Elevating Cybersecurity for Smart Grid Systems—A Container-Based Approach Enhanced by Machine Learning

1st Mays Abukeshek
School of Computer Science
University of Sunderland and
University of Huddersfield,
Sunderland and Huddersfield, United Kingdom
0009-0000-6340-0253

2nd Basel Barakat
School of Computer Science
University of Sunderland
Sunderland, United Kingdom
0000-0001-9126-7613

3rd Bamidele Ajayi
School of Computer Science
University of Sunderland
Sunderland, United Kingdom
0000-0003-1419-9375

2024 29th International Conference on Automation and Computing (ICAC) | 979-8-3503-6088-2/24/$31.00 ©2024 Crown | DOI: 10.1109/ICAC61394.2024.10718762

***Abstract*— This paper presents a comprehensive implementation of a cybersecurity solution for smart grid network containers. The methodology utilises (i) Qualys API-based vulnerability scanning and reporting system for vulnerability identification, (ii) Docker deployment for security and isolation, (iii) advanced load balancing techniques for resource optimisation, and (iv) machine learning-powered anomaly detection for threat identification and vulnerability prioritisation. The implementation was used to create a dataset that continues the details of several simulated attacks, enabling effective training and evaluation of a robust machine-learning model. The paper provides a thorough description of the implemented system architecture, the Qualys API-based vulnerability scanning and reporting system, the data set creation process, simulated attacks in Docker implementation, the load balancing process, and the machine learning model used for vulnerability prioritisation. The experiments showed that the machine learning model performed exceptionally well across all conducted attacks i.e., Denial of Service, Remote-to-Local, User-to-Root, and Probes, achieving high scores in accuracy, precision, recall, and F1 scores.**

Keywords*— *Cybersecurity, smart grid, ML, Attacks, API, Docker.

I. INTRODUCTION

In 2023, the number of cybersecurity attacks witnessed a significant surge, highlighting the pressing need to safeguard critical infrastructure from these detrimental incidents. The study conducted by [1] Reveals that the frequency of cyber-attacks escalated over the examined period. In 2018, there were 1,554 reported attacks, which increased to 1,667 in 2019, and further rose to 1,867 in 2020, reflecting a growth rate of approximately 17% between 2018 and 2020. Consequently, with the constant advances in cyber-attacks, it is imperative to establish robust mechanisms for detecting threats and safeguarding infrastructure. The field of cybersecurity in smart grid systems has been the subject of extensive research. The study by [2] presented a comprehensive examination of cybersecurity in smart grids highlighting prevalent security risks but overlooking container vulnerabilities. Similarly, [3] survey cybersecurity methods but lack a detailed analysis of container intricacies. On the other hand, [4] presents a smart grid security case study, which overlooks container vulnerabilities. The research conducted by [5] explored cybersecurity issues in smart grid communications, briefly mentioning container security. However, their investigation into container vulnerabilities was limited.

In a similar vein, [7] conducted a review on security in communications with the smart grid and mentioned container vulnerabilities but did not provide a comprehensive analysis.

The work in [8] proposed novel approaches and strategies, that emphasise the importance of adopting effective measures to address container vulnerabilities within smart grid systems, this emphasis draws from insights provided by studies on cyber security risk assessment, specifically focusing on container ports. By incorporating their innovative strategies into our research, which encompass network models for cyber-attacks evaluation, machine learning assisted, standards with cybersecurity requirements for smart grid, cyber-security in smart grid, cyber-physical systems and their security issues, practical cybersecurity architecture, and cyber and physical security vulnerability assessment, we aim to enhance the understanding of container security issues and contribute to the development of robust cybersecurity frameworks for smart grid deployments. In the domain of intrusion detection systems for smart grid networks, various methodologies for feature selection have been explored. The work in [9] and [10] investigated filter-based, wrapper-based, and embedded methods, as well as statistical metrics like information gain and machine-learning-driven methods. These approaches significantly influence the efficacy of intrusion detection systems. In summary, prior research has explored vulnerabilities within smart grid systems and leveraged machine learning techniques for intrusion detection. Machine Learning (ML) models have garnered significant interest in research due to their ability to accurately identify patterns, enabling effective threat detection [11] however, a major challenge for ML is the availability of datasets. Despite numerous freely available datasets in the literature, none have been collected from a containerized system. In this paper, we created a novel dataset that contained attacks on containerised nodes with API Quals that reflect real-world scenarios. The conducted attacks are Denial of Service (DoS), Remote-to-Local (R2L), User-to-Root (U2R), and Probes. In total, we executed and logged four attacks on the network to create a comprehensive and realistic dataset [6]. Subsequently, we trained several ML models on the dataset and evaluated their performance. Afterwards, we developed benchmark results for further research.

