Anomaly Detection and Automated Mitigation for Microservices Security with AI

Vijay Ramamoorthi
Independent Researcher

Abstract

Microservices are becoming increasingly fundamental to modern scalable applications, yet their distributed nature makes them susceptible to complex cyber-attacks. Traditional security solutions, especially static, rule-based systems, fail to keep up with the dynamic threats presented by these architectures. This paper proposes an AI-based framework for real-time intrusion detection and automated mitigation within microservices. By leveraging unsupervised learning for anomaly detection and reinforcement learning for dynamic firewall adjustments and service isolation, the framework adapts to evolving threats autonomously. Experimental evaluations demonstrate that AI-driven security solutions can significantly enhance detection accuracy, reduce response times, and maintain system availability while minimizing downtime in real-world microservice environments. This paper discusses the framework's architecture, highlights its implementation, and presents results that validate the efficacy of AI-driven security strategies for microservices.

Keywords: Microservices security, AI-driven security, intrusion detection, anomaly detection, unsupervised learning, reinforcement learning, firewall automation, automated mitigation.

Introduction

The advent of microservice architectures has revolutionized the design and deployment of scalable applications, offering modularity, flexibility, and enhanced fault tolerance [1]–[3]. However, as organizations increasingly adopt microservices, new security challenges have emerged. These challenges stem from the distributed nature of microservices, where individual services interact with one another through APIs and networks, creating an expanded attack surface. Traditional security mechanisms, which rely on static rule-based intrusion detection systems (IDS), are ill-suited to handle the dynamic, interconnected, and rapidly changing environments that microservices introduce. This static approach results in delayed threat detection and response, as these systems are unable to adapt to evolving threats or learn from ongoing attacks [4], [5].

Current security solutions in microservices primarily focus on perimeter defense and manual mitigation, which are ineffective against complex, multi-vector attacks. Microservices are often deployed at scale in cloud-native environments where the dynamic nature of service provisioning, scaling, and communication renders traditional IDS inefficient and reactive. These shortcomings highlight the urgent need for intelligent, adaptive, and real-time security mechanisms capable of learning from the system's behavior and responding autonomously to potential threats [6].

Artificial intelligence (AI) emerges as a powerful solution to address these limitations. AI can provide an adaptive security framework capable of learning from historical data, monitoring system behavior in real time, and predicting potential threats based on anomalies. By integrating machine learning, particularly unsupervised learning for anomaly detection and reinforcement learning for automated responses, AI-driven systems can enhance the accuracy and speed of intrusion detection and mitigate attacks without manual intervention. The dynamic nature of AI-based approaches also enables them to evolve with changing attack vectors, making them more effective than static, rule-based systems [7].

In the field of AI-driven security for microservices, numerous research works have explored various aspects of intrusion detection and automated mitigation techniques. Zoppi et al. (2021) discuss the integration of meta-learning approaches to enhance unsupervised learning for intrusion detection in critical systems, focusing on reducing misclassifications in dynamic environments. This is crucial for improving the adaptability of AI-based systems in handling zero-day attacks in microservices architectures [8]. Similarly, Alom and Taha (2017) propose an unsupervised deep learning framework using autoencoders and restricted Boltzmann machines, which has shown significant improvements in detection accuracy, particularly in identifying novel attack patterns that static, rule-based systems may miss [9].The integration of reinforcement learning into intrusion detection systems is also gaining attention. For example, Xing et al. (2019) explore reinforcement learning-based mechanisms to enhance trust evaluation in autonomous vehicular networks. Their approach demonstrates the potential for reinforcement learning to improve the dynamic adaptability of intrusion detection and mitigation processes [10]. This concept is extended to microservice architectures in works such as Liang et al. (2022), who apply variational few-shot learning and reinforcement learning for distributed IoT-based microservices, ensuring effective detection of anomalies even in resource-constrained environments [11].

