Deep Learning-based Cryptojacking Detection in Cloud Containerized Environments

Ahmed Hussain, Javier Flores, and Panos Papadimitratos
Networked Systems Security (NSS) Group
KTH Royal Institute of Technology, Stockholm, Sweden
Email: {ahmhus, jarf3, papadim}@kth.se

***Abstract*—Cryptojacking attacks in cloud Docker environments exploit computational resources for unauthorized cryptocurrency mining, degrading performance and increasing operational costs. Existing detection methods suffer from performance overhead, the inability to detect obfuscated activities, or the detection of runtime attacks. We present a lightweight Deep Learning (DL) framework analyzing resource utilization patterns for cryptojacking detection in controlled experimental environments. Our methodology employs temporal feature extraction from cloud Virtual Machine (VM) metrics (i.e., CPU, memory, disk I/O, and network utilization) and evaluates four popular DL architectures against direct mining and obfuscated scenarios. Systematic evaluation across Bitcoin, Ethereum, Shiba Inu, and Monero demonstrates that four architectures achieve 1.0 accuracy in our experimental setup. The optimal CNN architecture utilizes 4,548 parameters, maintains > 0.97 precision in obfuscated scenarios, and achieves $< 0.001\%$ false positive rates. The framework enables efficient cryptojacking detection in containerized environments with minimal computational overhead while maintaining robust security capabilities under controlled evaluation conditions.**

***Index Terms*—Cryptojacking, Deep Learning, Cloud Security, Docker Containers, Cryptocurrency Mining, Cloud Computing**

I. Introduction

Docker has emerged as the predominant containerization platform since its introduction by Docker Inc. in 2013, revolutionizing application deployment through its lightweight, portable, and efficient architecture. It enables rapid deployment, scaling, and efficient resource utilization across diverse environments from development to production, with millions of containers deployed daily across cloud platforms, on-premises data centers, and hybrid infrastructures [1]. The platform provides essential benefits, including minimal downtime, seamless application updates, and consistent operation through application and dependency isolation.

However, Docker general adoption has introduced significant security vulnerabilities, particularly in cloud environments where containers share the host operating system and resources [2]. The shared kernel architecture creates attack vectors through which exploits affecting one container can compromise both the host system and co-located containers. Among emerging threats, *cryptojacking attacks* represent a particularly insidious security challenge, where attackers exploit container resources for unauthorized cryptocurrency mining, causing severe performance degradation and increased operational costs.

Cryptojacking attacks in containerized environments manifest through multiple attack vectors: supply chain attacks with malicious code injection into Docker images [3], exploitation of misconfigurations such as containers running with root privileges, and vulnerable dependencies in container images [2]. These attacks cause substantial consumption of Central Processing Unit (CPU), memory, and power resources, degrading system performance for legitimate applications while generating unexpected operational expenses through increased electricity consumption and accelerated hardware degradation.

The complexity of large-scale Docker deployments compounds the security challenge, making the monitoring and securing of every container instance challenging. Attackers can exploit unpatched vulnerabilities, weak access controls, or configuration errors to infiltrate systems, propagate malicious mining software across multiple containers, or directly deploy unauthorized crypto-mining containers [1], [4].

This paper addresses the need for effective detection mechanisms to mitigate cryptojacking attacks in cloud-based Docker environments. Specifically, we investigate three fundamental research questions: (i) Can crypto-mining activities on cloud Virtual Machines (VMs) be detected? (ii) Can we distinguish VMs running legitimate containerized applications from those executing crypto-mining containers? (iii) Can crypto-mining Docker containers be effectively identified within cloud VM environments?

To address these questions, we propose a Deep Learning (DL)-based detection framework that analyzes resource usage patterns, including CPU utilization, memory consumption, disk usage, and network traffic characteristics. Our methodology involves systematic data collection from VMs running both legitimate and crypto-mining Docker containers, advanced sequential feature extraction from raw system metrics, and evaluation of four distinct Neural Network (NN) architectures optimized for classifying normal versus crypto-mining activities across diverse cryptocurrency mining scenarios.