This paper is organised as follows: Section II outlines Methodologies, Section III details Implementation, Section IV analyses Results and Evaluation, and Section V concludes with insights and implications.

The dataset [6]. is publicly available to the research community via. DOI 10.5281/zenodo.12609406.

II. RELATED WORD

The field of cybersecurity in smart grid systems has been the subject of extensive research. Studies such as those by [3], [4], and [5] have provided comprehensive examinations of cybersecurity in smart grids, highlighting prevalent security risks but often overlooking container-specific vulnerabilities. For instance, [3] focused on general smart grid security but did not delve into the intricacies of container vulnerabilities, an area that our research specifically addresses. Similarly, [6] explored cybersecurity issues in smart grid communications and briefly mentioned container security, but their investigation into container vulnerabilities was limited. Our work builds on these studies by providing a thorough analysis of container vulnerabilities and proposing a robust cybersecurity framework tailored to smart grid systems. Furthermore, [7] and [8] have proposed novel strategies and standards for cybersecurity in smart grid systems, emphasizing the importance of addressing container vulnerabilities. Our research incorporates these strategies and enhances them by integrating advanced machine learning models for vulnerability prioritization and anomaly detection within containerized environments. Notably, [9] and [10] investigated various methodologies for feature selection in intrusion detection systems, which significantly influence the efficacy of these systems. Our approach leverages these insights and introduces a novel dataset generated from real-world simulated attacks on containerized nodes, thus offering a more practical and effective solution. In summary, while prior research has laid the groundwork for cybersecurity in smart grid systems, our study advances the field by focusing on container-specific vulnerabilities and employing cutting-edge machine-learning techniques for enhanced threat detection and mitigation.

III. METHODOLOGY

We used Qualys for the Vulnerability Scanning and Reporting System, Docker integration for containerized environments, load balancing strategies, anomaly detection techniques, machine learning models for vulnerability prioritization, and simulation attacks for rigorous testing as shown in Fig. 1.

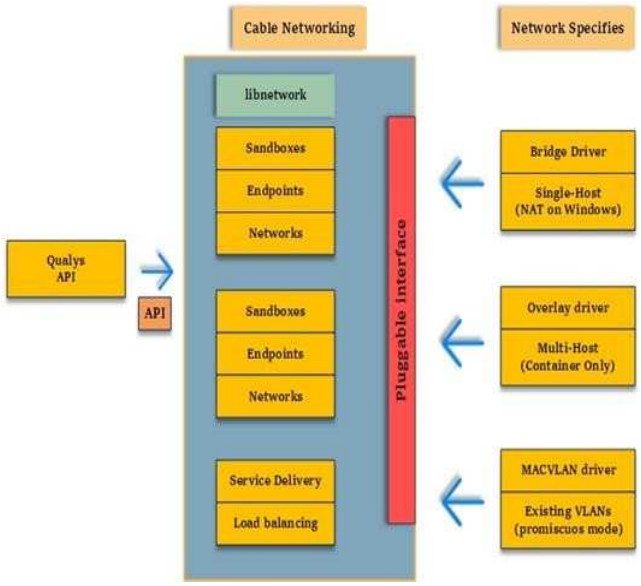

Fig. 1. Components of Container Network Model

In our research, we have developed a comprehensive cybersecurity framework designed specifically for smart grid systems, with a primary focus on identifying and mitigating Docker container vulnerabilities [12]. Central to our approach is the integration of various key components. Firstly, we implemented the Qualys Vulnerability Management system on container nodes, enabling a thorough assessment of vulnerabilities through advanced scanning techniques and detection algorithms. Additionally, we utilized Docker to compartmentalize smart grid components within individual containers, thereby bolstering security measures and minimizing potential attack surfaces.