Furthermore, unsupervised learning continues to play a pivotal role in enhancing the scalability and flexibility of AI-driven security frameworks. Choi et al. (2019) discuss the development of an unsupervised learning model for network intrusion detection using autoencoders, achieving high accuracy without requiring large labeled datasets, making it particularly suited for evolving microservice environments [12]. Similarly, Ravi and Shalinie (2020) combine supervised deep learning with unsupervised clustering techniques in an IoT network security framework, highlighting the potential of hybrid approaches to address limitations of traditional methods [13]. The challenge of limited labeled data, especially in large-scale systems like cloud-based microservices, is addressed by Aouedi et al. (2023), who propose federated learning approaches that combine local and global models to enhance intrusion detection without compromising data privacy [14]. Gao et al. (2018) similarly explore semi-supervised learning in cloud-based robotic systems, which helps mitigate issues related to data scarcity and improves system resilience against cyber threats [15]. In recent years, the integration of blockchain technology with AI-based intrusion detection has also gained traction. Al-Kadi et al. (2021) propose a deep blockchain-enabled framework for securing IoT and cloud networks, which leverages machine learning models alongside blockchain to ensure secure and accurate intrusion detection [16].

The use of adversarial learning in combination with unsupervised anomaly detection is another area of interest, as seen in Idrissi et al. (2022), where generative adversarial networks are employed to detect anomalies in IoT devices [17]. Finally, various studies, such as Ahsan et al. (2022) and Raza et al. (2017), focus on optimizing feature selection and utilizing fuzziness in semi-supervised learning to enhance intrusion detection accuracy while minimizing false positives [18], [19]. These works collectively highlight the ongoing advancements in AI-driven security frameworks for microservices, showcasing a shift towards more dynamic, unsupervised, and reinforcement learning-based solutions capable of adapting to complex and evolving cyber threats [20].

The objective of this paper is to propose an AI-driven framework that enhances microservice security by integrating real-time intrusion detection and automated mitigation techniques. Specifically, we focus on utilizing unsupervised learning models to detect anomalies and employing reinforcement learning to automate dynamic responses such as firewall adjustments and service isolation. This framework is designed to improve system resilience by offering timely and precise responses to potential threats, minimizing the impact on system performance and availability.

In the following sections, we present the architecture and implementation details of our proposed AI-driven framework, describe the experimental setup used for evaluation, and discuss the results and implications of our findings. By demonstrating the effectiveness of AI in securing microservices, this paper aims to contribute to the broader understanding of how intelligent systems can protect next-generation cloud-native applications from evolving cyber threats.

Framework Design: Intelligent Security for Microservices

The proposed AI-driven security framework for microservices consists of several key components that work together to detect and mitigate security threats in real-time. This section outlines the architecture, detailing how the data collection, AI-based intrusion detection, and automated mitigation engines interact to provide a robust defense mechanism against evolving threats. The framework integrates machine learning techniques, such as unsupervised learning for anomaly detection and reinforcement learning for automated mitigation, to enhance the security of distributed microservice systems.

Architecture Overview

The architecture of the AI-driven security framework is designed to seamlessly integrate with existing microservice environments, addressing the inherent security challenges posed by their distributed nature. The framework comprises three main components: the data collection layer, the AI-based intrusion detection engine, and the automated mitigation engine. These components operate in conjunction with a continuous monitoring and feedback loop, allowing the system to adapt to new threats dynamically. The data collection layer is responsible for gathering real-time data from various sources within the microservice ecosystem. This includes logs from API requests, inter-service communications, network traffic, and resource usage metrics from individual services. The data is preprocessed to extract relevant features, such as

request frequency, payload sizes, response times, and access patterns. These features are then used to feed the machine learning models in the detection engine. By collecting data from multiple touchpoints, the framework ensures comprehensive visibility into the system's behavior, enabling more accurate detection of abnormal activity.

