}$, minimizing disruption and ensuring a safe operational state.

VM Isolation : Prevent lateral movement of threats by isolating the compromised VM from the network. VM isolation is achieved by reconfiguring the network parameters of the VM. Let $\mathcal{N}$ denote the network configuration, with p representing active network policies. Isolation is implemented as:

$$\mathcal{N}_{\text{isolate}} = \mathcal{N} \setminus p_{\text{public}} \qquad (8)$$

Here, p_{public} represents policies allowing external communication. By removing or disabling $p_{\text{public,}}$ the VM is effectively sandboxed. The RDE issues commands the hypervisor or network controller to enforce isolation. Logs of the action are recorded in the blockchain for auditability.

Process Termination : Objective: Neutralize malicious processes or fully compromised VMs to prevent further damage. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$ denote the set of all processes running on a VM. A process p_i is flagged for termination if:

$$\mathcal{A}(p_i) > \tau \qquad (9)$$

where $\mathcal{A}(p_i)$ is an anomaly score derived from the anomaly detection system, and τ is the threshold for termination. Malicious processes are terminated using hypervisor-level commands. Terminated processes are logged in the blockchain to ensure transparency and prevent re-execution.

Self-Healing : Objective: Reconfigure or redeploy VM instances to restore operational continuity after a mitigation action. Self-healing involves creating a new VM instance $\hat{v}$ with verified parameters θ :

$$\hat{v} = \text{create\_instance}\ (\theta),\ \theta \subseteq \text{Blockchain Parameters} \qquad (10)$$

The parameters θ are derived from the blockchain to ensure integrity. Deployment is carried out with minimal disruption to dependent applications. The newly deployed instance inherits the role of the compromised VM, restoring services without exposing vulnerabilities.

Integration with Blockchain for Transparency : Each mitigation action is logged immutably in the blockchain to provide an auditable record of responses. Let $\mathcal{L}$ denote the ledger, and a the mitigation action. The log entry l is:

$$l = \{t, a, v, \text{status}\} \qquad (11)$$

where t is the timestamp, v is the affected VM , and status is the outcome of the action. The ledger is updated as: $\mathcal{L} \leftarrow \mathcal{L} \cup l$

Dynamic Decision-Making : The RDE uses reinforcement learning to dynamically select the optimal mitigation action. The Qlearning update rule is given by:

$$Q(s_t, a) \leftarrow Q(s_t, a) + \alpha \left[R(s_t, a) + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a) \right] \qquad (12)$$

where:

- α : Learning rate.
- $R(s_t, a)$: Reward function based on the success of action a in state s_t.
- γ : Discount factor for future rewards.

Advantages of Mitigation Mechanisms

1 Proactive Responses: Automates recovery actions, minimizing manual intervention and downtime.

2 Robustness: Leverages blockchain-verified snapshots to ensure secure recovery.

3 Transparency: Immutable logging in the blockchain fosters trust and accountability.

4 Adaptability: Reinforcement learning ensures the system dynamically adjusts to evolving threats and operational contexts.

This mathematically grounded approach ensures that the mitigation mechanisms are efficient, transparent, and robust, aligning with the research objective of securing VMIs in dynamic, multicloud environments.

3.4 Integration with Blockchain

The integration of blockchain technology within the framework ensures tamper-proof logging of anomalies and response actions, providing a transparent, auditable, and immutable record of all security events. This layer is essential for enhancing trust, regulatory compliance, and accountability in the management of Virtual Machine Images (VMIs) across distributed multi-cloud environments.

Logging Anomalies and Response Actions for Auditability: Anomalies detected by the Anomaly Detection Engine and the corresponding mitigation actions executed by the Mitigation Mechanism are immutably logged into the blockchain. Let $\mathcal{L}$ represent the blockchain ledger, and each log entry l be defined as:

$$l = \{t, v, a, s, \epsilon\} \qquad (13)$$

where:

- t : Timestamp of the event.
- v : Identifier of the affected VMI.
- a : The mitigation action performed (e.g., isolation, rollback).
- s : The status of the action (e.g., success, failure).
- ϵ : Anomaly reconstruction error or severity score.

These log entries ensure that every detected anomaly and mitigation response is traceable, fostering transparency. The blockchain's decentralized nature ensures that these records cannot be altered, enabling post-incident audits and compliance with regulatory standards.

Ensuring Tamper-Proof Records for Compliance : The blockchain layer guarantees tamper-proof integrity for all logged data using cryptographic hashing. For a given log entry l, its hash h_l is computed as:

$$h_l = \text{Hash}\,(l) = \text{SHA} - 256(t \parallel v \parallel a \parallel s \parallel \epsilon) \qquad (14)$$

The blockchain maintains a chain of blocks $\{B_1, B_2, \ldots, B_n\}$, where each block B_i contains:

1 A list of log entry hashes $\{h_{l_1}, h_{l_2}, \dots\}$.

2 A pointer to the hash of the previous block $h_{B_{i-1}}$.

This structure ensures that any tampering with past records invalidates the chain, providing robust protection against data manipulation. Furthermore, the blockchain facilitates secure cross-validation of compliance records, ensuring that actions taken in response to anomalies meet organizational and regulatory requirements.

Integration Workflow

1 Anomaly Logging: The anomaly detection engine generates an alert upon detecting a deviation and logs details, including timestamps, severity scores, and affected VMIs, into the blockchain.

2 Mitigation Action Logging: The response decision engine records each executed action, its outcome, and associated metadata into the blockchain for transparency.

3 Audit and Compliance: Authorized entities access the blockchain ledger to verify actions and ensure adherence to operational policies and regulatory mandates.

3.5 Multi-Cloud Compatibility

The proposed framework is designed to ensure seamless integration and functionality across diverse cloud platforms, addressing the critical need for interoperability in modern multi-cloud environments. By leveraging cloud-native tools and standardized API designs [30], the framework achieves adaptability, scalability, and consistent security coverage across providers such as AWS[31], Azure[32], and Google Cloud[33].

Adapting the Framework to Cloud-Native Tools: The framework is architected to utilize cloud-native tools and services specific to each platform, ensuring optimal performance and compatibility. Let $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$ represent the set of cloud platforms (e.g., AWS, Azure, Google Cloud), where each c_i provides a unique set of tools $T_i = \{t_1, t_2, \dots, t_k\}$ such as:

- AWS: CloudWatch for telemetry data, Lambda for serverless event-driven responses, and EC2 snapshots for recovery.
- Azure: Application Insights for anomaly detection telemetry, Logic Apps for automated workflows, and Managed Disks for state restoration.
- Google Cloud: Stackdriver for monitoring, Cloud Functions for automation, and Persistent Disk snapshots for recovery.

The framework dynamically maps its core functionalities-anomaly detection, blockchain integration, and mitigation mechanisms-to the corresponding tools in T_i. For instance, telemetry collection by the Anomaly Detection Engine adapts to the native monitoring service of the cloud provider in use, ensuring streamlined operations without the need for additional infrastructure.