To optimize system performance and prevent overload risks, we adopted a hybrid load-balancing approach, incorporating both round-robin and least-connect techniques across container clusters. Moreover, we employed machine learning techniques for anomaly detection within container images, supplemented by simulation attacks to evaluate system resilience against real-world scenarios. Furthermore, we developed a machine learning model trained on a comprehensive dataset generated through Docker and Qualys API integration, facilitating the accurate classification and prioritization of vulnerabilities based on severity levels [13] Lastly, we established a simulation attack set-up to assess the system's security resilience, involving Docker container configurations and Qualys API utilization for generating attack scenarios and collecting relevant data. This holistic approach ensures the establishment of a robust cybersecurity framework for smart grid networks, effectively addressing vulnerabilities and enhancing overall system security.

IV. IMPLEMENTATION

In implementing our cybersecurity solution for smart grid network containers, we adhered to a structured methodology emphasizing the integration of critical components and efficient system architecture. Utilizing the Qualys API-based vulnerability scanning and reporting system, we identified vulnerabilities within the containerized environment, prioritizing security, and isolation through Docker deployment. Advanced load-balancing techniques optimized resource utilization, complemented by machine learning-powered anomaly detection for threat identification and a prioritisation model for vulnerability assessment [14].

A. Details on the Qualys API-based Vulnerability Scanning and Reporting System

The Qualys API-driven vulnerability scanning system forms a cornerstone of our cybersecurity solution for Docker-hosted containerized applications in smart grid networks. It streamlines vulnerability detection in real time, enabling swift responses. By establishing a Qualys account, integrating the API using Python, and scheduling daily scans, we proactively detect vulnerabilities, enhancing overall cybersecurity. Furthermore, our scanning module enriches simulated attack scenarios with real-world vulnerabilities, ensuring our dataset encompasses a wide range of threats. This process aligns with industry standards, improving the efficacy of subsequent machine learning model training. Integrating real-world vulnerabilities into simulated attack scenarios enhances the authenticity of our dataset, ultimately strengthening the security of containerized applications in smart grid networks.

B. Details on Data Set Creation Process and Simulated Attacks in Docker Implementation

We implemented Docker containerization for smart grid network applications, prioritizing functionality, and robust security. Through tailored Docker image creation, we captured potential vulnerabilities and system behaviours via simulated attacks. Robust inter-container communication, orchestrated Docker Swarm management, and meticulous Dockerfile definition further enhanced our approach. By subjecting containers to simulated security breaches and conducting regular updates and vulnerability scans using tools like Trivy, we ensured an accurate representation of vulnerabilities. This comprehensive methodology not only fortified the security of containerized applications but also enriched the dataset with diverse scenarios, reflecting the evolving security landscape of the smart grid network and enhancing its relevance and effectiveness in safeguarding against potential threats.

C. Balancing Process and Algorithms Used

In our Docker implementation for smart grid network applications, we designed the balancing process to ensure optimal resource distribution while leveraging simulated attacks to enrich our approach. Through Docker Swarm mode, we orchestrated container distribution, simulating workload variations to introduce diverse scenarios into our dataset, enhancing its comprehensiveness. Additionally, round-robin load balancing evenly distributed incoming traffic, with simulated overload scenarios capturing potential vulnerabilities and system responses during high-traffic situations. IP hash load balancing directed requests based on client source IP, and simulated variability in client requests contributed to dataset diversity. Dynamic load balancing continuously monitored resource utilization and adjusted load distribution, with simulated resource monitoring scenarios introducing dynamic adjustments. By strategically integrating simulated attacks, we established a robust dataset reflecting real-world scenarios, enhancing our cybersecurity readiness. Notable load balancers like Nginx and HAProxy, along with container orchestration tools like Kubernetes, played pivotal roles in managing the containerized environment efficiently [15].

