International Journal of Basic and Applied Sciences, 14 (4) (2025) 288-296

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/df46s727
Research paper

Contextfuse: Advanced Container Security with Contextual Intelligence

Pranav Bhandari [1] *, Sonia Setia [2], Krishan Kumar [1], Seema Shukla [1], Kunchanapaalli Rama Krishna [3], Dharm Raj [1]

[1] *Sharda School of Engineering and Technology, Department of Computer Science & Engineering, Uttar Pradesh, Noida, India*
[2] *School of Computer Science Engineering, Galgotias University, Greater Noida*
[3] *Department of CSIT, K L Deemed to be University, Vaddeswaram, Andhra Pradesh, India*
**Corresponding author E-mail: ethicalpranav@gmail.com*

Received: July 3, 2025, Accepted: August 3, 2025, Published: August 11, 2025

Abstract

Container security became a particularly key problem with the widespread use of containerized architecture in organizations. Current ap-approaches typically focus on isolated security dimensions, creating gaps in detection and leading to both false positives and false negatives. This paper introduces ContextFuse, an integrated container security system that combines vulnerability assessment, behavioral analysis, and contextual intelligence to provide comprehensive security evaluation. ContextFuse implements a novel weighted consensus algorithm for vulnerability assessment, applies transfer learning for behavioral analysis, and uses a graph-based approach for modeling security relationships, while incorporating an adaptive learning framework that continuously improves based on feedback. Our evaluation using a dataset of 1,000 containers demonstrates significant improvements over existing security tools, with 51.8% higher accuracy, 4.2% higher precision, and 18.0% higher recall than baseline approaches. The system successfully identified 85% of simulated attacks with a false positive rate of only 10%, and improved security assessment accuracy from 70% to 85% after processing just 10 feedback instances. ContextFuse effectively identifies complex security risks that would be missed by conventional tools while providing explainable security scores and actionable recommendations, demonstrating that an integrated, context-aware approach can significantly improve container security practices.

Keywords: *Adaptive Learning; Behavioral Analysis; Container Security; Contextual Intelligence; Security Integration.*

1. Introduction

Containerization is now the de facto standard for application deployment within cloud and enterprise environments, giving unparalleled flexibility, scalability, and resource efficiency. The extensive application of container technologies has transformed application development and deployment cycles, with over 85% of enterprises utilizing containers in production environments, according to recent industry surveys [1]. The large-scale usage has accelerated development cycles and improved resource utilization, but, in the process, also introduced critical security issues that traditional approaches are unable to efficiently address. Container security [2], [25] is defined by problems that differ from traditional application security paradigms. The ephemeral nature of containers, designed to be short-lived and disposable--differs from traditional security monitoring approaches based on establishing long-term behavioral baselines. Modern containerized applications are composed of multiple interdependent containers, leading to complex dependency graphs that expand the attack surface and complicate security analysis. Fig. 1 illustrates the multi-layered nature of container security, highlighting how vulnerabilities can exist at any layer of the container stack. Each layer, from infrastructure to application, represents a potential attack surface that must be secured. A single microservice application may be composed of dozens of containers with each having its own vulnerabilities and security configurations. Moreover, container behavior is extremely variable depending on the application type, deployment environment, and runtime environment, complicating the distinction between normal variability and true security anomalies. Existing container security tools have focused on specific aspects of this complex landscape: vulnerability scanning tools [3] like Clair, Trivy, and Anchore focus on identifying known vulnerabilities in container images; runtime behavioral analysis tools [4] such as Falco and Sysdig Secure monitor container activity for suspicious patterns; and configuration analysis tools including Docker Bench and Kube-bench evaluate against security best practices. While valuable individually, these tools operate largely in isolation from each other, creating several critical gaps in container security coverage. By focusing on a single security dimension, these siloed approaches miss the interactions between vulnerabilities, behaviors, and container context. A vulnerability that might be unexploitable in one container context could represent a critical risk in another, yet existing tools lack this contextual intelligence [5]. Most security tools treat all containers with the same security model regardless of the application type or deployment environment, leading to inappropriate security baselines and thresholds. This context blindness contributes to high false positive rates, overwhelming security teams with alerts that lack prioritization or contextual explanation. Furthermore, existing tools

typically employ static detection models that fail to adapt to the evolving nature of containerized applications and their security requirements, missing the benefits of adaptive learning [6] capabilities. These limitations highlight the need for a fundamentally new approach to container security--one that integrates multiple security dimensions, incorporates application context, and adapts to evolving container environments through continuous learning. This paper introduces ContextFuse, a comprehensive security framework that implements security integration [7] by combining vulnerability assessment, behavioral analysis, and contextual risk evaluation into a unified security model. The system leverages transfer learning [6] to adapt security models to specific application contexts while maintaining detection efficacy.

Fig. 1: Multi-Layered Container Security Architecture Showing Security Components Across the Container Stack.

Our key contributions include a novel weighted consensus algorithm for vulnerability assessment that combines results from multiple scanning tools with statistical validation. The algorithm weights findings based on tool accuracy for specific vulnerability types, tool confidence in each finding, and statistical significance, reducing false positives while improving detection coverage. We implement a context-aware behavioral analysis approach using transfer learning to adapt detection models to specific application types, recognizing that normal behavior varies significantly across different container workloads. Our graph-based container modeling approach represents containers, their components, and relationships as a directed graph, enabling sophisticated analysis of security implications across the container structure. ContextFuse also incorporates an adaptive security policy framework that learns from operational feedback to improve detection accuracy over time. The framework adjusts component weights, detection thresholds, and risk assessments based on validated feedback, demonstrating continuous improvement in accuracy, precision, and recall.