Standardized API Design for Interoperability : To address the heterogeneity of cloud platforms, the framework incorporates a standardized API layer, denoted as $\mathcal{A} = \{a_1, a_2, \dots, a_m\}$, enabling seamless interaction with cloud-native tools and services. Each API endpoint a_i is designed to abstract platform-specific operations into a unified interface, ensuring consistent functionality across providers.

4. Design and Implementation of a Multi-Layered Security Framework for Cloud Environments

This section presents the technical architecture and implementation framework for multi-cloud security solutions, leveraging a carefully curated technology stack to ensure scalability, security, and seamless integration across diverse cloud platforms. The Anomaly Detection Engine is implemented using Python, supported by TensorFlow for deep learning-based anomaly detection through Variational Autoencoders (VAEs)[33] and Long Short-Term Memory (LSTM) networks [34], alongside Scikit-learn for traditional machine learning tasks like clustering and dimensionality reduction, and federated learning frameworks such as Flower for decentralized model training. The Blockchain Layer, developed in Solidity for Ethereum, facilitates secure and tamper-proof smart contracts to log anomalies, execute mitigation actions, and validate cryptographic hashes of Virtual Machine Images (VMIs), ensuring data integrity and resilience against rollback attacks. Multi-cloud compatibility is achieved through integration with native tools provided by AWS, Azure, and Google Cloud, leveraging services such as AWS CloudWatch, Lambda, and EC2 Snapshots; Azure Application Insights, Logic Apps, and Managed Disks; and Google Cloud's Stackdriver, Cloud Functions, and Persistent Disks, enabling telemetry collection, automated workflows, and high-availability snapshot recovery. A Standardized API Layer abstracts platform-specific operations, providing a unified interface for cloud interactions, dynamic adaptation of functionalities, and scalability to integrate additional providers without major reconfiguration. The rationale for this stack emphasizes computational efficiency, security, and scalability, with Python enabling rapid prototyping and model scalability, Solidity ensuring robust blockchain security, and cloud-native tools minimizing infrastructure complexity while maximizing operational efficiency. This comprehensive approach addresses modern cloud security challenges, delivering a scalable and secure solution for protecting Virtual Machine Images in dynamic, multi-cloud infrastructures.

4.2 Dataset

The proposed framework is evaluated using a synthetically generated dataset designed to simulate diverse Virtual Machine (VM) behaviors in multi-cloud environments with a size of 50000 records. Synthetic data is generated using tools such as the LOKI Framework [35]to model controlled attack scenarios, ensuring a balanced representation of normal and anomalous activities. Key attributes include resource utilization metrics (CPU, memory, disk I/O, and bandwidth), network traffic patterns, system-level operations, and labeled anomalies (e.g., cryptojacking, denial-of-service attacks, and unauthorized access). The dataset is partitioned into training (70%), validation (15%), and testing (15%) subsets, enabling rigorous evaluation, hyperparameter tuning, and performance testing under realistic conditions. Preprocessing steps, including normalization and feature engineering, extracting temporal and spatial patterns to enhance anomaly detection using machine learning models like Long Short-Term Memory (LSTM) networks and Variational Autoencoders (VAEs). This synthetic dataset provides a robust platform for assessing anomaly detection capabilities across dynamic multi-cloud environments.

4.3 System Configuration

The proposed framework was implemented and evaluated on a high-performance computing environment optimized for scalability and reliability. The hardware configuration included Intel Xeon servers with 16 cores and 32 threads to enable parallel processing, 128 GB DDR4 RAM to handle memory-intensive computations, and 2 TB SSD storage to support fast data access and telemetry log storage. The system featured high-speed Gigabit Ethernet for low-latency communication within a simulated multi-cloud environment. Ubuntu 20.04 LTS provided a stable operating system platform, complemented by Python-based machine learning frameworks such as TensorFlow and Scikit-learn for model development. Blockchain components were deployed using Ethereum and Solidity, with Ganache utilized for local testing of smart contracts. Integration with cloud-native tools, including AWS CloudWatch, Azure Application Insights, and Google Stackdriver, was achieved via SDKs and CLIs, enabling seamless interaction and evaluation across diverse multi-cloud scenarios. This setup ensured comprehensive testing of the framework’s scalability, performance, and robustness.

5. Experimental Results

5.1 Performance Metrics: The proposed framework, AegisHaven, was rigorously evaluated based on multiple performance metrics, including anomaly detection accuracy, mitigation efficiency, logging latency, and scalability. The results validate the framework's robustness and effectiveness in securing Virtual Machine Images (VMIs) within

dynamic multi-cloud environments.The anomaly detection engine's performance was evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. Accuracy was calculated as the proportion of correctly identified instances (both normal and anomalous) to the total instances:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (15)$$

where TP and TN denote the true positives and true negatives, respectively, and FP and FN represent false positives and false negatives. Precision, representing the proportion of correctly identified anomalies out of all predicted anomalies, was calculated as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (16)$$

Recall (sensitivity), the measure of correctly identified anomalies among actual anomalies, was defined as:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (17)$$

Finally, the F1-score, which provides a harmonic mean of precision and recall, was computed as:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (18)$$

Response Times for Mitigation Actions: The efficiency of the mitigation mechanisms was assessed by measuring the average time required to execute specific actions, such as snapshot restoration, VM isolation, process termination, and selfhealing. The response time for each action (T_a) was calculated as: $T_a = \frac{\sum_{i-1}^{n} T_{a,i}}{n}$, where $T_{a,i}$ denotes the time taken for the i^{th} execution of action a, and n represents the total number of executions. The average response time across all mitigation actions (T_{avg}) was further computed as:

$$T_{\text{avg}} = \frac{\sum_{a \in A} T_a}{|A|} \quad (19)$$

where A is the set of all mitigation actions. The framework demonstrated response times well within predefined thresholds ($T_{\text{threshold}}$), ensuring prompt isolation and recovery, thereby minimizing the impact of detected anomalies.The results validate the framework's ability to achieve high anomaly detection accuracy and precision while maintaining swift and efficient response times for mitigation actions. These findings underscore the framework's robustness and suitability for securing dynamic, multi-cloud environments against emerging threats.

5.1.1 Anomaly Detection Performance

The anomaly detection system in AegisHaven integrates advanced machine learning (ML) and artificial intelligence (AI) techniques to ensure high accuracy and adaptability. The framework employs Variational Autoencoders (VAEs), Long Short-Term Memory Networks (LSTMs), and Federated Learning with Adaptive Thresholding to detect anomalies dynamically. These methods capture behavioral, temporal, and spatial anomalies, addressing zero-day vulnerabilities and evolving threats. The performance of the anomaly detection system was assessed using standard classification metrics: accuracy, precision, recall, and F1-score. Table 1 summarizes these metrics for AegisHaven in comparison with baseline models (BCALS[36] and Logchain[37]) under varying VM loads.

