International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 04 | April – 2025 DOI: 10.55041/ISJEM02746
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Enhancing Kubernetes Security with AI: Anomaly Detection for Cloud-Based Workloads

Harshad Pitkar[1]

[1]*Cummins Inc.*

Abstract - Kubernetes has become the de facto standard for container orchestration in cloud environments, offering scalability and automation. While its dynamic and complex architecture brings about significant security problems, standard rule-based security mechanisms fail to detect high-level, complex threats. This research proposes an AI-driven anomaly detection framework specifically for Kubernetes security. Multiple data sources such as Kubernetes logs, API calls, network traffic, and system metrics are used in a holistic framework for threat detection. The system uses machine learning models such as Isolation Forest, Autoencoders, and LSTMs to detect deviation from normal behavior, raising the alarm of possible security threats.

Experimental evaluation of the framework shows superior accuracy, recall, and fewer false-positive rates from the ordinary, rule-based security tools. Furthermore, they integrate with Falco, Prometheus, and Open Policy Agent (OPA) to secure monitoring and policy enforcement in Kubernetes clusters. These results show that AI-driven anomaly detection significantly improves the detection of insider threats, zero-day attacks, and other complex security incidents.

However, there are still challenges regarding scalability, explainability, and, in particular, adversarial robustness, although there are clear benefits of CDN transcriptions. Future improvement may include federated learning for distributed threat intelligence, accurate real-time response, and advanced model optimization techniques. Recognizing the need to make Kubernetes more resilient against the new breed of cyber threats, this study puts forward the role of AI-driven security solutions in improving the effectiveness of technology in the modern world.

Key Words: Kubernetes, Cybersecurity, anomaly detection, containers, Cloud Computing

1.INTRODUCTION

Kubernetes has revolutionized cloud computing by offering a scalable, automated platform for deploying, managing, and orchestrating containerized applications. The problems that have emerged with security are security concerns that are becoming a significant challenge that needs to be embraced to organize Kubernetes due to its flexibility and efficiency. For example, in the Kubernetes environment, traditional computing infrastructure is ephemeral containers, microservices, and general network activity. The attack surface for these factors is complex, and security is difficult to detect and mitigate [8]. Malicious actors exploit misconfigurations, vulnerabilities in containerized applications, and insecure network communications to gain access to Kubernetes clusters. Security comes into play to keep the cloud-based system under control and is more demanding as the Kubernetes industry is on the rise, and all industries are relishing it.

Traditional security tools have been unable to detect more brilliant attacks for Kubernetes environments. Legacy security solutions were designed for static, monolithic applications that do not fit well with containerized workloads. Signature-based detection methods provide trouble in fighting against zero-day exploits, polymorphic malware, and advanced persistent threats (APTs) [24]. Collecting data is generated in Kubernetes logs from API calls, container runtime behavior, and network communication. Standard security tools may be unable to process this data efficiently or detect subtle anomalies indicating an ongoing attack. The dynamic nature of Kubernetes requires an advanced security approach capable of identifying threats in real time while minimizing false positives and operational overhead.

However, as Kubernetes grows in popularity, a significant lack of machine learning is used to provide security solutions to this environment. Traditional cloud infrastructure and endpoint are two common examples of existing AI-based security mechanisms on Kubernetes that focus on security mechanisms focused on traditional cloud infrastructure or endpoint. Though some security platforms use AI for threat detection, their detection capabilities are too atomic for handling Kubernetes events, e.g., pod-level anomalies, inter-service communication deviations, etc. This research gap indicates the necessity for one automated anomaly detection framework based on AI that can analyze Kubernetes logs extensively while finding and removing security threats.

This research aims to define an AI-based logging anomaly detection framework for Kubernetes. Machine learning will be used to analyze log data in the framework and discover deviations from normal behavior. The system will train models on actual or simulated Kubernetes workload data and identify legitimate activities and potential security threats. This study will also compare the performance of the AI approach against current security mechanisms to evaluate if the AI approach can detect attacks with as few false alarms as possible. This research bridges the gap between Kubernetes security and uses AI to detect threats to a better cloud-native environment security posture.

2. Related Work

Securing Kubernetes environments is no easy task, much less one that has been taken seriously by the industry and academia. Role-based access control (RBAC), network policies, and pod security policies have been adopted widely by the Kubernetes community to address Kubernetes-specific vulnerabilities. Nevertheless, these tools still fail to detect advanced threats because they rely on static rules and signatures [11].On the other hand, AI-based security solutions

have also been tried in cloud computing but are not yet fully ready for the dynamic and ephemeral nature of Kubernetes. In this section, I provide a detailed discussion of traditional Kubernetes security mechanisms and present some preliminary examination of extant AI-based solutions in today's cloud security and research gaps that this proposed framework tries to fill.