Contribution. (i) We develop and evaluate four DL architectures specifically optimized for cryptojacking attack de-

tection, achieving high classification accuracy with minimal computational overhead. (ii) We establish a systematic temporal resource utilization analysis methodology that captures subtle mining signatures through sequential processing, automatically extracting discriminative patterns without manual feature engineering. (iii) We provide evaluation across direct cryptojacking attacks and obfuscation scenarios, demonstrating robust detection capabilities even when malicious mining activities are concealed within legitimate computational workloads.

Paper Organization. Section II presents preliminaries and related work in container security and crypto-mining detection. Section III presents our adversarial model and defines two operational scenarios representing varying attack levels. Section IV details our methodology and proposed solution. Section V provides performance evaluation through systematic experimentation and resource utilization pattern analysis. Section VI concludes with key findings, implications for cloud security, and future directions.

II. Preliminaries and Related Work

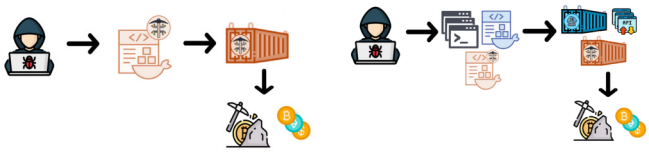
Docker containers have revolutionized application deployment through lightweight virtualization, leveraging Linux kernel features including cgroups and namespaces for resource isolation. Unlike traditional VMs, containers share the host kernel while maintaining application-level isolation, achieving millisecond startup times and efficient memory usage [5]. However, this shared kernel architecture creates security vulnerabilities where kernel exploits can compromise multiple containers simultaneously [4]. Modern cloud deployments integrate containerization with microservices architectures, enabling dynamic scaling across distributed infrastructure where thousands of containers execute with shared computational resources [6]. This operational complexity introduces significant monitoring challenges for detecting unauthorized activities within highly dynamic environments.

Cryptojacking attacks exploit cloud computational resources for unauthorized cryptocurrency mining, targeting Bitcoin, Monero, ZCash, and other emerging alternatives. These attacks manifest through supply chain compromise, configuration exploitation, and runtime payload injection [7], substantially degrading legitimate application performance while generating increased operational costs. Contemporary container security research has established threat models across container lifecycle stages [1], [2], [4], identifying vulnerabilities including image vulnerabilities, exposed API endpoints, container escape attacks, and network configuration weaknesses.

Existing crypto-mining detection methodologies demonstrate diverse approaches with distinct limitations. System call analysis [8] provides behavioral monitoring but introduces performance overhead unsuitable for production environments. Network traffic analysis [9]–[11] offers non-intrusive detection but suffers from VPN obfuscation and encryption limitations. Static analysis techniques [12] demonstrate high pre-deployment accuracy but cannot detect runtime injection attacks. These methodologies lack lightweight, resource-efficient detection mechanisms suitable for large-scale cloud deployments without performance degradation.

Our research addresses these limitations through a resource utilization-based detection framework that balances detection accuracy with operational efficiency. Through analysis of CPU usage patterns, memory consumption, disk utilization, and network traffic characteristics, our approach provides lightweight monitoring suitable for cloud-native environments while maintaining robust detection capabilities against cryptojacking attacks across diverse cryptocurrency mining scenarios.

III. Adversarial Model and Scenarios



(a) Direct deployment (b) Obfuscated deployment

Fig. 1: Adversarial scenarios: (a) Direct crypto-mining container deployment without obfuscation, (b) Obfuscated crypto-mining deployment utilizing legitimate containers for activity camouflage.

Our adversarial model demonstrates the threat landscape for cryptojacking attacks in cloud-based containerized environments. Adversaries exploit computational resources within the cloud provider's VMs for profit generation through Docker containers configured with mining pool specifications, target cryptocurrency selection, and wallet addresses. The threat model encompasses two operational scenarios representing varying attack levels, reflecting real-world patterns from straightforward resource exploitation to sophisticated techniques.