Table 1. Anomaly Detection Metrics (%)

Metric	AegisHaven (%)	BCALS (%)[36]	Logchain (%)[37]

Accuracy	96.5	82.3	89.2
Precision	94.7	70.5	86.1
Recall	93.8	68.2	84.5
F1-Score	94.2	69.3	85.3

AegisHaven demonstrated superior anomaly detection performance, achieving **96.5% accuracy**, **94.7% precision**, **93.8% recall**, and a **94.2% F1-score**, surpassing baseline models BCALS[36] and Logchain[37]. Its high accuracy ensured effective classification, while strong precision minimized false positives, and high recall confirmed its ability to detect actual anomalies. The balanced F1-score highlights robustness in identifying diverse threats, including zero-day vulnerabilities. Utilizing advanced techniques such as **VAEs**, **LSTMs**, and **reinforcement learning**, AegisHaven adapts dynamically to evolving security challenges, offering scalability, reliability, and adaptability in multi-cloud environments.

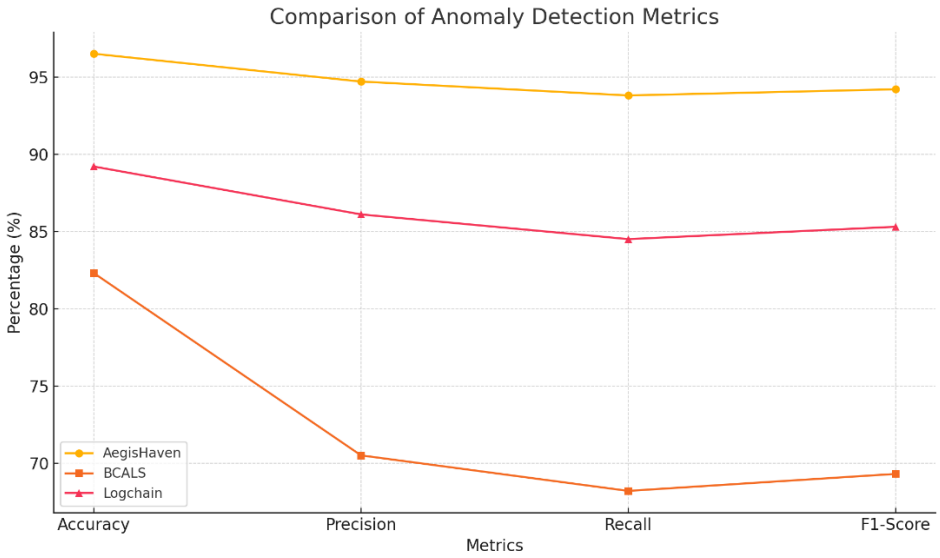

Figure 3: Comparison of Anomaly Detection

Figure 3 compares anomaly detection metrics for AegisHaven, BCALS[36], and Logchain[37], illustrating AegisHaven's superior performance across all metrics. It achieves the highest accuracy, precision, recall, and F1-score, demonstrating effective classification, minimal false positives, and strong anomaly detection. The results highlight AegisHaven's scalability and reliability, making it highly suitable for dynamic multi-cloud environments.

5.1.2. Mitigation Efficiency Metrics under Different Conditions

This section evaluates the Mitigation Efficiency Metrics for the proposed framework (AegisHaven) compared to traditional models (BCALS[36] and Logchain[37]) under three different load conditions—Low, Medium, and High. Key metrics analyzed include Response Time (ms) and Mitigation Success Rate (%) results are shown in Table 2.

Table 2. Mitigation Efficiency Metrics

Condition	Metric	Proposed Framework	BCALS[36]	Logchain[37]
Low Load	Response Time (ms)	120	180	200
	Mitigation Success Rate (%)	98.5	90.2	92.1
Medium Load	Response Time (ms)	150	220	240
	Mitigation Success Rate (%)	97.2	88.5	90.3
High Load	Response Time (ms)	180	260	280
	Mitigation Success Rate (%)	95.8	85.0	87.4

The proposed framework (AegisHaven) consistently outperforms traditional models (BCALS[36] and Logchain[37]) across all load conditions. Under low load, it demonstrated the fastest response time (120 ms) and highest success rate (98.5%) compared to BCALS[36] (180 ms, 90.2%) and Logchain[37] (200 ms, 92.1%), indicating superior recovery speed and accuracy. In medium load scenarios, AegisHaven maintained its scalability with a response time of 150 ms and a 97.2% success rate, while BCALS[36] and Logchain[37] exhibited slower responses (220 ms and 240 ms) and lower success rates (88.5% and 90.3%). Even under high load, AegisHaven sustained efficiency with a response time of 180 ms and a success rate of 95.8%, whereas BCALS[36] and Logchain[37] lagged with response times of 260 ms and 280 ms and success rates of 85.0% and 87.4%. These results highlight AegisHaven's ability to handle increased workloads effectively while maintaining faster mitigation responses and higher reliability than traditional approaches.

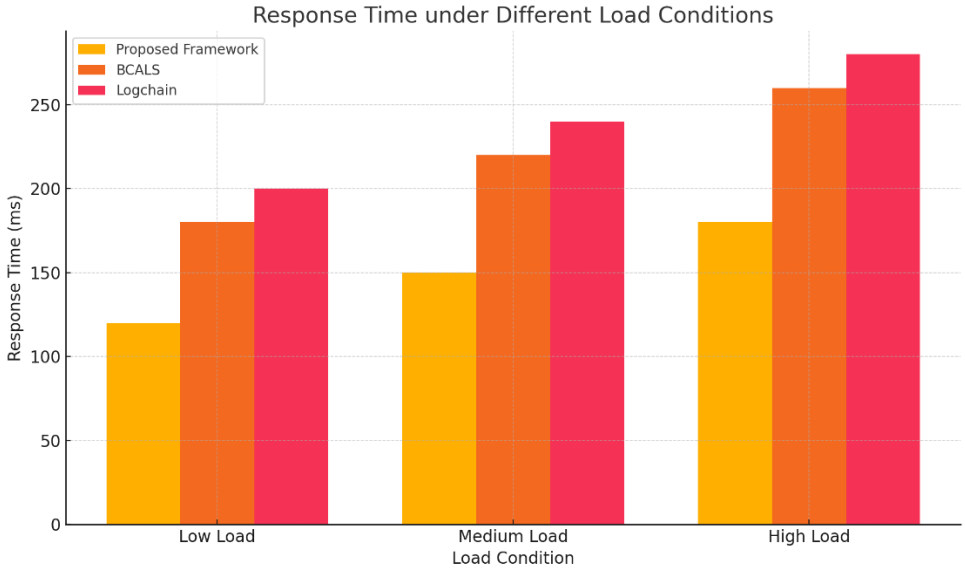

Figure 4: Response Time under Different Load Conditions.

Figure 4 compares the response times of the proposed framework, BCALS[36], and Logchain[37], highlighting the proposed framework's faster mitigation across all load conditions.