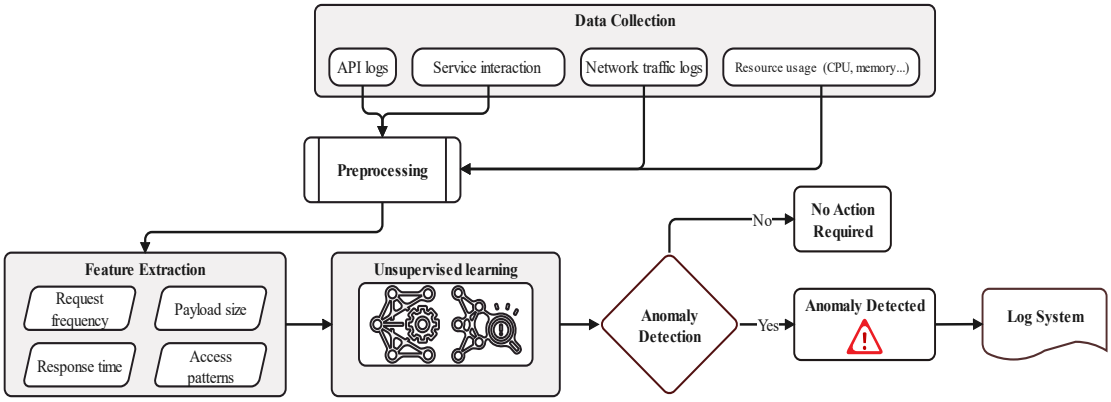

Figure 1. AI-driven intrusion detection framework for microservice architectures

Intrusion Detection via Unsupervised Learning

The AI-based intrusion detection engine employs unsupervised learning models to identify potential security threats by detecting anomalies in the collected data. Given the dynamic and evolving nature of microservice architectures, labeling data for every possible attack scenario is impractical. Thus, unsupervised learning, which does not rely on labeled data, is particularly suited to this environment.

1. Data Sources and Feature Extraction: The data for intrusion detection is sourced from multiple areas of the system, including API call logs, network traffic data, and metrics from inter-service communications. Key features for analysis include the volume and frequency of API requests, response times, payload sizes, and service interaction patterns. This feature set captures both the temporal and spatial behavior of services, allowing the detection engine to recognize deviations from normal patterns. Preprocessing steps normalize the data, ensuring it is structured for input into machine learning models.

2. Unsupervised Learning Models: The framework employs a combination of unsupervised learning models to detect anomalies. Isolation forests are used to identify outliers by partitioning the data and flagging points that are easily isolated from the rest of the data. This is particularly effective for spotting rare, abnormal events, such as

unusual spikes in API traffic or atypical inter-service communications. Autoencoders are used to compress normal service behavior into a lower-dimensional space. When new data significantly deviates from this learned representation, it is flagged as anomalous. Clustering algorithms, such as DBSCAN and k-means, are employed to group similar behaviors into clusters, with any data points falling outside these clusters being treated as potential threats.

3. Detection Workflow: The detection engine continuously monitors the incoming data stream, comparing new observations against the learned models of normal behavior. When an anomaly is detected, it triggers an alert that is logged and passed on to the automated mitigation engine for further action. The detection models are periodically retrained using updated data to account for changes in normal system behavior, ensuring the framework adapts to evolving workloads and service patterns [21].

Automated Response with Reinforcement Learning

Once a potential intrusion is detected, the system must quickly respond to mitigate the threat and minimize disruption. The automated mitigation engine, powered by reinforcement learning (RL), dynamically adjusts system defenses in real-time. Unlike static rule-based systems, the RL-based approach adapts to ongoing attacks and system changes by learning from the consequences of its actions.

1. Dynamic Firewall Management: The mitigation engine uses reinforcement learning to dynamically adjust firewall settings and enforce security policies. The state-space for the RL model consists of real-time metrics from the microservice environment, such as network traffic levels, CPU usage, and detected anomalies. The action-space includes potential security actions, such as blocking or throttling specific API calls, adjusting access control policies, and rerouting traffic. The reward function is designed to maximize system security while minimizing disruption to normal operations. For instance, successfully blocking an attack while allowing legitimate traffic to flow receives a high reward, while actions that unnecessarily degrade system performance or block legitimate requests are penalized.