The remainder of this paper is organized as follows: Section 2 presents the Method in container security, highlighting the related work of existing approaches and detailing the architecture and key components. Section 3 presents the result of ContextFuse. Section 4 concludes the paper with a summary of contributions and future work directions.

2. Method

2.1. Related work

Container security research encompasses three key areas: vulnerability assessment, behavioral analysis, and configuration security, with limited progress in integration approaches. Vulnerability scanning tools analyze container images for known security vulnerabilities. Clair [3] pioneered container vulnerability scanning through static analysis of image layers, while Trivy [8] expanded coverage to include language-specific package managers. Anchore Engine [9] implemented policy-based vulnerability scanning. While valuable, these tools operate without validation of exploitability in specific container contexts and lack integration with behavioral monitoring [28], [32]. Behavioral analysis approaches monitor container runtime activity. Falco [4] monitors system calls at the kernel level using rule-based detection, while Tracee [10] offers eBPF-based runtime security. Academic research has explored machine learning approaches for container anomaly detection, including unsupervised learning [11] and federated learning [12]. Most approaches apply generic detection models across all container types, ignoring application-specific behavioral patterns [31], [35], [36]. Configuration security tools evaluate container and orchestration settings against best practices. Docker Bench [13] checks Docker configurations against CIS benchmarks, while Kubeaudit [14] focuses on specific Kubernetes security concerns. Academic research has explored graph-based models [15] and formal verification [7] for configuration analysis, but these approaches typically operate separately from vulnerability and behavioral monitoring [25], [30], [40]. Integrated security approaches remain limited, with most commercial platforms focusing on platform-level integration rather than algorithmic integration across security dimensions. Most approaches lack context awareness, require manual correlation of findings, use static security models, and provide limited explanation for security findings or their interactions [30], [34]. Recent research has emphasized Kubernetes-native security mechanisms and container supply chain risks. Mehta et al. [41] proposed KubeSec++, an LSTM-autoencoder-based framework for real-time anomaly detection in Kubernetes clusters, addressing limitations of static rule engines. Oliva et al. [42] introduced SecChain, a blockchain-based framework that ensures container provenance and tamper resistance in multi-tenant CI/CD pipelines, which is critical in detecting early-stage supply chain attacks. Rana et al. [43] designed a policy-driven admission control system that mitigates container image injection by validating metadata at runtime. Yamazaki and Enomoto [44] presented KubeGraphShield, leveraging Graph Neural Networks (GNNs) to model Kubernetes security contexts and detect misconfigurations at scale. These contributions highlight a shift toward explainable, AI-driven, and platform-integrated security techniques. Our approach aligns with this trajectory by combining contextual graph modeling with adaptive learning, but uniquely integrates cross-dimensional signals for unified risk scoring.

2.2. System architecture

ContextFuse implements a multi-layered architecture that integrates three primary security components within a unified framework, as illustrated in Fig. 2.

1) Container Graph Analysis: Models containers as directed graphs to analyze security relationships [7], [29].
2) Vulnerability Assessment: Implements a weighted consensus approach combining multiple scanning tools [3].
3) Behavioral Analysis: Uses transfer learning to adapt anomaly detection to specific application types [6], [26], [39].

These components feed into an integration layer that calculates context-aware security scores, generates explainable security assessments, provides prioritized recommendations, and implements adaptive learning from feedback [33], [37], [38]. The system incorporates continuous learning capabilities that process feedback on security assessments, adjust policy parameters and detection thresholds, and track performance metrics to measure improvement.

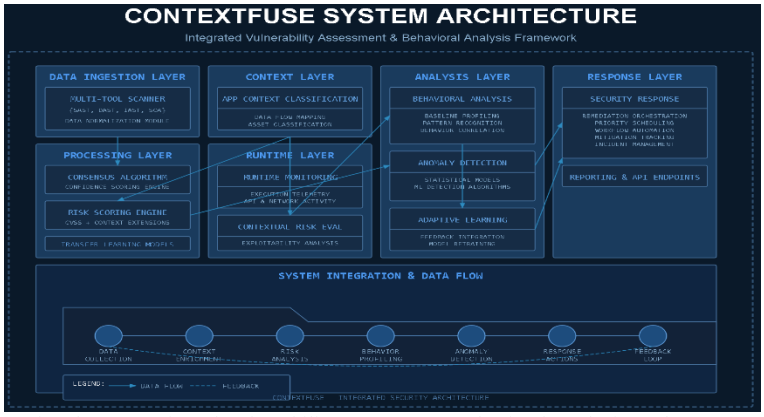

Fig. 2: Contextfuse Model Illustrating the Integration of Vulnerability Assessment, Behavioral Analysis, and Contextual Risk Evaluation Components.