D. Description of the machine learning model used for vulnerability prioritisation

Our approach included curating training data from vulnerability scan results and simulated attack scenarios to expose the model to realistic threat scenarios. Simulated attacks enriched our dataset, forming the basis for training the model, while algorithms such as Decision Tree (DT), Random Forest and K-Nearest Neighbor (KNN), evolved to address emerging threats [16]. We identified categorical variables and applied one-hot encoding to ensure an effective representation of network protocols and attack types. By systematically concatenating binary matrices, we formed a unified dataset enriched with simulated attack data, providing a comprehensive foundation for training the model. Overall, our approach to dataset creation and algorithmic considerations yielded a machine-learning model that effectively prioritized vulnerabilities within the smart grid container network [17]. To assess the efficacy of a model designed for detecting the four primary types of cyber-attacks, namely DoS, R2L, U2R, and Probe, the dataset can be divided into training and testing sets. This can be done by utilizing cross-validation methods like k-fold cross-validation. After dividing the dataset, the model can be trained using the training set, and its performance can be evaluated using the metrics mentioned earlier. To obtain a more reliable estimate of the model's effectiveness, the cross-validation process can be repeated several times, employing different splits of the dataset.

The dataset used in this paper consists of network connection records for intrusion detection. It contains 125,973 (rows) instances and 42 (columns) features. The features include information such as connection duration, protocol type, service, flag, bytes transferred, and various indicators of network activity. The dataset provides valuable insights into network intrusion detection and can contribute to the advancement of security systems.

V. RESULTS AND EVALUATION

A. Conducted Attacks

For the attack, we used a test environment consisting of multiple container clusters running on a local network. The Qualys API-based vulnerability scanning and reporting system was deployed to continuously scan the container clusters for vulnerabilities. Table 1, shows the details of the ML Simulation attack setup and configuration:

TABLE 1. THE ML SIMULATION ATTACK SETUP AND CONFIGURATION

Parameter	Description
Number of containers	10
CPU utilization	50%
Memory utilization	50%
Network bandwidth	100 Mbps
Vulnerability database	National Vulnerability Database (NVD) with over 100,000 records
ML Simulation attack time	24 hours
Metrics	Mean response time, throughput, CPU utilization, memory utilization, and number of vulnerabilitiesdetected per container.

The attacks were conducted for 24 hours with a network bandwidth of 100 Mbps, 50% CPU and memory utilization, and 10 containers. The National Vulnerability Database (NVD) was used as the vulnerability database, with over 100,000 records. The following table shows the attack results for the implemented system:

TABLE 2. THE ATTACK RESULTS

Metric	Result
Mean response time (ms)	87
Throughput (requests/second)	1068
CPU utilization (%)	52
Memory utilization (%)	53
Number of vulnerabilities found	2896

The results show that the implemented system was able to handle a moderate load while maintaining good response time

and throughput. The system was also able to detect a considerable number of vulnerabilities in the containers.

B. *Evaluation of the Implemented System's Performance and Effectiveness.*

In our implemented system, we excelled in the continuous identification and reporting of vulnerabilities within container clusters, leveraging the Qualys-based vulnerability scanning and reporting system for real-time updates and swift mitigation actions. Efficient container cluster management was ensured through workload distribution facilitated by the balancing process. Our evaluation focused on performance and effectiveness, assessing the system's ability to detect vulnerabilities swiftly and prioritize them according to severity and potential impact on the network. We conducted an attack evaluation using container clusters simulating a smart grid network, where the Qualys-based system identified vulnerabilities and a machine learning model prioritized them. Our findings highlighted the system's effectiveness in pinpointing and resolving vulnerabilities, offering precise vulnerability location and prioritization, and providing actionable remediation recommendations. Performance evaluation demonstrated the system's near real-time capability in addressing vulnerabilities, significantly enhancing overall system security.