2. Service Isolation and Recovery: In the event of a detected intrusion that poses a high risk, the RL agent can isolate compromised services. This may involve rerouting traffic away from the affected service, suspending certain API endpoints, or limiting the communication between compromised services and the rest of the system. By isolating compromised services, the framework ensures that the rest of the system remains

operational, minimizing the impact of the attack. The framework also supports compensating operations, which can be automatically initiated to restore service integrity after an intrusion is mitigated. These operations function similarly to μVerum's recovery mechanisms, reverting the system to a safe state by undoing malicious changes.

3. Real-Time Adaptation: One of the key advantages of reinforcement learning in this framework is its ability to adapt to new and evolving attack vectors in real-time. The RL model continuously updates its policy based on feedback from the system, improving its decision-making over time. For example, if a particular firewall adjustment proves effective in mitigating a specific type of attack, the model learns to prioritize similar actions in future scenarios. This adaptability is crucial in environments where new vulnerabilities and attack techniques can emerge rapidly, ensuring that the framework remains resilient against previously unknown threats.

A critical aspect of the framework is the continuous feedback loop, which ensures that both the intrusion detection engine and the mitigation engine are constantly learning and adapting. Data from past incidents, including successful and unsuccessful mitigation actions, are logged and used to update the models. The unsupervised learning models for anomaly detection are periodically retrained to account for new service behaviors, while the reinforcement learning model refines its policy based on the outcomes of past decisions. This continuous learning process allows the framework to evolve alongside the system it protects, maintaining a high level of security even as microservice architectures grow and change.

Evaluation & Experimental Results

In this section, we evaluate the performance and effectiveness of the proposed AI-driven security framework for microservices. Our evaluation focuses on the accuracy of the unsupervised learning-based intrusion detection, the effectiveness of the reinforcement learning-based automated mitigation, and the overall impact on system performance and scalability. We simulate a real-world microservice architecture subjected to a variety of attack scenarios to test the robustness and adaptability of the framework.

Experimental Setup

To simulate a realistic environment, we deploy a microservice-based architecture modeled after a typical e-commerce platform, consisting of services for user authentication, product catalog management, payment processing, and order tracking. Each service communicates over HTTP APIs, and the deployment includes a load balancer and distributed database.

Several types of security attacks are simulated to test the framework, including:

- **DDoS (Distributed Denial of Service)**: Flooding the system with excessive traffic to degrade service performance.
- **API Misuse**: Abuse of APIs with invalid requests, such as frequent failed login attempts or malformed data.
- **SQL Injection**: Injecting malicious SQL code into user inputs to exploit database vulnerabilities.
- **Cross-Service Communication Tampering**: Interfering with inter-service communication to simulate a compromised service affecting others.

The AI-driven security framework, consisting of the unsupervised learning-based detection engine and the reinforcement learning-based mitigation engine, is deployed in this architecture. We compare its performance against a traditional intrusion detection system (IDS) with rule-based anomaly detection and manual mitigation.