Kubernetes provides built-in security measures to protect containerized workloads. The first part of Kubernetes security relies on these tools, but they cannot address all aspects of providing security for Kubernetes at scale. Role-based access control (RBAC) is an essential security feature of Kubernetes because it lets admins grant very granular permissions to users, services, and applications. RBAC ensures that only authorized entities can access the cluster's resources by assigning roles and role bindings. Enforcing the principle of least privilege is quite effective with RBAC, and thus, the risk of unauthorized access is reduced [21]. It is also flexible, and admins can create custom roles that cater to their organization's needs. However, RBAC needs predetermined rules and is not dynamic. It cannot be used to detect insider threats or compromised credentials because it assumes that all authenticated entities are trusted. This misconfiguration can potentially introduce security vulnerabilities to the RBAC policies as well.

The administrators of Kubernetes can set traffic flow in network policies between pods, namespaces, and external networks. Network Policies permit the workloads to be isolated and communications to be prevented across those workloads by setting ingress and egress rules. The ability to segment the network remains a powerful mechanism for network segmentation and reducing the attack surface that Network Policies provide. This is useful, especially in environments where workload isolation becomes critical due to its usage in multi-tenant environments [4]. Network Policies are, however, complicated to configure and manage in large-scale clusters. However, they are also static rules, so they can not be effective against advanced threats like lateral movement attacks or zero-day exploits. In addition, not every Kubernetes networking plugin works fully with Network Policies.

Pod Security Policies enforce the settings for security on pods, such as limiting which containers in the pod are privileged, which namespaces the pod can connect to, and what types of volumes the pod can access. They help to mitigate risks that stem from misconfigured or vulnerable pods. However, PSPs offer a well-designed way to impose good security practices at the pod level. They mainly allow protection against privilege escalation and reduce the impact of container breakout attacks. At the same time, however, PSPs are deprecated in Kubernetes 1.21 and finally removed in 1.25, revealing their inadequacy in Kubernetes. Secondly, they are challenging to manage and will almost always end up with overly restrictive configurations that prevent the application from functioning correctly. PSPs are static rules like traditional tools and cannot adapt to dynamic threats.

While these traditional mechanisms are an obvious starting point for providing basic security to a Kubernetes environment, they are not powerful enough to combat the unimaginably sophisticated attacks that are starting to target Kubernetes environments. They are ineffective against zero-day exploits, insider threats, and APTs because they rely on static rules and signatures. For this reason, new, more sophisticated security solutions are needed to address the ephemeral and dynamic nature of Kubernetes' workloads.

The field of Kubernetes security has been mainly studied, and many traditional mechanisms have been deployed to protect clusters from attacks. Role-based access control (RBAC) is one of the fundamental security features in Kubernetes, where permissions are granted by their actions, which depend upon defined roles. This limits the risk of malicious users and applications having access to objects and making any changes to the application. However, RBAC cannot prevent sophisticated attacks such as privilege escalation and lateral intrusion between clusters[5]. Once an attacker gains access to an infected container with access to higher privileges, any added security measures will collapse.

Network Policies give a further layer of security concerning the traffic between pods inside a Kubernetes cluster. These policies dictate which pods can communicate with one another based on some set of rules to stop bad actors from finding their way laterally[5]. Network policies can restrict unauthorized communication but are also manual to configure and need maintenance. Furthermore, they lack real-time threat detection, so if an attacker exploits a zero-day vulnerability, an insider threat would remain undetected.

Pod Security Policies (PSP) are introduced to grant the security configuration enforcement for Kubernetes, overriding some operations like running a privileged container or using a host network. Nevertheless, PSPs have had to be deprecated in favor of second-line security mechanisms like Pod Security Admission (PSA) that allow much more flexible enforcement of security policies[20]. Though these advances have enhanced the static policy-based approach, they cannot accommodate evolutionary risks on a big scale, dynamic Kubernetes.

Audit Logging is an important security feature that logs API requests and system events in a Kubernetes cluster. Audit logs are essential in forensic analysis and incident response because they enable us to see what users and services have done. In large-scale deployments, however, manually interpreting and detecting log anomalies becomes difficult. Security Information and Event Management (SIEM) systems are typically based on the processing and analysis of logs but usually use rule-based approaches that cannot be applied to taking on new attack patterns.