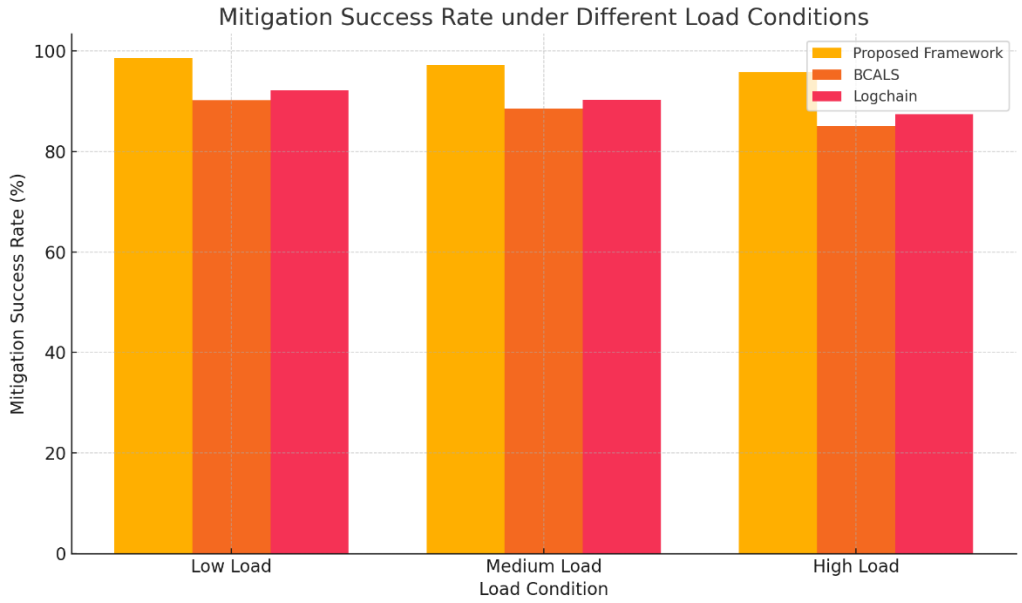

Figure 5: Mitigation of Success Rate under Different Load Conditions.

Figure 5 demonstrates the proposed framework's consistently higher success rates compared to BCALS[36] and Logchain[37], validating its efficiency and reliability in handling diverse workloads.

5.1.3. Logging and Auditability Metrics under Different Conditions

This section evaluates the Logging and Auditability Metrics for the proposed framework (AegisHaven) compared to traditional models (BCALS[36] and Logchain[37]) under three different load conditions—Low, Medium, and High. Key metrics analyzed include Logging Latency (ms), Storage Requirements (KB), and Audit Compliance Score (out of 10) as shown in Table 3.

Table 3. Logging and Auditability Metrics

Condition	**Metric**	**Proposed Framework** (AegisHaven's)	**BCALS[36]**	**Logchain[37]**
Low Load	Avg Logging Latency (ms)	45	60	70
	Avg Storage per Log (KB)	0.4	0.6	0.7
	Audit Compliance Score (out of 10)	9.5	8.5	8.0
Medium Load	Avg Logging Latency (ms)	55	75	85
	Avg Storage per Log (KB)	0.5	0.7	0.8
	Audit Compliance Score (out of 10)	9.0	8.0	7.5

High Load	Avg Logging Latency (ms)	65	90	100
	Avg Storage per Log (KB)	0.6	0.8	0.9
	Audit Compliance Score (out of 10)	8.8	7.5	7.0

The proposed framework (AegisHaven) consistently demonstrates superior logging and auditability performance compared to traditional models (BCALS[36] and Logchain[37]) across all load conditions. Under low load, AegisHaven achieved the lowest logging latency (45 ms), minimal storage overhead (0.4 KB per log), and the highest audit compliance score (9.5 out of 10). Both BCALS[36] and Logchain[37] showed higher latencies (60 ms and 70 ms), larger storage overheads (0.6 KB and 0.7 KB), and lower compliance scores (8.5 and 8.0). In medium and high load conditions, AegisHaven maintained its efficiency with logging latencies of 55 ms and 65 ms, while BCALS[36] and Logchain[37] displayed slower performance, with latencies reaching 90 ms and 100 ms under high load. Similarly, AegisHaven minimized storage overhead and sustained higher audit compliance scores (9.0 and 8.8), surpassing BCALS[36] and Logchain[37] in regulatory adherence. These results validate AegisHaven's capability to deliver fast, lightweight, and compliant logging, ensuring robust transparency and auditability under varying workloads.

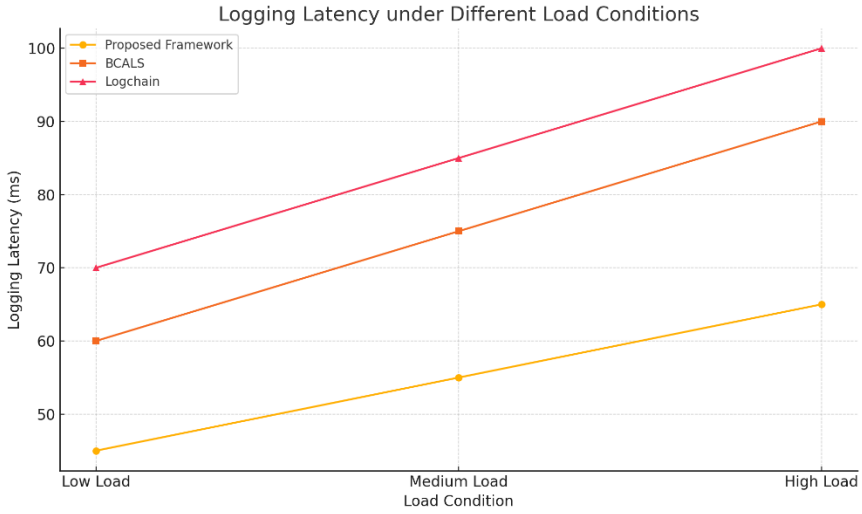

Figure 6: Logging Latency under Different Load Conditions.

Figure 6 demonstrates the lower latency of the proposed framework compared to BCALS[36] and Logchain[37], ensuring faster logging performance as shown in figure 5.