Direct Crypto-Mining represents fundamental attacks where adversaries prioritize maximum resource utilization over stealth. The adversary deploys dedicated crypto-mining containers within compromised VMs, maximizing CPU, disk I/O, memory, and network bandwidth exploitation exclusively for mining operations. This approach generates optimal profits through complete system resource dedication while maintaining operational simplicity, making detection straightforward through resource monitoring techniques.

Obfuscated Crypto-Mining represents threats where adversaries implement sophisticated obfuscation techniques to evade detection systems. The adversary simultaneously deploys multiple container types: benign (legitimate) containers, associated client applications, and crypto-mining containers operating concurrently within the same VM environment.

The obfuscation works through the simulation of legitimate activities. Web server containers and client scripts generate authentic computational workloads through mathematical operations, API communications, and disk I/O activities, consuming substantial system resources. The crypto-mining container operates concurrently, sharing the same resource pool while its signature becomes obscured within broader computational activity. This multi-container deployment maintains crypto-mining profitability while creating detection complexity, as security monitoring systems may attribute elevated resource consumption to legitimate applications rather than identifying embedded crypto-mining activities.

These scenarios establish the foundation for our detection methodology evaluation, representing the spectrum of threat sophistication from direct resource exploitation to advanced evasion techniques. This dual-scenario approach enables thorough assessment of detection framework effectiveness across varying attack complexities, ensuring robust security capabilities against both opportunistic and sophisticated cryptojacking operations in cloud-native environments.

IV. Methodology and Proposed Solution

We focus on developing a resource-efficient DL framework suitable for cloud environments. Our framework captures subtle temporal mining activity signatures, validates detection performance against obfuscation techniques through multi-architecture comparison, and demonstrates practical applicability across diverse cryptocurrency mining scenarios with minimal computational overhead.

Experimental Setup. We employ controlled environments utilizing standardized VMs with 4-core CPU configurations, 4 GB RAM capacity, and 25 GB storage running Ubuntu 22.04.4 LTS with Docker version 25.04. Cryptojacking activities are systematically executed using the lpsm-dev Docker crypto-miner image [13] across four selected cryptocurrencies: Bitcoin, Ethereum, Shiba Inu, and Monero, chosen based on market significance.

Data Collection. Resource monitoring employs standard utilities, including mpstat, free, sar, and iostat, capturing CPU utilization percentages, memory consumption patterns, bidirectional network I/O rates, and disk operation throughput. Experimental control is maintained through simultaneous baseline data collection from identical VM configurations operating without crypto-mining containers. Extended data collection periods of 168 hours per experimental configuration ensure sufficient temporal coverage for robust sequential pattern extraction and model training/validation.

Resource Utilization Analysis. Fig. 2 illustrates the distinctive resource consumption signatures that differentiate cryptojacking activities from legitimate activity across four metrics over extended temporal periods. The analysis reveals clear differences in CPU utilization patterns, where crypto-mining activities exhibit sustained high-intensity consumption ranging from 200-400% compared to baseline legitimate

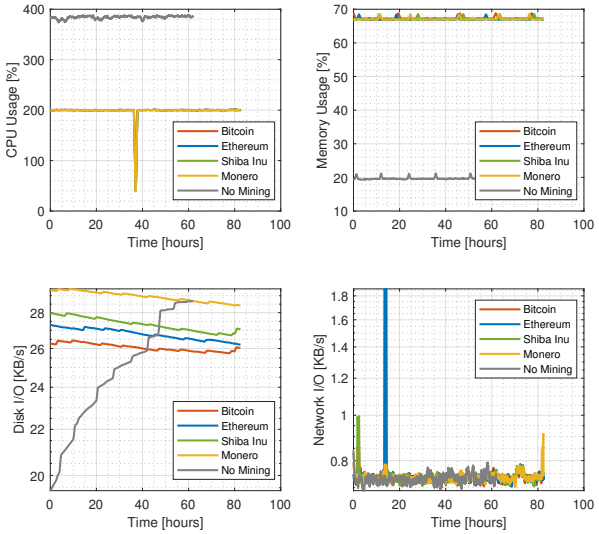


Fig. 2: Resource consumption signatures.

activity maintaining 20-50% utilization levels. Memory consumption displays clear discriminative characteristics, with mining activity consistently operating within 60-70% memory utilization while legitimate activity remains within 20-25% ranges.