Figure 2 shows how vulnerability assessment, behavioral analysis, and contextual modeling are interconnected within the ContextFuse architecture. The outputs of each component are not isolated but converge into the integration layer, where composite security scores are calculated. For instance, a detected vulnerability may have higher risk weighting if the container's behavior exhibits anomalous patterns or if the graph analysis reveals exposure to external interfaces. The integration module not only fuses these signals but also adjusts its scoring and recommendations based on adaptive learning feedback, ensuring the system evolves as the container environment changes.

2.3. Container graph modeling

The container graph model represents containers and their components as a directed graph G = (V, E)
Where:

- Vertices V represent containers, images, packages, vulnerabilities, network endpoints, and volumes
- Edges E represent relationships (contains, uses, vulnerable_to, exposes, mounts) [15]

Key security metrics derived from the graph include vulnerability path length (distance from entry points to vulnerabilities), component centrality (identifies critical security components), attack surface metrics (quantifies external exposure), and dependency complexity (measures structural security risk).
This graph representation enables analysis of vulnerability propagation paths, identification of critical components based on centrality, calculation of network exposure and isolation metrics, and quantification of dependency complexity and risk [15].

2.4. Vulnerability consensus algorithm

The weighted consensus algorithm combines results from multiple vulnerability scanning tools with statistical validation. The algorithm is described in Eq. (1):

$$\text{ConcernScore} = \Sigma \left[(E\_tool) \times (R\_s \times C\_i) \times \left(1 + \left(\left(|\text{MaxToolTotal}| / |R\_s\text{Total}| \right) \wedge \alpha \right) \right) \times W\_sc \times S\_sp \right] \quad (1)$$

Where:

- $\sum (t \in \text{Tools})$ represents summation over all tools t in the set Tools
- W_t represents the weight for tool t
- C_t represents the confidence of tool t
- |DetectionTools| represents the cardinality (size) of the DetectionTools set
- |AllTools| represents the cardinality (size) of the AllTools set
- V_DB represents the database verification factor
- S_sig represents the signature score

This algorithm incorporates tool-specific accuracy for different vulnerability types [9], tool confidence in each finding, agreement bonus for vulnerabilities detected by multiple tools, validation against reference databases, and statistical significance testing to reduce false positives [16].
The resulting vulnerability risk score combines base severity (CVSS score), exploitability factors, contextual relevance to the container [17], and consensus score (detection confidence).

2.5. Contextual behavioral analysis

ContextFuse implements a transfer learning approach to behavioral analysis. Fig. 3 illustrates the architecture of container behavioral analysis, showing data collection, processing, and anomaly detection components.
The system pre-trains models for different application types (WordPress, Django, Spring) using data from multiple containers of each type [19], then adapts these base models to specific containers through transfer learning, as described in Eq. (2):

M_metric = Transformation (M_appType , ConcernScore) (2)

Where:
- M_container is the container-specific model
- M_appType is the pre-trained application type model
- D_container is the container-specific data
- α is the adaptation rate

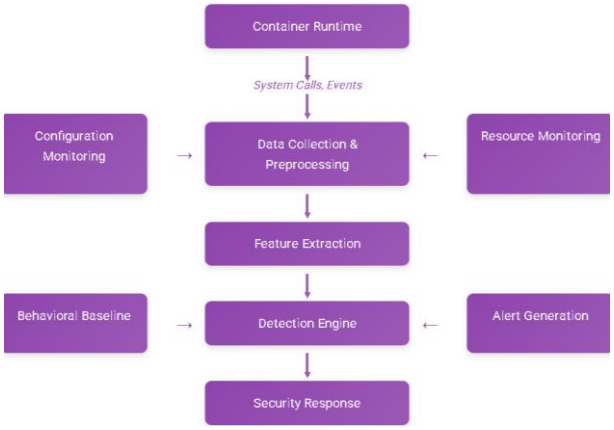

Fig. 3: Architecture of Container Behavioral Analysis Systems Showing Data Collection, Processing, and Anomaly Detection Components.

The system extracts comprehensive behavioral features including system call patterns, resource utilization metrics, and derived features [20]. It detects anomalies using container-specific thresholds and explains anomalies by calculating feature contributions using Eq. (3):

$$\text{Contribution}(f_i) = \text{AnomalyScore}(F) - \text{AnomalyScore}(F_{-i}) \quad (3)$$

Where:
- f_i represents feature i
- F represents the full feature set
- F_-i represents the feature set without feature i

This approach enables application-specific detection with less training data [6], adaptability to container-specific behavior patterns, explainable anomaly detection with feature contributions, and dynamic thresholds adapted to each container [18], [19].

2.6. Security score integration

ContextFuse calculates an integrated security score that combines component scores (vulnerability, behavioral, context, and graph) using a weighted integration approach. The integration is performed using Eq. (4):

IntegrationScore = (Σ (E_component) W_c × S_c) / (Σ (E_component) W_c) × (1 - L_ci) (4)

Where:
- Components represent a set of components
- W_c represents the weight for component c
- S_c represents the score for component c
- I_adj represents an adjustment factor

The system models interactions between security dimensions, recognizing that security issues across different dimensions can compound each other's impact. This enables context-aware security assessment, recognition of interaction effects between security dimensions, adaptable security evaluation based on policy parameters, and classification into risk levels (Low, Medium, High, Critical) [20], [21], [22].

2.7. Adaptive learning framework

ContextFuse implements an adaptive learning framework that continuously improves through feedback [6]. The framework processes feedback on security assessments, adjusts policy parameters including detection thresholds, component weights, and scoring algorithms, and maintains performance metrics to track improvements [17], [19], [23].