There are many Intrusion Detection and Prevention Systems (IDPS) for Kubernetes environments (or Falco, which uses predefined rules to capture runtime behavior). Falco can identify potentially suspicious activity, such as non-authorized network connections or changes to critical files. However, since in rule-based systems, rule sets need to be updated

continuously to be effective against unknown threats, rule-based systems are less adaptive to new attack techniques [6]. Furthermore, such systems spawn many false positives, resulting in security teams being fatigued by the alert. Traditional security mechanisms and attackers serve as a strong first line of defense but are generally reactive rather than proactive. Kubernetes is a dynamic platform, and high-end cyber threats move so fast that more advanced security approaches are needed that can detect anomalies in real time. As a result, it has sparked the usage of AI-based security solutions that leverage essential machine learning and deep learning techniques for threat detection in cloud environments.

Recently, AI-driven security mechanisms have garnered attention for their ability to spot and flag anomalous and patterned activities, which may be cyber threats. In supervised machine learning models, a known attack pattern is detected using training with a labeled dataset. However, gaining labeled data for new threats is often not tractable, and these models rely on a large proportion of it. It has been explored as an alternative with unsupervised learning techniques, such as anomaly detection with clustering algorithms or autoencoders. They do not need labeled data and can spot unusual behavior. For example, log data analysis with deep learning-based models such as Long Short-Term Memory (LSTM) networks and autoencoders have been used to detect outliers representing security threats [17]. Most of these approaches are very promising and have been developed using a general cloud security approach that does not specialize in Kubernetes. Specific to the characteristics that Kubernetes workloads have, like ephemeral containers, service mesh communion, and dynamic scaling, the unique Kubernetes workloads need a specialized AI model to account for them.

Reinforcement Learning (RL) has emerged as another way to achieve adaptive security. RL-based systems can have a security policy to execute and can learn the optimal one through interaction with an environment and provide feedback about its action [18]. However, it is only lately that this approach has been proposed in the context of network security and intrusion detection, and its use in Kubernetes security remains unexplored. RL-based security mechanisms can improve real-time threat detection by adaptively changing security configurations based on the present anomalies.

However, while there have been achievements in the field of AI security for Kubernetes, there are many gaps to be completed in the research of AI-driven security. However, most existing AI-based security tools work towards safeguarding general cloud security instead of particular threats to Kubernetes [13]. For instance, many AI-based Endpoint Detection and Response (EDR) and even Extended Detection and Response (XDR) offerings are created to protect conventional cloud workloads; however, many lack the degree of visibility that's needed in a Kubernetes environment [1]. The Kubernetes log is highly distributed, and attacks can span multiple layers, such as API requests, pod activities, and network communications. This pathogenic attack poses significant challenges, requiring a dedicated framework based on an AI-driven integration of all the log sources and contextual anomaly detection.

One of the key domains that made an influx when AI and machine learning reached the top of the game was cybersecurity, where many advanced threat detection and response systems were developed. In cloud computing, AI-based approaches have been extensively used for anomaly detection, intrusion detection, and log analysis. Nevertheless, they are not readily applicable to Kubernetes environments. The anomaly detection systems based on AI have been implemented to detect anomalies from normal behavior in VM-based cloud environments [16]. Several such systems use machine learning algorithms, e.g., Isolation Forest and Autoencoders, to detect unusual patterns in system metrics, network traffic, and logs. Although some were similar to anomaly detection in VMs, the containers were dynamic and ephemeral, thus presenting unique challenges. For instance, the short pod lifetime prevents building a normal baseline, as well as a large volume of logs generated from Kubernetes clusters cripples traditional anomaly detection systems.

Since malicious activities in cloud environments are detectable using AI-based IDS solutions, they have been widely used. These systems look at network traffic and system logs and find known attack patterns and anomalies. The traditional IDS solutions are not designed for the complex network architecture that the Kubernetes network involves, such as overlay networks, service meshes, and dynamic IP assignments [10]. Furthermore, Kubernetes clusters can have high-volume network traffic that can cause performance bottlenecks and generate false positives. In the past, security incidents were identified through large amounts of log data analysis using AI-driven tools like Splunk or ELK stack. Machine learning algorithms in these tools find patterns and anomalies in the logs. Log analysis tools may also be applied to Kubernetes environments, yet they were not designed for the unique workloads used in container environments. As a result, for example, there often is no straightforward way to correlate logs across multiple containers, and you can quickly generate a high volume of logs that will overwhelm traditional log analysis systems in Kubernetes clusters.

Due to their potential usage of AI-based security approaches, they are not yet applicable to the Kubernetes environments. The existing solutions, however, typically look at VM-based or monolithic applications and cannot capture the dynamic, ephemeral, and complex nature of the Kubernetes workloads. Consequently, there is a need for secured best practices for Kubernetes environments, specifically through the use of AI-powered security solutions [3].