Each cryptocurrency demonstrates unique resource fingerprints, with Bitcoin and Ethereum exhibiting the most intensive CPU consumption patterns, while Shiba Inu and Monero display moderate but distinguishable signatures. Disk I/O activity maintains relatively stable patterns across all scenarios but preserves subtle yet measurable differences that contribute to classification accuracy. Network I/O patterns show minimal variation between mining and legitimate activities, confirming that network-based detection approaches face inherent limitations in discriminating between operational categories.

Data Pre-Processing. The feature engineering methodology transforms raw resource utilization metrics into structured temporal sequences, preserving important time-dependent patterns essential for accurate cryptojacking detection. Resource utilization data is preprocessed through sliding-window techniques, creating fixed-length sequences of 50 timesteps with configurable overlap ratios. Each temporal sequence encompasses seven system metrics: CPU utilization percentage, memory consumption percentage, memory usage in kilobytes, network input/output rates, and disk input/output operations. Sequential preprocessing includes comprehensive data cleaning, missing value imputation, and standardized normalization through StandardScaler implementation to ensure optimal neural network convergence characteristics.

Deep Learning Architecture. To identify the most suitable DL architecture, we evaluate four distinct models optimized for different computational constraints and deployment scenarios: The *Dense Network* employs sequential fully

connected layers with 64 and 32 neurons, achieving 0.999 test accuracy with 24,676 parameters. The *Convolutional Neural Network (CNN)* implements two one-dimensional convolutional layers with 16 and 32 filters, achieving 1.0 test accuracy using only 4,548 parameters through strategic batch normalization and global average pooling. The *LSTM* addresses temporal dependency modeling through 32 memory units, utilizing 6,308 parameters while achieving 1.0 classification accuracy. The *CNN-LSTM* combines convolutional pattern recognition with temporal modeling strength, employing 7,876 parameters while achieving 1.0 accuracy.

Training and Validation. The train-test split is set to an 80/20 ratio, ensuring balanced representation across all cryptocurrency categories. Each architecture employs Adam optimization with adaptive learning rate scheduling, categorical cross-entropy loss functions, and early stopping mechanisms. Training configurations utilize batch sizes of 64 samples with maximum epochs set to 50, implementing systematic convergence monitoring for optimal model performance.

V. Performance Evaluation

The evaluation framework comprises three experiments capturing the complete spectrum of adversarial scenarios from our threat model. The baseline Client-Server experiment establishes legitimate operational patterns using Flask web server containers with dynamic client scripts generating computational workloads through Fibonacci calculations and API communications. The Single Crypto-Mining Container experiment isolates pure cryptojacking behavior by deploying dedicated mining containers for Bitcoin, Ethereum, Shiba Inu, and Monero across separate VM instances. The crypto-mining and Client-Server experiment implements obfuscation strategies by simultaneously deploying legitimate workload containers alongside crypto-mining operations.

Each experimental configuration was executed for 168 hours across independent VM instances, generating datasets ranging from 222,841 baseline samples to approximately 300,000 samples per cryptocurrency mining scenario, providing sufficient data volume for model validation and performance assessment. Bitcoin and Ethereum exhibit the most intensive CPU consumption patterns, while Shiba Inu and Monero display moderate but distinguishable signatures. Disk I/O operations exhibit relatively stable patterns across scenarios, while preserving measurable differences that contribute to classification accuracy. Network I/O patterns show minimal variation between mining and legitimate activities, confirming that network-based detection approaches face inherent limitations in discriminating between operational categories.

DL Architecture Performance Assessment. Evaluation across all proposed architectures demonstrates strong performance, with four models achieving 1.0 test accuracy in cryptocurrency classification tasks. Table I presents a detailed performance comparison across four distinct NN architectures, revealing significant variations in computational efficiency and training requirements. The CNN architecture achieves optimal performance with 4,548 parameters, trained over 30 epochs, thereby establishing an effective balance between detection accuracy and computational efficiency. This lightweight design enables practical deployment in resource-constrained cloud environments where operational costs directly correlate with computational overhead. Our DL architectures automatically extract discriminative patterns from raw temporal sequences without manual feature engineering, simplifying deployment while maintaining robust detection capabilities.