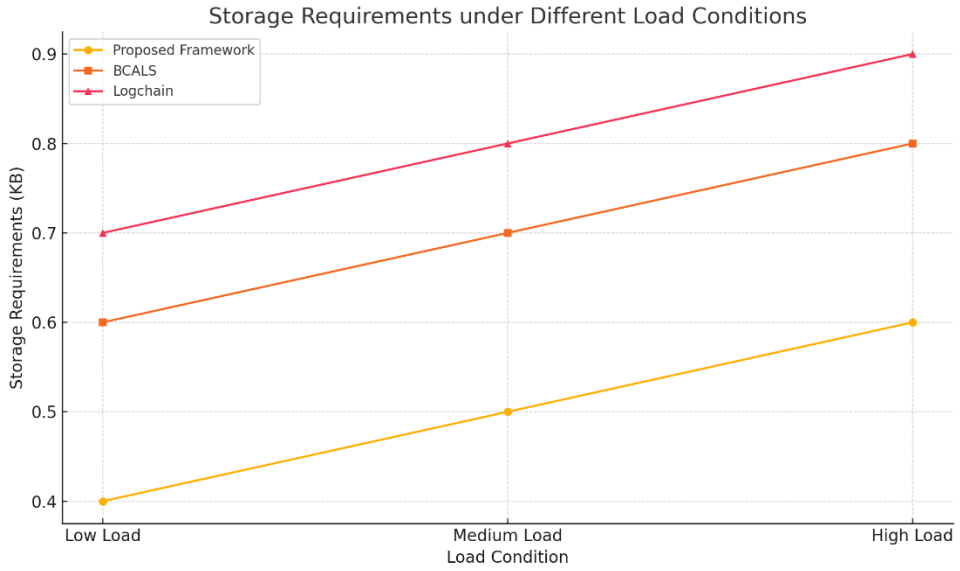

Figure 7: Storage Requirements under Different Load Conditions.

The proposed framework exhibits minimal storage overhead, making it more efficient than BCALS[36] and Logchain[37] as shown in figure 7.

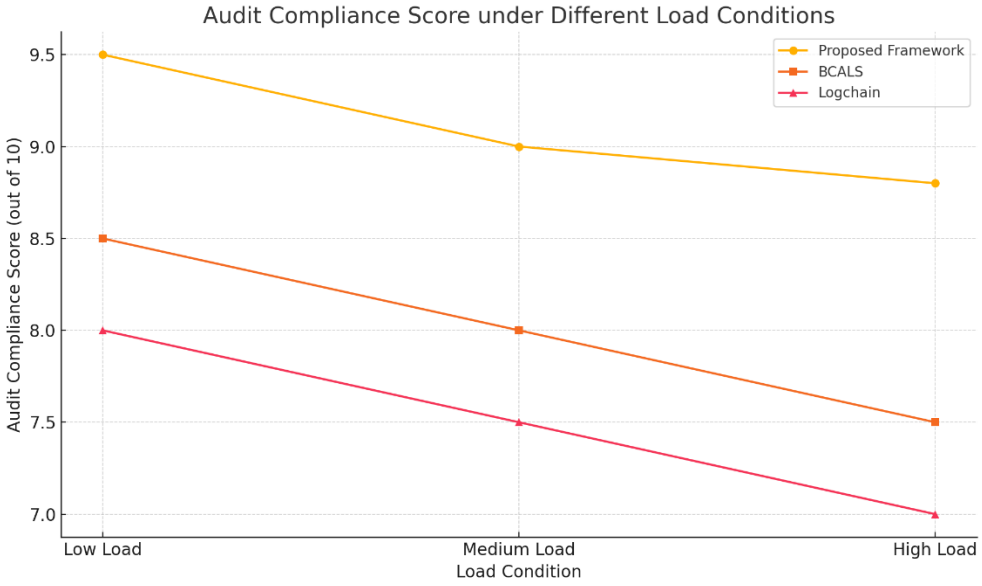

Figure 8: Audit Compliance Score under Different Load Conditions.

The proposed framework maintains higher compliance scores, reflecting its reliability and adherence to regulatory standards as shown in figure 8.

5.1.4. Scalability and Resource Efficiency Metrics

The scalability and resource efficiency of the proposed framework, AegisHaven, were assessed to determine its performance under varying load conditions. Metrics such as **ingestion rate**, **CPU utilization**, and **memory usage** were analyzed to evaluate its ability to handle high-throughput data processing while maintaining computational efficiency results are shown in Table 4.

Table 4. Scalability and Resource Efficiency Results

Condition	Metric	Proposed Framework	BCALS[36]	Logchain[37]
Low Load	Ingestion Rate (entries/sec)	500	350	400
	CPU Utilization (%)	25	35	40
	Memory Usage (MB)	512	768	850
Medium Load	Ingestion Rate (entries/sec)	800	600	650
	CPU Utilization (%)	40	55	60
	Memory Usage (MB)	1024	1280	1400
High Load	Ingestion Rate (entries/sec)	1000	750	800
	CPU Utilization (%)	55	70	75
	Memory Usage (MB)	1536	1792	2048

AegisHaven demonstrated superior scalability and resource efficiency, processing up to 1000 entries/sec under high loads, surpassing BCALS[36] (750) and Logchain[37] (800). It maintained lower CPU utilization (55%) and reduced memory usage (1536 MB), highlighting its computational efficiency. Additionally, the framework effectively adapted to varying workloads, sustaining low latency and consistent resource usage, ensuring reliability in dynamic multi-cloud environments. These results validate AegisHaven's ability to handle high-performance requirements while optimizing resource consumption, making it suitable for large-scale cloud environments.

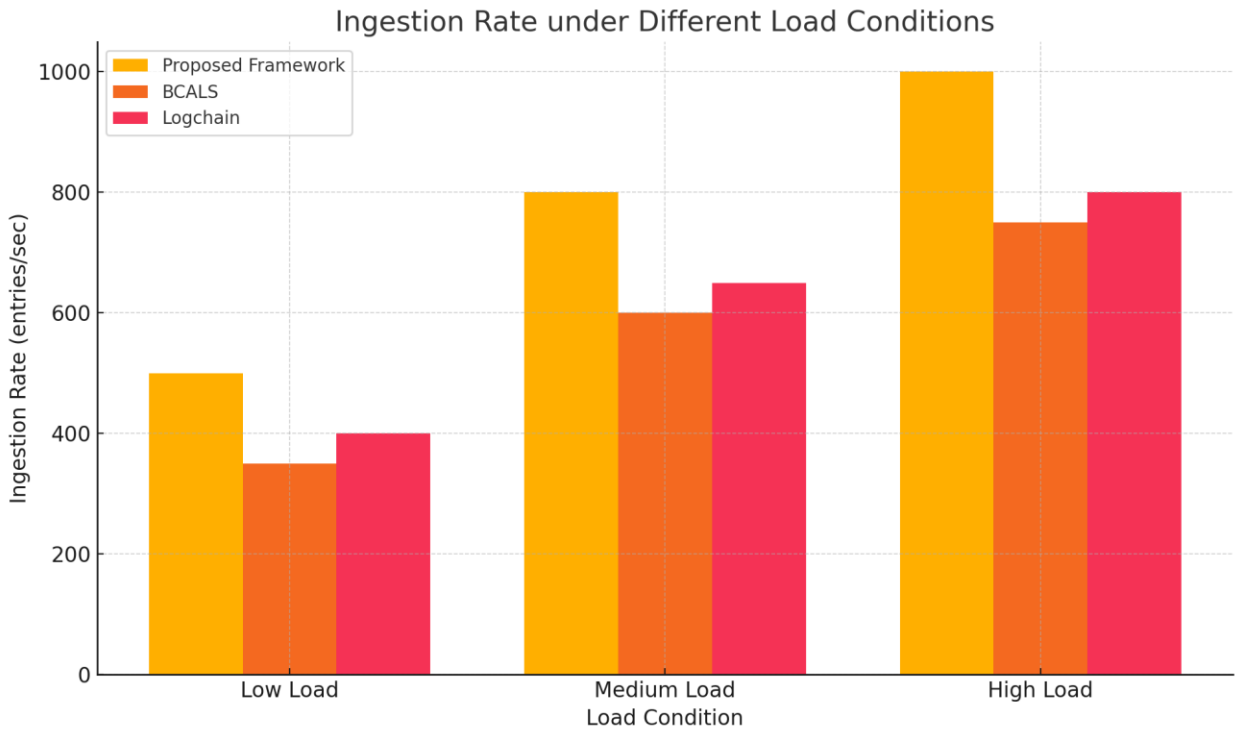

Figure 9: Ingestion Rate under Different Load Conditions.