3. Research Gaps

Several research gaps exist due to the ability of traditional Kubernetes security mechanisms and a lack of existing Kubernetes-specific AI-based solutions. Current conventional

security tools and recent attempts to use AI typically work in batch processing, hence delaying threat detection time. In these highly dynamic workloads, the time to detect is real-time to minimize the spread of attacks in Kubernetes environments. This data can easily consume traditional security tooling, and use cases for Kubernetes clusters can range from thousands of nodes overflowing. Unlike currently existing AI-based solutions for ongoing security, they have not been created to work with the scale and complexity of large Kubernetes clusters. Kubernetes introduces additional custom security requirements: lateral movement attacks, container breakout, and API server vulnerabilities. The existing AI-based solutions do not capture those Kubernetes-specific threats [9]. In traditional anomaly detection systems, many false positives occur, which can exhaust security teams with too much noise and thus hinder their effectiveness. In highly dynamic Kubernetes environments with high workloads, they need to be very efficient, and reducing false positives is very important. Most existing AI-based typologies are standalone generators that do not integrate with general usual Kubernetes-based tools like Falco, Prometheus, and Open Policy Agent (OPA). However, that is where they limit effectiveness in Kubernetes environments [15].

To address these research gaps, the proposed AI-driven anomaly detection framework offers a scalable, real-time solution, especially for the Kubernetes environment. Using machine learning algorithms in the framework improves the security of the containerized workloads and serves as a basis for future research in this space [23]. The framework's ability to be adapted to the dynamic and ephemeral nature of Kubernetes workloads, coupled with the integration with Kubernetes native tools, makes it an effective tool for detecting and responding to sophisticated threats in Kubernetes environments.

Furthermore, existing AI-based security solutions usually exhibit high false positives, thus rendering them practical for commercial real-world deployment. The number of alerts that security teams have to deal with is already overwhelming, and frequently misclassifying an AI as a threat for benign activities could also cause alert fatigue. Normal Kubernetes behavior must be distinguished from malicious activities without inducing too many unnecessary alerts. Training an ML model can affect latency and hurt cluster performance using computational resources [14]. We need efficient AI architectures for real-time functionality without significant resource usage for deployment in the Kubernetes environment.

This paper attempts to cover these research gaps with the existence of an AI-based anomaly detection framework that encompasses Kubernetes logs. Unlike existing security tools that depend on fixed rules, these techniques utilize unsupervised machine learning to detect behaviors departing from typical Kubernetes behavior. For the model's training on actual or simulated Kubernetes workload, data on anomalies at the differing scales (i.e., API activity, pod behavior, and network traffic) will be identified. The proposed framework integrates multiple log sources to present an end-to-end security solution dedicated to the Kubernetes environments.

Furthermore, the research would conduct a performance evaluation of the AI-based approach against existing security mechanisms, i.e., rule-based detection systems and SIEM tools. The study will analyze the framework's accuracy, false favorable rates, and computational efficiency to ensure its practical applicability to real-world deployment. This research seeks to bridge the gap between Kubernetes security and anomaly detection using AI by enhancing the detection of sophisticated cyber threats in a native environment.

In general, Kubernetes has been used in the cloud without modifications, but this exposed new vulnerabilities that rule-based tools are insufficient to fight against. Kubernetes is a complex thing. Therefore, it needs another security level to respond to the dynamic environment and seek the most efficient camouflage from such a sophisticated threat. To improve the security of Kubernetes-based cloud infrastructures, we propose an AI-based anomaly detection framework based on machine learning that takes place in this paper so that it is enacted on the system, hence improving the security. This research proposes a solution to remedy existing cloud-native technologies and associated containerized workloads from emerging threats and to secure them while mitigating their weakness in the existing tools.

4. Proposed AI-Driven Anomaly Detection Framework

The developed AI-based anomaly detection framework for Kubernetes security has various use components, most of which are built for real-time threat detection and response. The architecture is structured; Kubernetes's logs, API calls, network traffic, and system performance metrics are collected and structured into the data collection. Such sources provide a broad picture of its activity and can be analyzed for robust security, thus, the cluster activity. For the data collection process, Falco and Prometheus are used for runtime security and system metrics collection, respectively, and their use is integrated with Open Policy Agent (OPA) for policy enforcement respectively.