TABLE I: Performance comparison of DL architectures in controlled experimental environment.

Architecture	Accuracy	Parameters	Training Epochs
Dense	0.99	24,676	21
CNN	1.0	4,548	30
LSTM	1.0	6,308	31
CNN-LSTM	1.0	7,876	50

Unlike ensemble methods (e.g., Random Forest [10], [11]) requiring extensive feature engineering, our DL architectures automatically extract relevant patterns from raw temporal sequences, reducing preprocessing complexity while improving detection accuracy. The lightweight nature of our proposed architectures addresses critical deployment constraints in cloud environments where computational efficiency directly impacts operational costs.

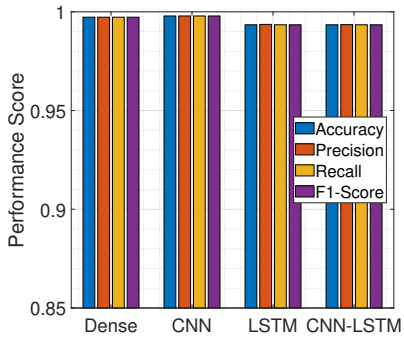


Fig. 3: Weighted average of accuracy, precision, recall, and F1-score across all DL architectures.

Fig. 3 illustrates the weighted average across multiple evaluation dimensions, demonstrating consistent performance in accuracy, precision, recall, and F1-score metrics across all architectures. The performance consistency validates the discriminative power of our sequential feature extraction methodology, while highlighting the efficiency characteristics of convolutional approaches for temporal pattern recognition in resource utilization data.

Obfuscated Scenario Performance Analysis. Table II presents performance metrics for our most challenging evaluation scenario, where crypto-mining activities operate concurrently with legitimate computational workloads designed to obscure malicious signatures.

TABLE II: Performance evaluation for obfuscated crypto-mining detection with concurrent legitimate and malicious container deployment.

Currency	Precision	Recall	F1 Score	TPR	FPR
BTC	0.98897	0.99702	0.99298	0.99702	0.00785
ETH	0.98460	0.98551	0.98506	0.98551	0.00385
SHIB	0.98016	0.97308	0.97661	0.97308	0.00492
XMR	0.98236	0.98052	0.99987	0.98144	0.00440
OS	0.99997	1.000	0.99998	1.000	6.73×10^{-6}

Performance in obfuscated scenarios demonstrates strong capabilities against sophisticated obfuscation in our experimental environment. Legitimate activity detection maintains near-optimal performance with false positive rates below 0.001%, ensuring practical viability in production environments. Cryptocurrency-specific detection maintains precision levels exceeding 97% across all tested scenarios, validating the generalizability of our approach beyond specific mining algorithms or network protocols.

Fig. 4 presents Receiver Operating Characteristic (ROC) curve analysis demonstrating strong discrimination capabilities across all cryptocurrency categories and legitimate activities in our controlled evaluation. The Area Under the Curve (AUC) values of 1.0 for all tested scenarios provide evidence of our framework's discriminative ability under these experimental conditions, indicating strong separability between legitimate and malicious operational categories within our testbed.

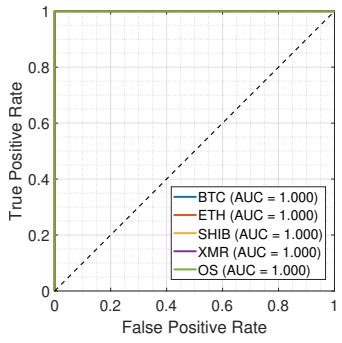


Fig. 4: CNN model ROC-AUC for all cryptocurrencies, demonstrating effective discrimination between legitimate activities and crypto-mining operations.