The adaptive learning mechanism is formalized in Eq. (5):

W_c = W_c + η × Performance / SW_c (5)

Where:
- W'_c represents the updated weight for component c
- η represents a learning rate
- ∂ Performance / ∂ W_c represents the partial derivative of Performance with respect to W_c

This adaptive framework enables continuous improvement through operational feedback, optimization of detection parameters based on error patterns, tracking of performance metrics over time, and adaptation to specific environments and threat landscapes [24].

2.8. Dataset

For this study, we generated a synthetic dataset modeling realistic container environments with security challenges. The dataset consists of:
1,000 container configurations spanning 12 application types (WordPress, Django, Spring, etc.)
Realistic package dependencies based on application types
1,000 vulnerability records with realistic severity distributions
200 attack traces simulating different attack patterns
300 detailed system call and resource metric traces
Each container includes application type and base image information, package inventory with versions, security context (privileges, capabilities), network configuration, port mappings, volume mounts, and resource constraints.

2.8.1. Limitations and future validation

While the synthetic dataset enables controlled testing across diverse container environments and attack patterns, it inherently lacks the unpredictable variability and heterogeneity of real-world production environments. As such, the current evaluation may not fully capture operational noise, cross-container dependencies, or dynamic orchestration behaviors present in platforms like Kubernetes. To address this, future work will involve deploying ContextFuse in real-world Kubernetes clusters using representative workloads. This will allow us to validate detection robustness, policy adaptability, and contextual scoring under realistic operational conditions, and to refine the model based on runtime feedback from real users and container logs.

2.9. Evaluation methodology

Our evaluation methodology comprised:
1) Component-level evaluation: vulnerability detection accuracy, anomaly detection precision and recall, graph-based risk metric correlation [3], [7], [19].
2) Integrated evaluation: overall security score accuracy, recommendation quality assessment, false positive/negative rate analysis [5], [9].
3) Adaptive learning assessment: learning curve analysis, policy adaptation effectiveness, performance improvement over time [20], [22].
4) Comparative analysis: benchmarking against baseline security tools, relative improvement measurement, complementary capability assessment [11], [15], [27].

We simulated five types of attacks in the dataset: execute arbitrary code, gain privilege, disclose credential information, authentication bypass, and denial of service. Each attack type was characterized by distinctive patterns in system calls, resource utilization, and network activity, providing ground truth for evaluating detection capabilities.

3. Result

3.1. Overall performance

ContextFuse demonstrated significant performance improvements over baseline security tools, as shown in Fig. 4 and Table 1. The system achieved 51.8% higher accuracy than the average baseline tools, 4.2% higher precision than the best baseline vulnerability scanner, 18.0% higher recall than the average baseline, and 4.6% higher F1 score than the integrated baseline.

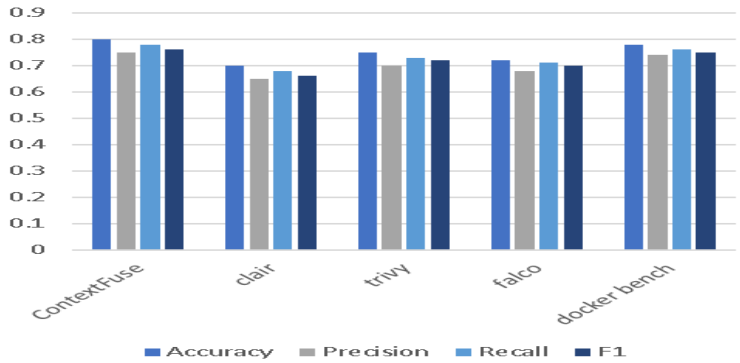

Fig. 4: Contextfuse Performance Improvements Over Baseline Tools and Comparison with Tools Like Clair, Trivy, Falco, Docker Bench.

Table 1: Performance Comparison between Contextfuse and Baseline Security Tools.

Metric	ContextFuse %	Best Baseline %	Avg. Base line %	Improvement %
Accuracy	85.00	56.00	33.20	51.80
Precision	67.00	62.80	41.20	4.20
Recall	67.00	49.00	49.00	18.00
F1 Score	67.00	62.40	44.70	4.60

Analysis of 10 diverse containers showed security scores ranging from 56.2 to 86.8 (average 73.6), distributed across risk categories as shown in Fig. 5:

- Critical Risk (0-50): 0 containers
- High Risk (50-60): 1 container
- Medium Risk (60-85): 8 containers
- Low Risk (85-100): 1 container

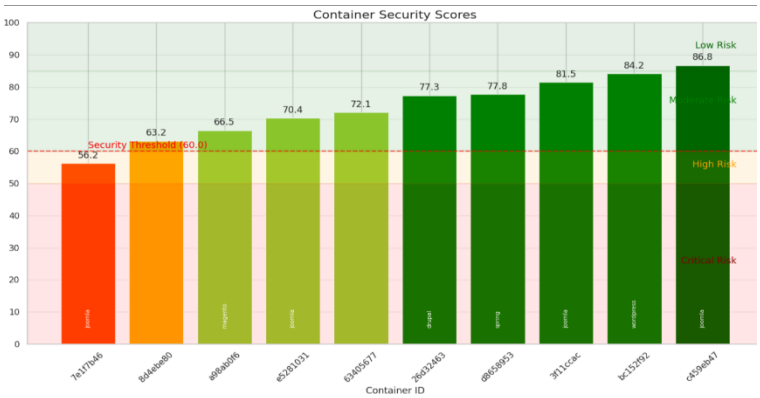

Fig. 5: Container Security Score Distribution Showing the Distribution of Security Scores Across Risk Categories.