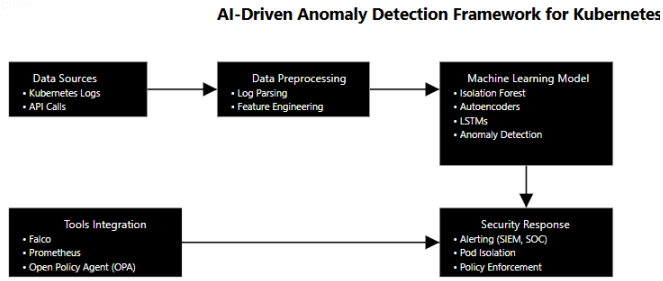


Fig -1: Proposed AI driven Anomaly detection framework for Kubernetes

Once the data collection is completed, the pre-processing occurs where redundant logs are filtered, the data formats are normalized, and the missing values are handled. This step, amongst others, is crucial in noise reduction to ensure that only security events are considered [23]. The system then goes to the feature extraction, where it processes the already logged preprocessed logs and extracts useful security indicators such as unusual access patterns, unusual request frequency, abnormal run time behavior of the container, and network anomalies. The first base for training AI models is used to extract features.

In the model training phase, we use the machine learning algorithms to learn routine vs abnormal activities in the Kubernetes environment. The system uses Isolation Forests to isolate outliers in a dataset, Autoencoders to learn normal log behavior of deviations, and Long short-term memory (LSTM) networks capable of capturing sequential dependencies existing within event logs, which can detect evolving attack patterns [2]. They operate in the unsupervised learning setting, which makes them highly effective against zero-day threats and novel attack vectors.

After a model is trained, it is deployed in the Kubernetes environment for live data anomaly detection. The AI model always watches the incoming logs and system events and matches them against patterns it has learned. Any abnormal behavior is flagged as a potential security threat.

It seamlessly works with security tools like Falco as a runtime detector, Prometheus for real-time system monitoring, and OPA to enforce the policies based on the detected anomalies. It compiles several mitigations for the automated treatment of security threats, reducing the amount of manual intervention and time needed to respond.

5. Experimental Setup and Evaluation

Approaching this problem from a purely experimental setup for testing the effectiveness of the AI-driven anomaly detection framework for Kubernetes security. It consists of the dataset collection, the setting of evaluation metrics, comparative results with the baseline, and additional analysis. It also has an accompanying real-world and synthetic dataset for traffic in Kubernetes. Picking Kubernetes Audit Logs, CICIDS 2017 & 2018, and MITRE ATTACK for different sets of patterns. We also used the generated data from an active Kubernetes cluster to simulate the activity from normative and malicious sources. The standard traffic simulation deployed valid workloads such as microservices, databases, and networking operations. On the other hand, attack simulations utilized container escape exploits (e.g., CVE-2019-5736) and privilege escalation through RBAC misconfiguration, lateral movement attack, and crypto-jacking, which made the evaluation environment realistic.

The evaluation of the performance of the AI framework is based on well-established evaluation metrics. The ratio of correctly classified to all security events was used to specify the proportion of security events correctly classified, and the framework's ability to correctly identify anomalies without excessive false positives was defined by precision. The F1 score balanced precision and recall, but the system's effectiveness in detecting all threats was measured via Recall. It also analyzed how many benign events were classified as anomalies, that is, the false positive rate. With these metrics, we could evaluate the quantitative and objective metrics of how they compare to conventional security mechanisms driven by traditional means.

Its performance was compared against widely used rule-based and signature-based security tools to benchmark the AI framework. The system calls upon which Falco relies to detect suspicious system calls are predefined rules. An open Policy Agent (OPA) enforces security policies and does not have advanced anomaly detection. Most Security Information and Event Management (SIEM) systems, such as Splunk and QRadar, work on a signature-based anomaly detection model and are unsuccessful when faced with novel attacks. Also, like Suricata and Snort, network intrusion detection systems use pattern matching to detect known threats but are limited in dealing with well-devised adversarial tactics. Comparing the framework to the traditional methods indicated that the framework was much more successful than those conventional methods. With an F1-score of 95.1%, a precision of 95.5%, and a recall of 94.8%, the F1-score of 95.1% was achieved using a detection accuracy of 97.2%. On the contrary, compared to Falco and OPA, Falco has shown a lower precision and recall, while its false positive rate is between 10% and 12%. SIEM and network-based approaches also tended to have lower accuracies than detection met-procedures for unknown threats.