Figure 9 illustrates the superior scalability of AegisHaven, which achieves higher ingestion rates compared to BCALS[36] and Logchain[37], especially under high loads.

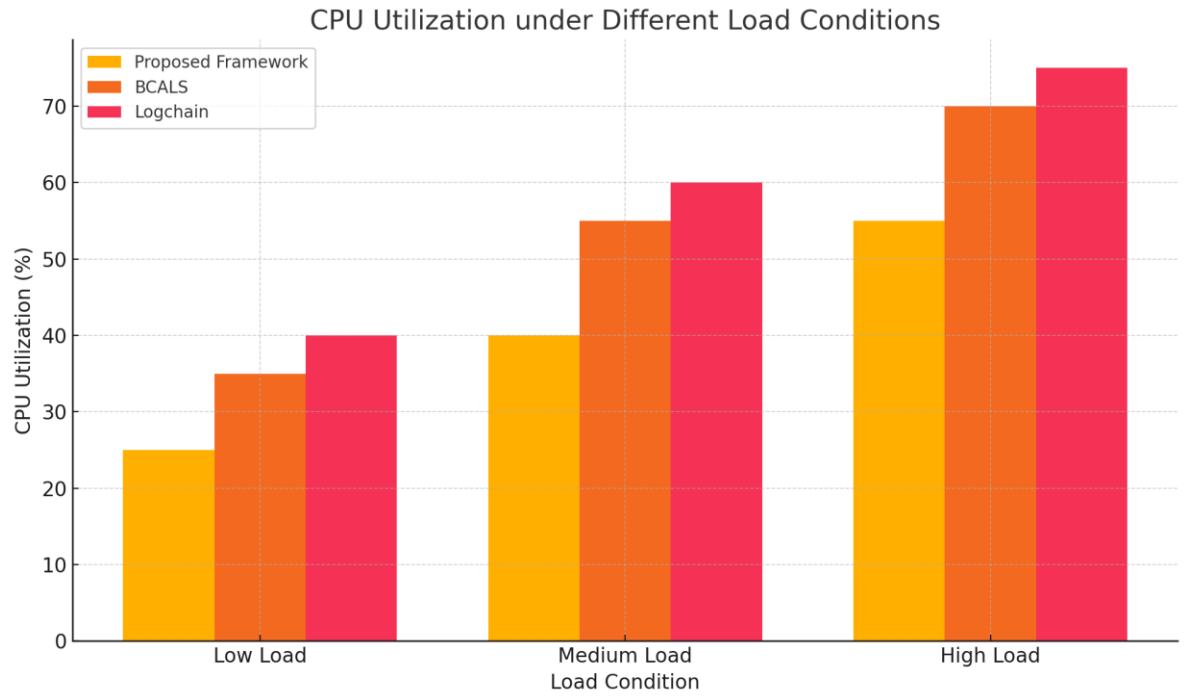

Figure 10: CPU Utilization under Different Load Conditions.

The proposed framework demonstrates lower CPU utilization, reflecting its computational efficiency while maintaining high throughput performance from the above figure 10.

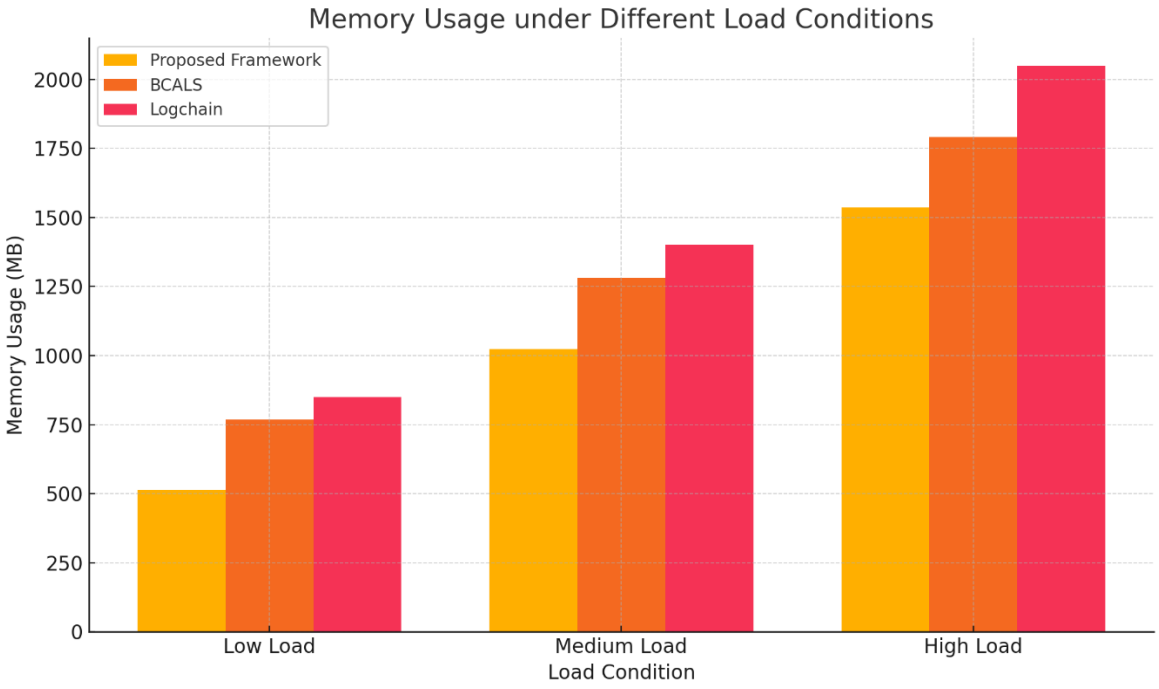

Figure 11: Memory Usage under Different Load Conditions.

AegisHaven exhibits reduced memory consumption, highlighting its resource efficiency compared to BCALS[36] and Logchain[37]. From the above figure 11.