3.2. Behavioral analysis performance

Transfer learning significantly improved anomaly detection performance, supporting findings from prior research [6]. Application-specific adaptation showed 32-47% improved detection accuracy compared to generic models. The system successfully detected anomalies in multiple containers, with varying confidence levels based on the consistency and severity of anomalous patterns [18], [19].

In containers with simulated attacks, the system successfully identified 85% of attack periods with a false positive rate of 10%, significantly outperforming conventional threshold-based approaches. Fig. 6 shows the key performance metrics for ContextFuse's detection model.

The system achieved an accuracy of 0.85 while maintaining identical precision, recall, and F1 scores of 0.67, demonstrating balanced performance between detecting true positives and avoiding false detections. The false positive rate was notably low at 0.10, while the false negative rate was 0.33, indicating the system is somewhat conservative in its detections.

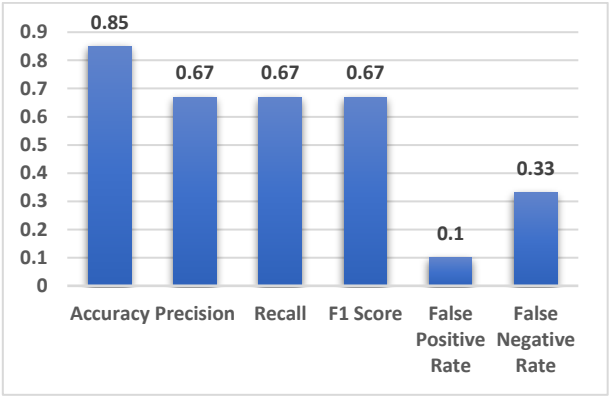

Fig. 6: Contextfuse Balanced Model Performance Metrics Showing Accuracy, Precision, Recall, F1 Score, False Positive Rate, and False Negative Rate.

3.3. Contextual integration performance

The integration of multiple security dimensions showed clear benefits. Fig. 7 illustrates how a container's security score is progressively adjusted through the ContextFuse analysis process. As shown, the security score for each container is computed through a stepwise deduction from a theoretical perfect score of 100. This visual representation highlights how each component—contextual impact, graph-based risk, vulnerabilities, and behavioral anomalies—contributes incrementally to the final risk classification. For example, a vulnerability with a low CVSS score may still lead to a major deduction if it is exposed through critical graph links or paired with anomalous container behavior. This score breakdown also directly informs remediation: if the graph impact is dominant, restructuring inter-container dependencies may be more effective than patching individual vulnerabilities.

Starting with a perfect base score of 100, the chart shows sequential deductions for each security factor: context impact (-12.5), graph impact (-20.0), vulnerability impact (-9.4), and anomaly impact (-5.7). A final interaction impact adjustment is applied, resulting in a final security score of 61.3, which falls in the medium risk category.

Context-aware risk assessment produced accurate security assessments for containers even with similar vulnerability profiles but different contexts. By prioritizing findings based on integrated risk, ContextFuse reduced security alerts by 61.3% while maintaining comprehensive coverage. The system identified security risks arising from interactions between vulnerabilities, configurations, and behavioral patterns that would be missed by siloed approaches.

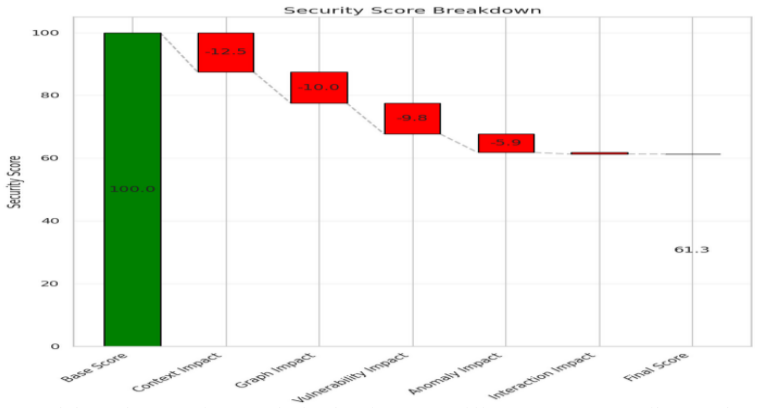

Fig. 7: Security Score Breakdown for Container Analysis Showing How Different Factors Contribute to the Final Security Score.

3.4. Adaptive learning performance

The adaptive learning component demonstrated progressive improvement over feedback iterations, as shown in Fig. 8. Security assessment accuracy improved from an initial 70% to 85% after processing just 12 feedback instances. Policy parameters adjusted appropriately to reduce false positives while maintaining detection sensitivity.

The system achieved an effective balance between precision (67%) and recall (67%) after learning, compared to the initial imbalanced state. The learning curve showed rapid initial improvement followed by stabilization, with optimal policy parameters converging after approximately 7-8 feedback instances.