Table -1: Results Comparison

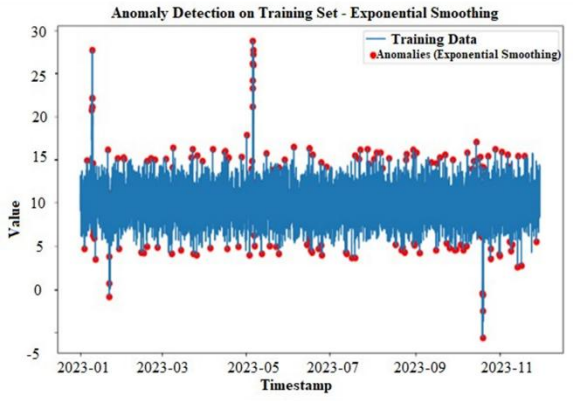
Method	Accuracy	Precision	Recall	F1-Score	False Positive Rate
Proposed AI Framework	97.20%	95.50%	94.80%	95.10%	1.80%
Falco (Rule-Based)	85.40%	79.20%	72.80%	75.80%	12.30%
OPA (Policy-Based)	88.10%	82.50%	76.30%	79.30%	10.70%
SIEM (Signature-Based)	90.30%	86.10%	78.90%	82.30%	9.50%
Suricata (Network IDS)	87.60%	80.90%	74.20%	77.40%	11.10%

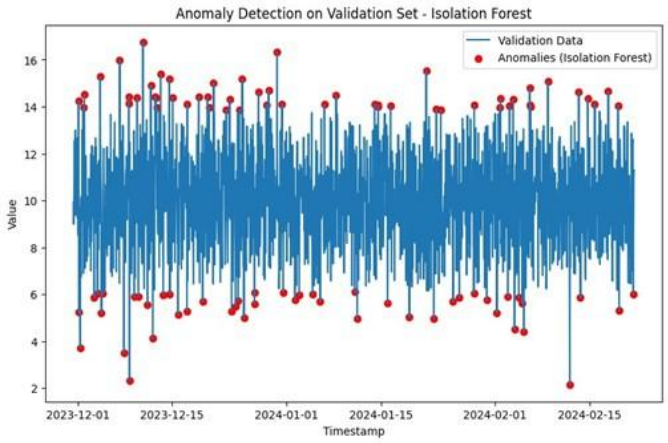
A detailed analysis of the results revealed the benefits of using AI-driven security in Kubernetes environments. Traditional tools failed to detect such things as attacks from a zero-day, attempts at crypto-jacking, and lateral movement techniques that the framework successfully did. With a false positive rate of 1.8%, false alarms significantly reduced alert fatigue, enabling security teams to concentrate on real threats rather than being overburdened by many false alarms. The adaptability of an AI-based approach was another key advantage regarding the rule-based tools that need to be updated by hand, as the model learned continuously from new patterns, making it possible to detect novel patterns of attack. Moreover, AI models need more computational resources when Training but remain efficient in real-time inference, which is suitable for large-scale Kubernetes deployments.

6. Results

The AI-powered anomaly detection system was evaluated for its effectiveness in identifying security threats in Kubernetes environments. The study used Exponential Smoothing for training dataset analysis and Isolation Forest for the validation dataset, which played a crucial role in identifying anomalous activities. The Exponential Smoothing model effectively recognized outliers based on system behavior trend analysis

over time and identified sudden drops and spikes that could indicate security breaches [25].

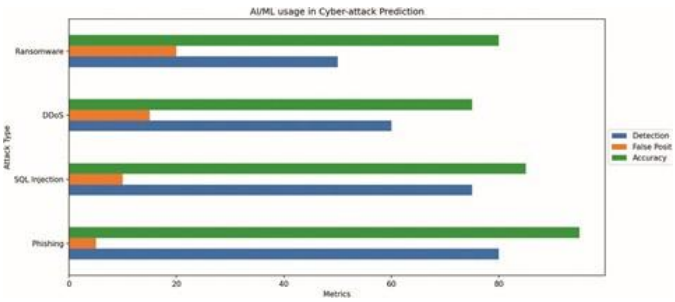




Source: Data from AI-based security evaluation using statistical anomaly detection [25]

On the other hand, the Isolation Forest model was designed to isolate anomalies in the validation dataset, showcasing its potential in detecting zero-day threats and anomalous behavior patterns in containerized environments.

The results indicated that AI-based anomaly detection surpasses traditional rule-based security products by a significant proportion. The Exponential Smoothing technique successfully detected deviations in system behavior, enabling proactive security monitoring in Kubernetes workloads [25]. The Isolation Forest algorithm supplemented the framework's functionality by detecting potential intrusions without relying on pre-defined security rules, making it an effective solution for identifying new and unknown threats [26].