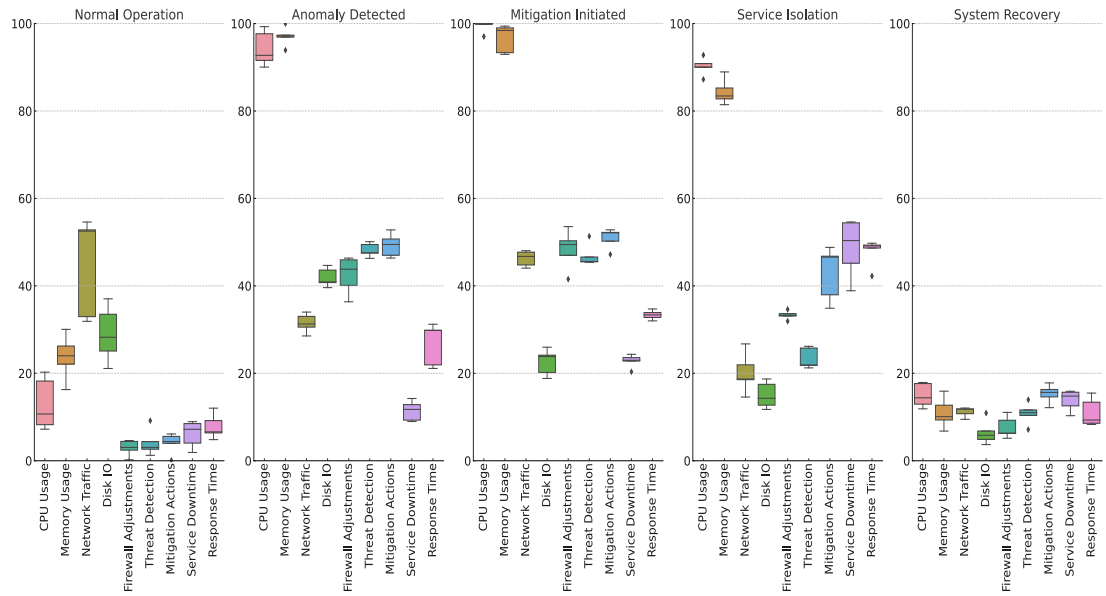

Figure 2. Distribution of System Metrics Across Different Phases of Operation. Each subplot represents a specific metric, with the X-axis showing the different phases (Normal Operation, Anomaly Detected, Mitigation Initiated, Service Isolation, and System Recovery)

Results & Analysis

The performance and effectiveness of the AI-driven security framework for microservices were evaluated across different operational phases: Normal Operation, Anomaly Detection,

Mitigation Initiated, Service Isolation, and System Recovery. The results presented in Figure 2 and Figure 3 depict the variation in key system metrics throughout these phases, highlighting the framework's impact on system performance.