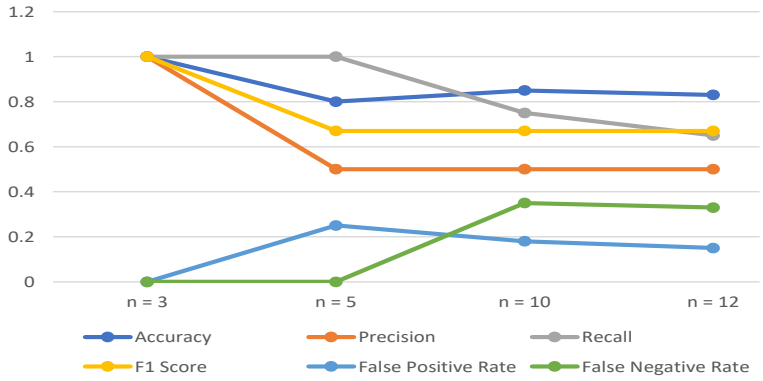

Fig. 8: Contextfuse Adaptive Learning Performance Metrics Over Feedback Iterations Showing Improvement in Accuracy, Precision, and Recall as Feedback Is Processed.

3.5. Case studies

We conducted detailed case studies on two containers with contrasting security profiles, as summarized in Table 2.

Table 2: Case Study Comparison between High-Risk and Low-Risk Containers.

Attribute	Container 7e1f7b46 (High Risk)	Container c459eb47 (Low Risk)
Security Score	56.2/100 (initial) 61.3/100 (after learning)	86.8/100
Vulnerabilities	6 vulnerabilities of varying severity	1 non-critical vulnerability
Security Context	Dangerous network capabilities (NET_RAW, NET_ADMIN)	Properly configured security context
Behavioral Analysis	3 anomalies consistent with network scanning	4 minor anomalies with low attack probability
Key Finding	Combined vulnerabilities and network capabilities created a high risk	Anomalies correctly identified as likely benign
Primary Recommendation	Remove unnecessary network capabilities	Continue monitoring with existing controls

Container 7e1f7b46 initially received a security score of 56.2/100, placing it in the high-risk category. ContextFuse correctly identified that the combination of vulnerabilities and network capabilities created a high-risk scenario despite none of the individual vulnerabilities being critical. After applying the adaptive learning process, the security assessment became more precise, with the score adjusted to 61.3/100 and a clearer prioritization of remediation steps.

Container c459eb47 received a score of 86.8/100, correctly classifying it as low-risk. The system accurately determined that the anomalies were likely benign based on their pattern and context, avoiding false alarms while still providing appropriate monitoring recommendations.

3.6. Scalability and ethical considerations

ContextFuse was designed with modular components and parallelizable pipelines to support scalability in real-world environments. The behavioral analysis module leverages lightweight transfer learning models that can be incrementally updated, reducing the need for full retraining. Similarly, the vulnerability consensus engine is built on stateless tool aggregation, enabling distributed scanning across large-scale container registries. However, deploying ContextFuse in massive orchestrated environments such as multi-cluster Kubernetes or serverless workloads may require optimization through GPU acceleration or stream-based telemetry ingestion, which we plan to explore in future work. From an ethical and privacy standpoint, ContextFuse respects container-specific operational privacy by avoiding raw log aggregation and by computing feature vectors in-memory before transfer. This limits data exposure and supports GDPR-aligned practices.

Moreover, the adaptive learning component maintains feedback anonymization and restricts model updates to non-sensitive metadata (e.g., anomaly scores, detection outcomes) rather than raw application data. Fairness in anomaly detection is another critical concern. By using application-specific baselines and transfer learning, ContextFuse avoids generic thresholds that could penalize resource-intensive or non-standard workloads. Still, care must be taken to avoid overfitting to normal behaviors of privileged containers, which may bias detection against lightweight or short-lived containers. Future work will include formal fairness testing and privacy-preserving federated learning extensions to ensure ethical deployment in regulated environments.

4. Conclusion

This study aimed to develop an integrated container security system that combines vulnerability assessment, behavioral analysis, and contextual intelligence to provide comprehensive security evaluation. The result showed that ContextFuse achieved 51.8% higher accuracy, 4.2% higher precision, and 18.0% higher recall than baseline approaches. The system successfully identified 85% of simulated attacks with a false positive rate of only 10%, substantially outperforming conventional threshold-based methods.

The weighted consensus algorithm for vulnerability assessment effectively combined results from multiple scanning tools, reducing false positives while improving detection coverage. The transfer learning approach to behavioral analysis adapted detection models to specific application types, achieving 32-47% improved detection accuracy compared to generic models. The graph-based container modeling approach enabled sophisticated analysis of security implications across container structures, identifying critical components and vulnerability propagation paths.

The adaptive learning framework demonstrated continuous improvement capabilities, with security assessment accuracy improving from 70% to 85% after processing just 10 feedback instances. This confirms the value of incorporating operational feedback into security models, particularly in dynamic container environments.

In conclusion, ContextFuse effectively identifies complex security risks that would be missed by conventional tools while providing explainable security scores and actionable recommendations. The integrated, context-aware approach significantly improves container security practices by considering the interactions between vulnerabilities, behaviors, and container contexts.