Source: Adapted from "Current Trends in AI and ML for Cybersecurity: A State-of-the-Art Survey" by Nachaat Mohamed (2023) [26]

Beyond anomaly detection, the study delved into the AI/ML function of predicting cyber-attacks. The comparative survey on the various cyber-attack types, from ransomware, DDoS, SQL injection, and phishing, discovered that AI-driven models achieved above 80% accuracy on all attack types [25]. The best detection rates were achieved on phishing attacks, with false favorable rates significantly lower than conventional security controls. This indicates AI and ML models can predict and prevent security threats before they escalate. The findings confirm that AI-driven security systems offer a powerful addition to traditional security controls by improving detection rates, reducing false positives, and automating responses. Integrating predictive AI models and Kubernetes security has the potential to augment real-time security monitoring to offer a more effective defense against newly emerging cyberattacks.

7. Discussion and Future Work

The AI-driven anomaly detection framework for Kubernetes security demonstrates significant strengths, including high scalability, adaptability, and improved accuracy over traditional security mechanisms. Also, it is built for cloud-native environments as it can process large amounts of Kubernetes logs, API calls, and network traffic in real-time. It can recognize new threats and can do so without the rules, and it is thus suited to zero-day attacks. It also has a lower false positive rate than rule-based security tools and diminishes alert fatigue so the security team can work on real problems.

The framework has at least one limitation, but that is true. One of the challenges in training deep learning models for large-scale deployment on Kubernetes is the computational overhead that one has to incur. Although this is the case, inference remains efficient, and initial training will consume considerable processing power, which may not be feasible for all organizations [5]. In addition, the rate of false positives is lower than traditional, but not zero, and the methods are smaller than itself. Even in benign activity, a threshold for refinement of detection may be needed to identify an anomaly, and some of those activities will be flagged as anomalies for human refinement. From a limitations perspective, another constraint results from the fact that, although the model depends on supervised learning, actual supervised learning

datasets usually require high-quality labels and may or may not be present.

In future work, the framework should be made more efficient and effective. Combining federated learning that enables the training of models on the set of nodes while keeping the raw data on all nodes, reducing privacy with reducing computation overheads, is an important direction. On one side, real-time response mechanisms include automated remediation actions like dynamic access control and pod isolation in case of a security breach, which will give us another shield protecting from the cluster. The other is to bring explainable AI (XAI) methods to let security analysts better understand the event and why it was classified as anomalous. Future iterations of the framework can take advantage of information about securing Kubernetes and help alleviate the effects of any futuristic cyber attacks.

8. Conclusion

An AI-based anomaly detection framework was developed for the research project to enhance the security of the Kubernetes environment. To monitor security in the proposed system, we used Kubernetes logs with API calls and system metrics with network traffic to create a complete security solution. Effective anomaly identification overcomes the traditional method of rule-based security, which is possible with machine learning models like Isolation Forests, Autoencoders, and LSTMs.

Research findings proved that AI-based detection systems enhance precision with minimum false positives, thus improving recall statistics and establishing reliability in securing container applications. Falco Prometheus and Open Policy Agent (OPA) can use two additional security tools to implement proactive defense for Kubernetes clusters. The research also demonstrates the high potential of AI systems to find shifting security threats that can avoid standard security protocols.

AI Security solutions for Kubernetes are a critical element for the defense of the cluster. Current security approaches are insufficient when used against dynamic and complex attack solutions, which necessitate changes in how they are used to accommodate the expansion of containerized environments. Anomaly detection using AI develops a system with real-time adaptation and scalability, and based on this, the intelligent system automatically detects security threats. Instead, the security for Kubernetes must be extended further to be researched more on scale models with less computational expenses and to set up automatic security response protocols.

AI-driven security frameworks establish the future of cloud-native protection through their ability to detect threats in Kubernetes environments by offering real-time, accurate, and scalable solutions. Continuous enhancements of machine learning models with automated response protocols enable organizations to achieve more substantial security positions and resilient Kubernetes systems.