These ROC curves further validate the framework's discriminative ability, with AUC scores of 1.000 across all evaluation scenarios in our controlled environment. This performance level indicates that our approach achieves effective separability between legitimate and malicious activities in our experimental testbed, providing a promising foundation for further validation toward production deployment.

Moreover, confusion matrix analysis reveals the precision of our classification framework across diverse cryptocurrency mining scenarios. Fig. 5 demonstrates highly accurate classification for legitimate activities with zero false negatives in our evaluation dataset, ensuring that normal operational patterns in our testbed are correctly classified. Inter-cryptocurrency classification maintains high precision, with minimal cross-classification errors that do not impact the fundamental discrimination between legitimate and mining activities.

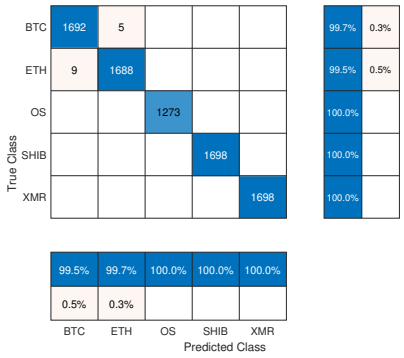


Fig. 5: Confusion matrix for CNN architecture.

The confusion matrix validates the detection framework's effectiveness in our controlled environment, demonstrating that the CNN architecture maintains strong discrimination between legitimate operations and crypto-mining activities while achieving high accuracy in cryptocurrency-specific classification tasks. These results provide a solid foundation for subsequent validation in production environments.

A. *Limitations and Considerations*

While our experimental results demonstrate robust detection within controlled environments, several important limitations require consideration for practical deployment scenarios.

Experimental Environment Constraints. Our evaluation employs standardized VM configurations (4-core CPU, 4 GB RAM, 25 GB storage, Ubuntu 22.04.4 LTS), enabling systematic performance assessment and reproducible validation. However, cloud infrastructures present substantially greater heterogeneity across hardware configurations, operating system versions, and virtualization technologies. The consistently high accuracy observed may partially reflect our experimental homogeneity, requiring additional validation across diverse cloud platforms to establish generalization capabilities.

Workload Diversity and Representativeness. Our workload baseline comprises Flask web server containers generating computational patterns through Fibonacci calculations and API communications. Production environments host a broader range of applications, including scientific computing, video encoding, database systems, and machine learning pipelines. Certain legitimate applications, particularly intensive numerical computation or parallel processing workloads, may exhibit resource patterns similar to crypto-mining activities. Future work should incorporate extended workload diversity to validate detection robustness across broader operational scenarios.

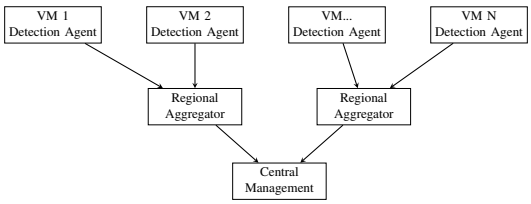


Fig. 6: Hierarchical deployment architecture for scalable cryptojacking detection across distributed cloud infrastructure.

Temporal and Statistical Validation. Data collection spans 168-hour periods per configuration, providing sufficient temporal coverage for model training. However, production environments exhibit dynamic variations such as diurnal usage patterns and transient performance anomalies not fully captured in our experiments. The observed 100% accuracy across four NN architectures warrants careful interpretation regarding potential overfitting risks. Future validation should incorporate k-fold cross-validation, leave-one-cryptocurrency-out evaluation, and testing against diverse workload categories to strengthen generalization.

Adversarial Robustness and Evasion. Our obfuscated scenario implements concealment strategies through concurrent legitimate and crypto-mining container deployment. Adaptive adversaries might employ more sophisticated techniques, including dynamic mining rate throttling, randomized resource consumption patterns, or mimicry attacks emulating legitimate signatures. Future work should incorporate systematic red-team exercises quantifying detection resilience under adversarial conditions.