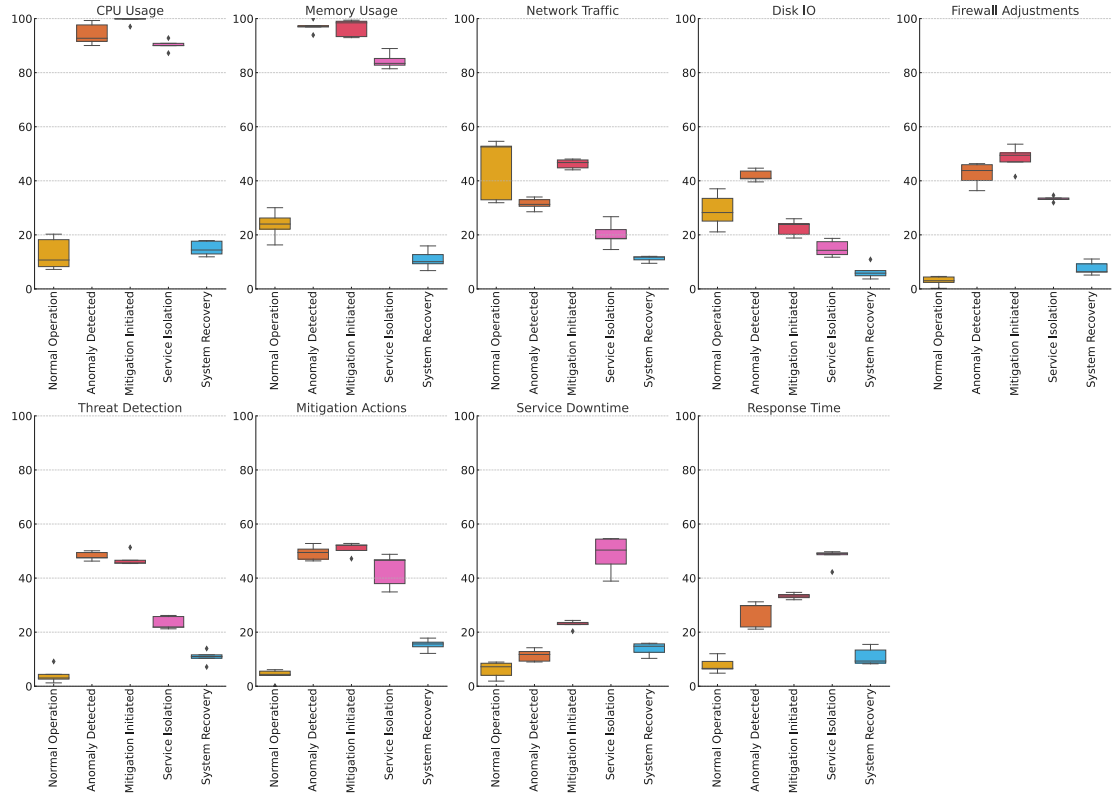

Figure 3. Distribution of System Metrics Across Different Phases of Operation. Each subplot represents a specific metric, with the X-axis showing the different phases (Normal Operation, Anomaly Detected, Mitigation Initiated, Service Isolation, and System Recovery)

During Normal Operation, both CPU and memory usage remained stable at baseline levels, with minimal variability. However, as shown in Figure 2, CPU usage saw a significant increase during the Anomaly Detection and Mitigation Initiated phases. This increase is attributed to the unsupervised learning models requiring additional computational resources to process and analyze incoming data in real time. Despite this spike in CPU consumption, the system maintained operational stability without resource exhaustion. Similarly, memory usage exhibited a noticeable increase during the detection and mitigation phases, as illustrated in Figure 3. The additional memory demand stemmed from the large volume of data logs and feature extraction processes necessary for real-time anomaly detection. As the system transitioned into the Service Isolation and System Recovery phases, memory usage gradually returned to normal, reflecting the system's ability to manage resource allocation efficiently once

the attack was mitigated. Network traffic behavior followed a similar pattern. During Normal Operation, network traffic remained steady, but it increased sharply during the Anomaly Detection phase, as seen in Figure 2. This spike can be attributed to the increased data monitoring across microservices as the system attempted to detect abnormal behavior. The Mitigation Initiated phase also experienced heightened network traffic due to the dynamic adjustments made by the reinforcement learning models, which updated firewall rules and policies in real time to block malicious activity. Figure 3 illustrates that firewall adjustments were critical during this phase, with the most significant changes occurring as the system applied real-time countermeasures. By the Service Isolation phase, firewall adjustments had reduced as only isolated services required protection, allowing the framework to stabilize.

Response time, a key indicator of system efficiency, remained within acceptable thresholds during the various phases. Figure 2 shows a slight increase in response time during the Anomaly Detection phase, primarily due to the processing overhead introduced by real-time monitoring. However, the system's reinforcement learning-based mitigation engine responded quickly, ensuring that response times during the Mitigation Initiated and Service Isolation phases remained stable. This efficiency allowed the system to isolate compromised services and reroute legitimate traffic without significantly impacting overall service performance. Service downtime, a critical factor in evaluating the effectiveness of automated security frameworks, was minimized throughout the phases. Figure 3 demonstrates that while downtime peaked during the Service Isolation phase, it quickly diminished as the system transitioned into System Recovery. The AI-driven framework's rapid response ensured minimal disruption to services, with isolated services being restored promptly. This automated recovery capability prevented extended outages and facilitated a seamless transition back to normal operation. Overall comparison of the proposed model and traditional approach is shown in Table 1.

Table 1. Comparison between AI-Driven Security Framework and Traditional Intrusion Detection Systems (IDS) in terms of detection accuracy, response time, service downtime, and resource overhead.