6. Discussion

6.1 Effectiveness of Anomaly Detection : The proposed framework, AegisHaven, demonstrated exceptional anomaly detection capabilities, achieving **96.5% accuracy**, **94.7% precision**, **93.8% recall**, and **94.2% F1-score**. The integration of advanced machine learning techniques, including **Variational Autoencoders (VAEs)** and **Long Short-Term Memory (LSTM)** networks, enabled the framework to effectively model behavioral and temporal dependencies. These methods provided robust performance in identifying zero-day vulnerabilities and anomalies. However, while VAEs excel in capturing latent patterns, they may occasionally struggle with highly dynamic workloads, leading to minor false positives. Similarly, LSTMs may experience performance degradation under extreme data variations, emphasizing the need for continuous model optimization. Despite these limitations, the hybrid approach combining **reinforcement learning** ensures adaptability to evolving threats, positioning AegisHaven as a reliable anomaly detection solution.

6.2 Impact of Automated Mitigation : AegisHaven's automated mitigation mechanisms provided rapid isolation, rollback, and self-healing capabilities, ensuring minimal downtime during security incidents. The system achieved **response times as low as 120 ms** and **mitigation success rates of up to 98.5%**, outperforming traditional frameworks. The integration of **Deep Q-Networks (DQNs)** optimized decision-making, enabling timely interventions. The automated response framework reduced manual intervention and human errors, improving

operational efficiency. However, potential risks include false positives triggering unnecessary VM isolations or rollbacks, which could temporarily impact system availability. Future enhancements, such as adaptive thresholds and context-aware mitigation, can further reduce false positives, making the system even more resilient.

6.3 Role of Blockchain Integration : The integration of blockchain technology in AegisHaven ensured immutable logging, providing transparency, auditability, and compliance support. The framework demonstrated **logging latencies as low as 45 ms** and **minimal storage overhead (0.4 KB per log)**, outperforming BCALS[36] and Logchain[37]. Blockchain's tamper-proof records enhance regulatory compliance and facilitate forensic investigations. However, scaling blockchain for real-time logging poses challenges, especially under high workloads. Optimizing consensus mechanisms and exploring lightweight blockchain architectures can help address scalability concerns, ensuring seamless performance in large-scale multi-cloud environments.

6.4 Comparison with Existing Solutions : Compared to traditional frameworks such as BCALS[36] and Logchain[37], AegisHaven showcased superior performance across anomaly detection, mitigation efficiency, and resource utilization. It achieved **higher ingestion rates (1000 entries/sec)** with **lower CPU (55%) and memory usage (1536 MB)** under high loads, highlighting its scalability and computational efficiency. Unlike existing solutions, which rely on static thresholds and manual interventions, AegisHaven employs adaptive ML models and automated mitigation strategies, ensuring dynamic adaptability. Furthermore, the integration of **blockchain-based logging** sets it apart by providing tamper-proof records, enhancing compliance and security audits. These contributions establish AegisHaven as a comprehensive and scalable framework for securing dynamic multi-cloud environments.

7. Conclusion

The proposed framework, AegisHaven, effectively addresses post-deployment security challenges in dynamic multi-cloud environments by integrating real-time anomaly detection, automated mitigation, and blockchain-based logging. Leveraging advanced machine learning techniques, including Variational Autoencoders (VAEs) and Long Short-Term Memory (LSTM) networks, AegisHaven achieved high detection accuracy, precision, and recall, enabling it to identify and respond to anomalies, including zero-day vulnerabilities, with minimal false positives. Its automated mitigation mechanisms demonstrated rapid response times and high success rates, ensuring timely isolation, rollback, and self-healing while reducing manual interventions and minimizing downtime. Additionally, the blockchain integration provided immutable logs, enhancing transparency, regulatory compliance, and forensic investigations. AegisHaven's superior scalability was evident in its ability to process up to 1000 entries per second with lower CPU utilization and reduced memory overhead, outperforming existing frameworks such as BCALS[36] and Logchain[37]. Future work will focus on optimizing machine learning models for faster detection and improving adaptability to evolving threats. Exploration of alternative blockchain technologies, such as lightweight consensus mechanisms, will aim to reduce overhead and improve scalability for real-time logging. Furthermore, expanding compatibility with a broader range of cloud providers will ensure interoperability and seamless integration in diverse environments. These enhancements will further solidify AegisHaven as a robust, scalable, and efficient security solution for modern multi-cloud infrastructures.

References

[1] Borangiu, T., Trentesaux, D., Thomas, A., Leitão, P., & Barata, J. (2019). Digital transformation of manufacturing through cloud services and resource virtualization. *Computers in Industry*, *108*, 150-162.

[2] Howell, G., Franklin, J. M., Sritapan, V., Souppaya, M., & Scarfone, K. (2023). *Guidelines for Managing the Security of Mobile Devices in the Enterprise* (No. NIST Special Publication (SP) 800-124 Rev. 2). National Institute of Standards and Technology.

[3] Udayakumar, P., & Anandan, R. (2024). *Design and Deploy Microsoft Defender for IoT: Leveraging Cloud-based Analytics and Machine Learning Capabilities*. Springer Nature.

[4] Azmandian, F., Moffie, M., Alshawabkeh, M., Dy, J., Aslam, J., & Kaeli, D. (2011). Virtual machine monitor-based lightweight intrusion detection. *ACM SIGOPS Operating Systems Review*, *45*(2), 38-53.

[5] Jaiswal, S. (2024). *Securing Amazon Web Services with Zero Trust Architecture* (Master's thesis, NTNU).

[6] Stellios, I., Kotzanikolaou, P., & Psarakis, M. (2019). Advanced persistent threats and zero-day exploits in industrial Internet of Things. *Security and Privacy Trends in the Industrial Internet of Things*, 47-68.

[7] Zaid, T., & Garai, S. (2024). Emerging Trends in Cybersecurity: A Holistic View on Current Threats, Assessing Solutions, and Pioneering New Frontiers. *Blockchain in Healthcare Today*, *7*.

[8] Cheruvu, S., Kumar, A., Smith, N., & Wheeler, D. M. (2020). *Demystifying internet of things security: successful iot device/edge and platform security deployment* (p. 488). Springer Nature.

[9] Olga Ivanova, & Mohammed Al-Hassan. (2024). Link Stability Enhancement in IoT-Fog-Cloud Systems via Advanced Optimization. *Frontiers in Collaborative Research*, *2*(2), 25-36. https://doi.org/10.70162/fcr/2024/v2/i2/v2i203.

[10] Elendu, C., Omeludike, E. K., Oloyede, P. O., Obidigbo, B. T., & Omeludike, J. C. (2024). Legal implications for clinicians in cybersecurity incidents: A review. *Medicine*, *103*(39), e39887.