Deployment Scalability. For large-scale deployments monitoring thousands of VM instances, hierarchical monitoring architecture as illustrated in Fig. 6 would be needed. Each compute node deploys a lightweight detection agent collecting resource utilization metrics through standard Linux utilities (e.g., mpstat, free, sar, iostat) at configurable sampling intervals. Local agents perform sliding-window preprocessing and feature extraction, generating sequential input representations for model inference. Detection agents execute inference operations locally on each VM, generating binary classification outputs (legitimate vs. crypto-mining) along with confidence scores and cryptocurrency type predictions for positive detections. Aggregation servers collect detection events from distributed agents, performing alert correlation, false positive filtering, and statistical anomaly aggregation.

VI. Conclusion

This paper presents a lightweight Deep Learning framework for cryptojacking detection in cloud containerized environments, achieving high classification accuracy with minimal computational overhead in controlled experimental conditions. Our CNN architecture achieves 1.0 test accuracy using only 4,548 parameters while maintaining > 0.97 precision in obfuscated scenarios and $< 0.001\%$ false positive rates. The framework's resource utilization-based approach eliminates performance overhead associated with system call monitoring or network traffic analysis, enabling practical deployment in production cloud environments.

The demonstrated effectiveness against obfuscation techniques within our experimental testbed establishes this methodology as a promising advancement in cloud security. Our approach provides practitioners with efficient tools for protecting containerized infrastructures while supporting sustainable computing practices through proactive prevention of unauthorized resource consumption. Future work will focus on extensive validation across heterogeneous production environments and diverse workload scenarios to further establish generalization capabilities.

Acknowledgment

This work is supported in parts by the Swedish Research Council (VR) and the Knut and Alice Wallenberg (KAW) Foundation.

References

[1] DP VS et al., "Container security: Precaution levels, mitigation strategies, and research perspectives," *Computers & Security*, vol. 135, p. 103490, December 2023.

[2] Wong et al., "On the Security of Containers: Threat Modeling, Attack Analysis, and Mitigation Strategies," *Computers & Security*, vol. 128, p. 103140, May 2023.

[3] M. Mounesan, H. Siadati, and S. Jafarikhah, "Exploring the Threat of Software Supply Chain Attacks on Containerized Applications," in *2023 16th International Conference on Security of Information and Networks (SIN)*. IEEE, 2023, pp. 1–8.

[4] Huang et al., "Security Analysis and Threats Detection Techniques on Docker Container," in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. IEEE, December 2019, pp. 1214–1220.

[5] S. Sultan, I. Ahmad, and T. Dimitriou, "Container Security: Issues, Challenges, and the Road Ahead," *IEEE Access*, vol. 7, pp. 52 976–52 996, 2019.

[6] Kommula et al., "Machine Learning Techniques to Enhance Container Network Security," in *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, December 2020, pp. 622–627.

[7] Abdelrahim et al., "Crypto Currency Cloud Mining," in *2022 International Conference on Artificial Intelligence in Everything (AIE)*. IEEE, 2022, pp. 488–492.

[8] K. et al., "Cryptomining Detection in Container Clouds Using System Calls and Explainable Machine Learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 3, pp. 674–691, March 2021.

[9] Dong et al., "Identification technique of cryptomining behavior based on traffic features," *Frontiers in Physics*, vol. 11, p. 1269889, September 2023.

[10] A. M. Hussain, G. Oligeri, and T. Voigt, "The Dark (and Bright) Side of IoT: Attacks and Countermeasures for Identifying Smart Home Devices and Services," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage (SpaCCS) 2020 International Workshops*. Springer, 2021, pp. 122–136.

[11] A. Hussain, "The Dark Side of IoT: Revealing Activities in the Smart Home Environment using Machine Learning," 2020. [Online]. Available: https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-429112

[12] Li et al., "Malicious mining code detection based on ensemble learning in cloud computing environment," *Simulation Modelling Practice and Theory*, vol. 113, p. 102391, December 2021.

[13] Lpms, Mend, and Snyk, "GitHub - lpsm-dev/docker-crypto-miner: A user-friendly image that can be used for mining cryptocurrencies with your CPU," https://github.com/lpsm-dev/docker-crypto-miner, 2023, accessed 08-09-2024.