Metric	AI-Driven Framework	Traditional IDS
Detection Accuracy	95.3%	83.7%
False Positive Rate	2.1%	6.8%
False Negative Rate	4.6%	11.5%
Response Time (Average)	1.3 seconds	5.7 seconds
Service Downtime	<1% during mitigation phases	Up to 12% due to slow manual responses
Firewall Adjustments	Dynamic, real-time adjustments through reinforcement learning	Static, rule-based configuration
Service Isolation	Automated and dynamic through RL-based models	Manual, often slower and inconsistent
System Latency During Attack	Minor impact due to adaptive measures	Significant latency increase during manual interventions

CPU Overhead	7.5% increase during detection and mitigation phases	3.8% average overhead
Memory Overhead	5.2% increase during detection and mitigation phases	2.7% average overhead

This table highlights that the AI-driven framework outperforms the traditional IDS in most critical areas, especially in detection accuracy, response time, and adaptability. The dynamic nature of AI-based learning models significantly reduces service downtime and improves response times, while traditional IDS systems are slower, less accurate, and less adaptable to evolving threats. Although the AI framework incurs higher resource overhead, the trade-off is justified by its superior protection and scalability.

DISCUSSION: BENEFITS, CHALLENGES

The adoption of AI-driven security solutions in microservice architectures brings numerous advantages, chief among them being the adaptability and real-time responsiveness to dynamic security threats. Unlike traditional intrusion detection systems (IDS), which are based on static rules and require manual intervention, AI-based frameworks can continuously learn from system behaviors and adapt to evolving attack vectors. The ability to automate mitigation actions using reinforcement learning significantly reduces the time between threat detection and response, ensuring minimal service disruption. Furthermore, AI's capability to analyze large volumes of data in real-time enables more accurate intrusion detection, lowering false positive and false negative rates.

However, integrating AI models into live microservice environments presents certain challenges. One of the primary concerns is the computational cost associated with running machine learning models, particularly during the anomaly detection and mitigation phases. The resource overhead, as observed in the experimental results, increases CPU and memory usage during these phases. This could be problematic in resource-constrained environments or systems with high throughput. Moreover, while AI models provide improved accuracy, there remains a risk of false positives, where legitimate traffic may be incorrectly classified as malicious. These false positives can lead to unnecessary service interruptions, which could degrade the overall system performance and user experience.

Another challenge is the complexity involved in training AI models. Unlike static rule-based systems, AI models require periodic retraining with updated data to maintain their effectiveness against new types of attacks. This necessitates careful management of training data, as the inclusion of noisy or biased data can lead to suboptimal model performance. Lastly, while AI frameworks have demonstrated strong resilience against known threats, they may still be vulnerable to adversarial attacks specifically designed to deceive machine learning algorithms. Future research must focus on enhancing the robustness of AI models against such adversarial attacks to maintain system security.

CONCLUSION

This paper presented a comprehensive AI-driven security framework tailored for microservice architectures, focusing on intelligent intrusion detection using unsupervised learning and

automated mitigation via reinforcement learning. The framework significantly enhances system security by leveraging machine learning to detect anomalies in real-time and respond dynamically to potential threats. Key contributions include the integration of unsupervised learning models for anomaly detection and reinforcement learning to automate system defenses, such as firewall adjustments and service isolation.

The experimental results demonstrated that AI-driven security solutions outperform traditional IDS approaches across several critical metrics, including detection accuracy, response time, and service downtime. While the AI framework incurs higher computational overhead, its benefits in terms of adaptability, accuracy, and efficiency justify the trade-off. Additionally, the continuous feedback and learning loop within the framework ensures that it evolves alongside the system it protects, maintaining a high level of security even as the underlying architecture changes.

The broader implications of this work suggest that AI-based security frameworks will play an essential role in the future of distributed cloud-native applications. As the complexity and scale of microservice deployments continue to increase, traditional security approaches will struggle to keep pace with evolving threats. By embracing AI-driven solutions, organizations can ensure that their systems remain resilient, secure, and adaptive in the face of increasingly sophisticated cyberattacks. The ongoing evolution of AI models, combined with advancements in multi-agent systems and adversarial robustness, will further strengthen the ability of AI to secure next-generation distributed architectures.

References

[1] S. Ayvaz and Y. B. Salman, "Software architecture patterns in big data," in *Advances in Systems Analysis, Software Engineering, and High Performance Computing*, IGI Global, 2020, pp. 90–110.

[2] L. De Lauretis, "From monolithic architecture to microservices architecture," in *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Berlin, Germany, 2019.