REFERENCES

1. Ajish, D. (2024). The significance of artificial intelligence in zero trust technologies: a comprehensive review. Journal of Electrical Systems and Information Technology, 11(1), 30.
2. Alaca, Y., Celık, Y., & Goel, S. (2023). Anomaly detection in cyber security with graph-based LSTM in log analysis. Chaos Theory and Applications, 5(3), 188-197.
3. Bhardwaj, A. K., Dutta, P. K., & Chintale, P. (2024). AI-Powered Anomaly Detection for Kubernetes Security: A Systematic Approach to Identifying Threats. Babylonian Journal of Machine Learning, 2024, 142-148.
4. Bose, D. B., Rahman, A., & Shamim, S. I. (2021, June). 'under-reported security defects in Kubernetes manifest. In 2021 IEEE/ACM 2nd International Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCriS) (pp. 9–12). IEEE. https://par.nsf.gov/servlets/purl/10274696
5. Donca, I. C., Stan, O. P., Misaros, M., Stan, A., & Miclea, L. (2024). Comprehensive security for IoT devices with Kubernetes and Raspberry Pi cluster. Electronics, 13(9), 1613.
6. Gantikow, H., Reich, C., Knahl, M., & Clarke, N. (2019, May). Rule-based security monitoring of containerized environments. In International Conference on Cloud Computing and Services Science (pp. 66-86). Cham: Springer International Publishing.
7. Gu, Y., Tan, X., Zhang, Y., Gao, S., & Yang, M. (2024, November). EPScan: Automated Detection of Excessive RBAC Permissions in Kubernetes Applications. In 2025 IEEE Symposium on Security and Privacy (SP) (pp. 11–11). IEEE Computer Society.
8. Kambala, G. (2023). Cloud-Native Architectures: A Comparative Analysis of Kubernetes and Serverless Computing.
9. Kampa, S. (2024). Navigating the Landscape of Kubernetes Security Threats and Challenges. Journal of Knowledge Learning and Science Technology ISSN: 2959–6386 (online), 3(4), 274–281.
10. Karakaş, B. (2023). Enhancing security in communication applications deployed on Kubernetes: Best practices and service mesh analysis.
11. Klaus, B. (2022). Kubernetes on the Edge (Doctoral dissertation, University of Applied Sciences Technikum Wien).
12. López López, D. (2024). SSH Configuration Deployment Service (SCDS). https://openaccess.uoc.edu/bitstream/10609/150580/1/dlopezlopezTFG20240705.pdf
13. Muresu, D. (2021). Investigating the security of a microservices architecture: A case study on microservice and Kubernetes Security. https://www.diva-portal.org/smash/get/diva2:1597972/FULLTEXT01.pdf
14. Park, H., EL Azzaoui, A., & Park, J. H. (2025). AIDS-Based Cyber Threat Detection Framework for Secure Cloud-Native Microservices. Electronics, 14(2), 229.
15. Raftopoulos, C. (2025). Cloud security with Docker and Kubernetes (Master's thesis, Πανεπιστήμιο Πειραιώς)
16. Rahman, A., Shamim, S. I., Bose, D. B., & Pandita, R. (2023). Security misconfigurations in open source Kubernetes manifest: An empirical study. ACM Transactions on Software Engineering and Methodology, 32(4), 1-36. https://par.nsf.gov/servlets/purl/10415562
17. Ramachandran, S., Agrahari, R., Mudgal, P., Bhilwaria, H., Long, G., & Kumar, A. (2023)
18. Ramachandran, S., Agrahari, R., Mudgal, P., Bhilwaria, H., Long, G., & Kumar, A. (2023). Automated log classification using deep learning. Procedia Computer Science, 218, 1722-1732.

19. Rice, L. (2023). Learning eBPF. " O'Reilly Media, Inc.". https://cilium.isovalent.com/hubfs/Learning-eBPF%20-%20Full%20book.pdf
20. Rönnbäck, M., & Åberg, F. (2022). Automatic enforcement of container security guidelines through policy as code.
21. Rostami, G. (2023). Role-based access control (RBAC) authorization in Kubernetes. Journal of ICT Standardization, 11(3), 237–260.
22. Shamim, S. I. (2021, August). Mitigating security attacks in Kubernetes manifests for violation of security best practices. In Proceedings of the 29th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering (pp. 1689–1690). https://dl.acm.org/doi/pdf/10.1145/3468264.3473495
23. Viktorsson, W., Klein, C., & Tordsson, J. (2020, November). Security-performance trade-offs of Kubernetes container runtimes. In 2020, the 28th International Symposium on Modeling, analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS) (pp. 1-4). IEEE. https://www.diva-portal.org/smash/get/diva2:1469183/FULLTEXT01.pdf
24. Yusof, Z. B. (2024). Exploration of Advanced Persistent Threats: Techniques, Mitigation Strategies, and Impacts on Critical Infrastructure. International Journal of Advanced Cybersecurity Systems, Technologies, and Applications, 8(12), 1–9.
25. M. Q. Mohammed, M. G. S. Al-Safi, and A. M. Faris, “Statistical Anomaly Detection for Enhanced Cybersecurity Using AI-Based Wireless Networks,” *Ingénierie des systèmes d information*, vol. 29, no. 5, pp. 1743–1754, Oct. 2024, doi: https://doi.org/10.18280/isi.290508.
26. N. Mohamed, “Current Trends in AI and ML for cybersecurity: a state-of-the-art Survey,” *Cogent Engineering*, vol. 10, no. 2, Oct. 2023, doi: https://doi.org/10.1080/23311916.2023.2272358.