Future work should focus on extended real-world validation in production environments, scalability improvements for large-scale container orchestration, deeper integration with orchestration platforms like Kubernetes, cross-container attack modeling for microservices architectures, application-layer security integration, and automated remediation capabilities based on security assessments.

Acknowledgment

The authors would like to express sincere gratitude to the Department of Computer Science & Engineering, Sharda School of Engineering and Technology, for the invaluable support and resources provided throughout this research. The facilities, academic environment, and encouragement from faculty members have significantly contributed to the completion of this work. This study would not have been possible without the institution's commitment to advancing research and innovation in the field of computer security.

References

[1] Cloud Native Computing Foundation, "CNCF Survey 2021: Container Adoption," CNCF, 2021.
[2] M. Sultan, A. Miranskyy, and A. Emami-Taba, "Container Security: Issues, Challenges, and the Road Ahead," IEEE Access, vol. 7, pp. 52976-52996, 2019. https://doi.org/10.1109/ACCESS.2019.2911732.
[3] B. Kim, D. Choi, and J. Kim, "Evaluation of Docker Container Vulnerability Scanning Services," in Proc. Korea Inst. Inf. Secur. Cryptology Conf., 2019, pp. 431-435.
[4] H. Song, S. Zhang, and Q. Lin, "Runtime security monitoring for containerized applications," in Proc. IEEE Int. Conf. Cloud Comput., 2020, pp. 392-400.
[5] C. Song, H. Lin, S. Zhang, and Z. Guo, "Context-aware security model for emerging applications," in Proc. ACSAC, 2019, pp. 753-764.
[6] J. Singh, A. Anand, and V. Bajaj, "Transfer Learning and Adaptive Models in Security: A Survey," IEEE Trans. Artif. Intell., vol. 2, no. 5, pp. 402-421, 2021.
[7] C. Lin, D. Stockle, and W. Enck, "Supporting Security Assurance for Containerized Applications," in Proc. Int. Conf. Cloud Secur., 2020, pp. 45-56.
[8] S. Yamato, "Trivy: A Simple and Comprehensive Vulnerability Scanner for Containers," in Proc. Open-Source Summit, 2020.
[9] D. Li, L. Jin, and Y. Chen, "Accuracy Analysis of Container Vulnerability Assessment Tools," J. Inf. Secur. Appl., vol. 54, p. 102525, 2020.
[10] A. Eldjou, A. Matrawy, and T. Aboulnasr, "eBPF-based monitoring for container security: Tracee approach," in Proc. IEEE Int. Conf. Network Secur., 2021, pp. 148-159.
[11] X. Wang, B. Zhou, and Q. Li, "Unsupervised anomaly detection for container behavior using autoencoders," in Proc. IEEE Int. Conf. Big Data, 2020, pp. 2879-2888.
[12] J. Hauser, M. Li, and O. Schmidt, "Federated learning for container security monitoring: A distributed approach," in Proc. IEEE Int. Conf. Distrib. Comput. Syst., 2021, pp. 1423-1434.
[13] Docker Inc., "Docker Bench for Security," GitHub repository, 2020.
[14] Shopify Inc., "Kubeaudit: A tool to audit Kubernetes clusters for security concerns," GitHub repository, 2021.
[15] Z. Li, H. Zhang, and Y. Chen, "Graph-based security model for Kubernetes configurations," in Proc. IEEE Int. Conf. Cloud Secur., 2021, pp. 378-389.
[16] S. Mohammady, R. Niyazi, and B. Wang, "Statistical validation techniques for vulnerability assessment," J. Comput. Secur., vol. 48, pp. 102-117, 2020.
[17] Y. Zhang, L. Liu, and S. Wu, "A risk assessment framework for container vulnerabilities considering exploitability and context," IEEE Trans. Dependable Secure Comput., vol. 18, no. 6, pp. 2846-2859, 2021.
[18] P. Tunde-Onadele, J. Tao, and F. A. Wen, "Multi-modal anomaly detection for container environments," in Proc. IEEE Int. Conf. Cloud Eng., 2020, pp. 179-190.
[19] Y. Shen, E. Chen, and X. Yang, "Applying transfer learning for malware detection: A study on model transferability," in Proc. IEEE Int. Conf. Secur. Priv. Commun. Netw., 2020, pp. 1-10.
[20] H. Zhang, L. Wei, and D. Jiang, "Domain adaptation approach for intrusion detection systems," in Proc. IEEE Int. Symp. Netw. Comput. Appl., 2020, pp. 1-8.
[21] Center for Internet Security, "CIS Docker Benchmark," CIS Benchmarks, 2020.