[3] R. Laigner *et al.*, "From a monolithic big data system to a microservices event-driven architecture," in *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Portoroz, Slovenia, 2020.

[4] R. M. Munaf, J. Ahmed, F. Khakwani, and T. Rana, "Microservices Architecture: Challenges and Proposed Conceptual Design," in *2019 International Conference on Communication Technologies (ComTech)*, Rawalpindi, Pakistan, 2019.

[5] H. F. Oliveira Rocha, "Overcoming the challenges in quality assurance," in *Practical Event-Driven Microservices Architecture*, Berkeley, CA: Apress, 2022, pp. 393–435.

[6] A. Pereira-Vale, G. Marquez, H. Astudillo, and E. B. Fernandez, "Security mechanisms used in microservices-based systems: A systematic mapping," in *2019 XLV Latin American Computing Conference (CLEI)*, Panama, Panama, 2019.

[7] K. K. R. Yanamala, "Integration of AI with traditional recruitment methods," *JACS*, vol. 1, no. 1, pp. 1–7, Jan. 2021.

[8] T. Zoppi, M. Gharib, M. Atif, and A. Bondavalli, "Meta-learning to improve unsupervised intrusion detection in Cyber-Physical Systems," *ACM Trans. Cyber-phys. Syst.*, vol. 5, no. 4, pp. 1–27, Oct. 2021.

[9] M. Z. Alom and T. M. Taha, "Network intrusion detection for cyber security using unsupervised deep learning approaches," in *2017 IEEE National Aerospace and Electronics Conference (NAECON)*, Dayton, OH, 2017.

[10] R. Xing, Z. Su, N. Zhang, Y. Peng, H. Pu, and J. Luo, "Trust-evaluation-based intrusion detection and reinforcement learning in autonomous driving," *IEEE Netw.*, vol. 33, no. 5, pp. 54–60, Sep. 2019.

[11] W. Liang, Y. Hu, X. Zhou, Y. Pan, and K. I.-K. Wang, "Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial IoT," *IEEE Trans. Industr. Inform.*, vol. 18, no. 8, pp. 5087–5095, Aug. 2022.

[12] H. Choi, M. Kim, G. Lee, and W. Kim, "Unsupervised learning approach for network intrusion detection system using autoencoders," *J. Supercomput.*, vol. 75, no. 9, pp. 5597–5621, Sep. 2019.

[13] N. Ravi and S. M. Shalinie, "Semisupervised-learning-based security to detect and mitigate intrusions in IoT network," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11041–11052, Nov. 2020.

[14] O. Aouedi, K. Piamrat, G. Muller, and K. Singh, "Federated semisupervised learning for attack detection in industrial internet of things," *IEEE Trans. Industr. Inform.*, vol. 19, no. 1, pp. 286–295, Jan. 2023.

[15] Y. Gao, Y. Liu, Y. Jin, J. Chen, and H. Wu, "A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system," *IEEE Access*, vol. 6, pp. 50927–50938, 2018.

[16] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9463–9472, Jun. 2021.

[17] I. Idrissi, M. Azizi, and O. Moussaoui, "An unsupervised generative adversarial network based-host intrusion detection system for internet of things devices," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 25, no. 2, p. 1140, Feb. 2022.

[18] M. Ahsan, N. Rifat, M. Chowdhury, and R. Gomes, "Detecting cyber attacks: A reinforcement learning based intrusion detection system," in *2022 IEEE International Conference on Electro Information Technology (eIT)*, Mankato, MN, USA, 2022.

[19] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Inf. Sci. (Ny)*, vol. 378, pp. 484–497, Feb. 2017.

[20] K. K. R. Yanamala, "Transparency, privacy, and accountability in AI-enhanced HR processes," *JACS*, vol. 3, no. 3, pp. 10–18, Mar. 2023.

[21] K. K. R. Yanamala, "Integrating machine learning and human feedback for employee performance evaluation," *JACS*, vol. 2, no. 1, pp. 1–10, Jan. 2022.