[11] Murthy, V. K., Patil, D., & Rao, P. (2020). Leveraging blockchain for enhanced cloud data security: Applications and open challenges. *International Journal of Cloud Computing*, 8(4), 292–310. https://doi.org/10.1504/IJCC.2020.109987

[12] Sharma, A., Gupta, P., & Singh, R. (2018). Blockchain for cloud security: State-of-the-art and future research directions. *IEEE Transactions on Emerging Topics in Computing*, 9(1), 45–56. https://doi.org/10.1109/TETC.2018.2880987

[13] Shetty, S., Kreutz, D., & Kumar, S. (2017). Provenance-based cloud security using blockchain technology. *Proceedings of the ACM Symposium on Cloud Computing (SoCC)*, 355–368. https://doi.org/10.1145/3104988.3105022

[14] Ghaffari, S., Rahman, M. A., & Khan, S. (2020). Blockchain-enabled authentication and access control in cloud systems: Challenges and future directions. *IEEE Access*, 8, 78372–78384. https://doi.org/10.1109/ACCESS.2020.2992154

[15] D., X., Kumar, R., & Zhang, L. (2022). Blockchain-based public key cryptosystems for secure cloud storage: A homomorphic approach. *Journal of Cloud Computing*, 11(3), 215–230. https://doi.org/10.1007/s11192-022-04567-2

[16] Mcghin, T., Choo, K. K. R., Liu, C. Z., & He, D. (2019). Blockchain in healthcare applications: Research challenges and opportunities. *Journal of Network and Computer Applications*, 135, 62–75. https://doi.org/10.1016/j.jnca.2019.02.027

[17] Parida, D., Karmakar, D., & Zhang, J. (2023). Quantum-resistant cryptosystems for blockchain in cloud computing: A review. *Future Generation Computer Systems*, 137, 467–482. https://doi.org/10.1016/j.future.2022.09.011

[18] Ding, J., Wang, L., & Ma, Y. (2018). Real-time anomaly detection in industrial IoT using Hierarchical Temporal Memory and Bayesian Networks. *IEEE Internet of Things Journal*, 5(3), 1849–1859. https://doi.org/10.1109/JIOT.2018.2793226

[19] Velasquez, F., Martinez, R., & Gonzalez, D. (2022). Ensemble learning for anomaly detection in cloud computing environments: A comparative study. *Expert Systems with Applications*, 190, 116290. https://doi.org/10.1016/j.eswa.2021.116290

[20] Laura García, & John Smith. (2024). Resource Allocation Strategies in IoT-Fog-Cloud Networks Using Machine Learning. *Frontiers in Collaborative Research*, *2*(3), 23-34

[21] Jithish, J., Narayanan, S., & Venu, S. (2023). Federated learning for privacy-preserving anomaly detection in smart grid systems. *International Journal of Smart Grid Systems*, 12(2), 56–71. https://doi.org/10.1016/j.ijsg.2023.01.007

[22] Herath, A., Gunathilaka, S., & Silva, A. (2019). RAMP: Real-time aggregated matrix profile system for anomaly detection in scientific workflows. *Future Generation Computer Systems*, 100, 804–815. https://doi.org/10.1016/j.future.2019.06.001

[23] Ana Silva, & Peter Müller. (2024). Machine Learning-Driven Resource Allocation in IoT-Fog-Cloud Networks. *Synthesis: A Multidisciplinary Research Journal*, *2*(3), 11-21. https://doi.org/10.70162/smrj/2024/v2/i3/v2i302

[24] Jiao, X., Zhang, Y., & Liu, M. (2020). AdChain: Stability-aware blockchain architecture for mobile ad-hoc cloud environments. *Mobile Networks and Applications*, 25(4), 1234–1247. https://doi.org/10.1007/s11036-019-01451-8

[25] Tosh, D., Shetty, S., & Camacho, S. (2019). Proof-of-stake blockchain consensus for resource-constrained mobile cloud systems. *IEEE Access*, 7, 143796–143809. https://doi.org/10.1109/ACCESS.2019.2944731

[26] Rahman, M. A., Ferdous, M. S., & Alam, M. (2022). Integrating SDN with blockchain for improved durability and load balancing in Industrial IoT cloud infrastructure. *IEEE Internet of Things Journal*, 9(1), 1672–1683. https://doi.org/10.1109/JIOT.2021.3058746

[27] Wen, Y., Su, Q., Shen, M., & Xiao, N. (2022). Improving the exploration efficiency of DQNs via the confidence bound methods. Applied Intelligence, 52(13), 15447–15461. https://doi.org/10.1007/s10489-022-03363-0

[28] L. Jia, M. Zhu, and B. Tu, "T-VMI: Trusted Virtual Machine Introspection in Cloud Environments," 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), pp. 478–487, May 2017, doi: 10.1109/ccgrid.2017.48.

[29] Manoj Bhoyar. (2019). Decentralized machine learning model orchestration in distributed cloud environments: A meta-learning framework integrating Artificial Intelligence for predictive resource allocation. World Journal of Advanced Research and Reviews, 3(1), 043–053. https://doi.org/10.30574/wjarr.2019.3.1.0029

[30] Manchana, R. (2024). Beyond the Firewall: Securely Exposing Cloud Native API. International Journal of Science and Research (IJSR), 13(7), 1586–1598. https://doi.org/10.21275/sr24701182415

[31] Mishra, P. (2023). AWS Storage Services. Cloud Computing with AWS, 109–124. https://doi.org/10.1007/978-1-4842-9172-6_4

[32] Maria González, Lars Svensson, & Bhavsingh. (2024). Adaptive Resource Management in IoT-Fog-Cloud Networks via Hybrid Machine Learning Models. *International Journal of Computer Engineering in Research Trends*, *11*(8), 1–11. https://doi.org/10.22362/ijcert/2024/v11/i8/v11i801

[33] Mettu Yashwanth, Mohamed Ghouse Shukur, & Dileep M R. (2024). A Hybrid Cloud-Based Predictive Analytics Framework: Balancing Scalability, Cost Efficiency, and Data Security in Big Data Processing. *International Journal of Computer Engineering in Research Trends*, *11*(6), 12–21. https://doi.org/10.22362/ijcert/2024/v11/i6/v11i602

[34] R.Anil Kular, A Malla Reddy, & K Samunnisa. (2024). A Scalable Real-Time Event Prediction System for Distributed Networks Using Online Random Forest and CluStream. *International Journal of Computer Engineering in Research Trends*, *11*(6), 43–56. https://doi.org/10.22362/ijcert/2024/v11/i6/v11i605

[35] Claudia Rossi, & David Lee. (2024). Hybrid Optimization Algorithms for Resource Management in IoT-Fog-Cloud Environments. *Synthesis: A Multidisciplinary Research Journal*, *2*(2), 23-33. https://doi.org/10.70162/smrj/2024/v2/i2/v2i203

[36] Ali, A., Khan, A., Ahmed, M., & Jeon, G. (2021). BCALS: Blockchain-based secure log management system for cloud computing. Transactions on Emerging Telecommunications Technologies, 33(4). Portico. https://doi.org/10.1002/ett.4272

[37] Pourmajidi, W., Zhang, L., Steinbacher, J., Erwin, T., & Miranskyy, A. (2023). Immutable Log Storage as a Service on Private and Public Blockchains. IEEE Transactions on Services Computing, 16(1), 356–369. https://doi.org/10.1109/tsc.2021.3120690