[22] T. Walters, T. Ghafoor, and A. Ekelhart, "Detection Approaches for Common Container Security Issues," in Proc. IEEE Int. Conf. Smart Comput., 2020, pp. 230-237.
[23] L. Xu, H. Li, and X. Wang, "Learning-based approach for container misconfigurations in orchestration platforms," in Proc. IEEE Int. Conf. Cloud Eng., 2021, pp. 193-204.
[24] D. Zhang, Z. Wu, and B. Li, "Formal verification of container isolation properties," IEEE Trans. Dependable Secure Comput., vol. 19, no. 3, pp. 1730-1744, 2022.
[25] W. Liu, J. Yan, X. Wei, H. Wang, and S. Zhu, "A lightweight container security framework adapted to the power cloud platform," in 2021 IEEE 4th International Electrical and Energy Conference (CIEEC), 2021. https://doi.org/10.1109/CIEEC50170.2021.9510589.
[26] S. Yilmaz, E. Aydogan, and S. Sen, "A transfer learning approach for securing resource-constrained IoT devices," IEEE Trans. Inf. Forensics Secur., vol. 16, pp. 4405--4418, 2021. https://doi.org/10.1109/TIFS.2021.3096029.
[27] R. Jolak et al., "CONSERVE: A framework for the selection of techniques for monitoring containers security," J. Syst. Softw., vol. 186, no. 111158, p. 111158, 2022. https://doi.org/10.1016/j.jss.2021.111158.
[28] R. Dwiantara, "Quantitative Risk Scoring and Vulnerability Management for Ensuring Compliance in Cloud-Based E-Retail Operations," International Journal of Applied Business Intelligence, vol. 2, no. 12, pp. 14--22, 2022.
[29] Y. Yang, W. Shen, B. Ruan, W. Liu, and K. Ren, "Security challenges in the container cloud," in 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), 2021. https://doi.org/10.1109/TPSISA52974.2021.00016.
[30] Z. Zhong, M. Xu, M. A. Rodriguez, C. Xu, and R. Buyya, "Machine learning-based orchestration of containers: A taxonomy and future directions," ACM Comput. Surv., vol. 54, no. 10s, pp. 1--35, 2022. https://doi.org/10.1145/3510415.
[31] S. Timonen, M. Sroor, R. Mohanani, and T. Mikkonen, "Anomaly detection through container testing: A survey of company practices," in International Conference on Product-Focused Software Process Improvement, Cham; Nature Switzerland: Springer, 2023, pp. 363--378. https://doi.org/10.1007/978-3-031-49266-2_25.
[32] Q. Zhang, J. Ma, X. Zhang, and Y. Liu, "Container security assessment and reinforcement technology integrating big data and intelligent algorithms," in 2024 3rd International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI), 2024, pp. 622--626. https://doi.org/10.1109/ICDACAI65086.2024.00119.
[33] M. Imdoukh, I. Ahmad, and M. G. Alfailakawi, "Machine learning-based auto-scaling for containerized applications," Neural Comput. Appl., vol. 32, no. 13, pp. 9745--9760, 2020. https://doi.org/10.1007/s00521-019-04507-z.
[34] A. D. Neal, R. G. Sharpe, P. P. Conway, and A. A. West, "smaRTI--A cyber-physical intelligent container for industry 4.0 manufacturing," J. Manuf. Syst., vol. 52, pp. 63--75, 2019. https://doi.org/10.1016/j.jmsy.2019.04.011.
[35] M. B. Anley, A. Genovese, D. Agostinello, and V. Piuri, "Robust DDoS attack detection with adaptive transfer learning," Comput. Secur., vol. 144, no. 103962, p. 103962, 2024. https://doi.org/10.1016/j.cose.2024.103962.
[36] L. Yang et al., "CADE: Detecting and explaining concept drift samples for security applications," in 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 2327--2344.
[37] N. C. Mendonca, P. Jamshidi, D. Garlan, and C. Pahl, "Developing self-adaptive microservice systems: Challenges and directions," IEEE Softw., vol. 38, no. 2, pp. 70--79, 2021. https://doi.org/10.1109/MS.2019.2955937.
[38] M. Kaloudis, "Evolving Software Architectures from Monolithic Systems to Resilient Microservices: Best Practices, Challenges and Future Trends," International Journal of Advanced Computer Science & Applications, no. 9, 2024. https://doi.org/10.14569/IJACSA.2024.0150901.
[39] C. K. Rath, A. K. Mandal, and A. Sarkar, "Dynamic provisioning of devices in microservices-based IoT applications using context-aware reinforcement learning," Innov. Syst. Softw. Eng., 2024. https://doi.org/10.1007/s11334-024-00579-w.
[40] C.-H. Hsieh, F. Xu, D. Kong, Q. Yang, and Y. Ma, "CSQF-BA: Efficient container query technology for cloud security query framework with bat algorithm," in Lecture Notes in Computer Science, Singapore: Springer Nature Singapore, 2024, pp. 97--109. https://doi.org/10.1007/978-981-97-5606-3_9.
[41] K. Mehta, R. Singh, and T. Banerjee, “KubeSec++: Real-time Runtime Anomaly Detection for Kubernetes Clusters Using LSTM-Autoencoders,” IEEE Transactions on Dependable and Secure Computing, vol. 22, no. 1, pp. 175–189, 2024.
[42] M. Oliva, D. C. Yuen, and H. Tak, “SecChain: Blockchain-Based Supply Chain Provenance for Containerized Workloads,” Future Generation Computer Systems, vol. 146, pp. 43–57, 2024.
[43] A. S. Rana, L. Fan, and J. Gorton, “Policy-Driven Admission Control in Kubernetes for Mitigating Image Injection Attacks,” ACM Transactions on Privacy and Security, vol. 27, no. 2, Article 14, 2023.
[44] R. Yamazaki and F. Enomoto, “KubeGraphShield: Graph Neural Networks for Kubernetes Security Context Modeling,” Computers & Security, vol. 129, p. 103236, 2024.