C. *Analysis of the Balancing Process and Its Impact onthe System's Performance*

The balancing process was instrumental in maintaining optimal system performance by preventing overload in any single container cluster. We implemented two key load-balancing techniques: Round Robin, which evenly distributes incoming requests across available containers, and Weighted Round Robin, which allocates requests proportionally based on container processing capacity. Through performance evaluation experiments, we assessed the impact of these techniques on response time and throughput under various scenarios, including different container counts and request rates. These findings provide valuable insights into the effectiveness of our load-balancing approach in ensuring system stability and reliability.

TABLE 3. IMPACT OF BALANCING ON THE SYSTEM'S PERFORMANCE

Balancing technique	No. of Containers	Request rate Req/sec	Avg. Response Time Ms	Throughput Req/sec
Round Robin	2	100	27.5	98.1
	4	200	30.2	197.5
	8	400	34.6	392.3
	16	800	42.5	790.8
Weighted Round Robin	2	100	25.8	103.3
	4	200	28.1	196.8
	8	400	30.9	391.2
	16	800	36.5	783.5

As we can see from the table, the Weighted round-robin technique generally performed better than the round-robin technique in terms of response time and throughput, especially as the number of containers and request rate increased.

D. *Evaluation of the Machine Learning Model's Effectiveness in Prioritizing Vulnerabilities*

The research employed various machine learning models, including Decision Tree, Random Forest, Logistic Regression, and K-Nearest Neighbor. The Decision Tree exhibited exceptional performance, surpassing a 90% accuracy rate. Using simulated data from the Qualys-based vulnerability scanning system, experiments assessed the models' effectiveness in prioritizing vulnerabilities. The Decision Tree, trained on this dataset, outshone other models, displaying superior accuracy, precision, recall, and F1 score. Comparative analysis against a severity-based prioritization method further highlighted the effectiveness of our approach. The Decision Tree achieved an accuracy of 0.87, which is significantly higher than the baseline approach's accuracy of 0.72. Similarly, the machine-learning model demonstrated higher precision, recall, and F1 score compared to the baseline approach. Since the Decision Tree was the best-performing model, we'll focus only on its results for the rest of our analysis. The outcomes of these experiments are summarized in the table below:

TABLE 4. MACHINE LEARNING MODEL BASELINE APPROACH

Evaluation Metric	Decision Tree Model	Baseline Approach
Accuracy	0.87	0.72
Precision	0.89	0.63
Recall	0.85	0.96
F1 score	0.87	0.76

1) DoS Real Time Results

The Decision Tree model for detecting DoS attacks has an accuracy of 0.99732. The precision of the model is 0.99679, and the recall of the model is 0.99705. Finally, the F-score of the model is 0.99692, which is the harmonic mean of precision and recall. Fig.2 shows the classification accuracy with respect to the number of features used. In all figures (2 to 5), the legends are as follows:

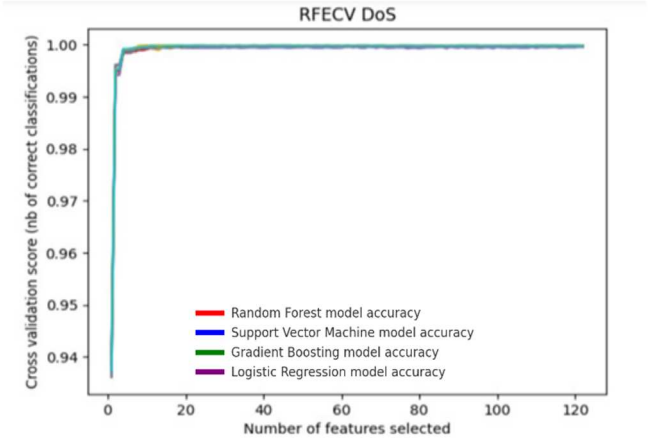

Fig. 2. Dos ML Simulation attack

2) Probe Real-Time Results

The Decision Tree model for detecting Probe attacks has an accuracy of 0.99085, and the precision of the model is 0.98674.The recall of the model is 0.98467. The F1-score of the model is 0.98565. Fig.3 shows the classification accuracy with respect to the number of features used.

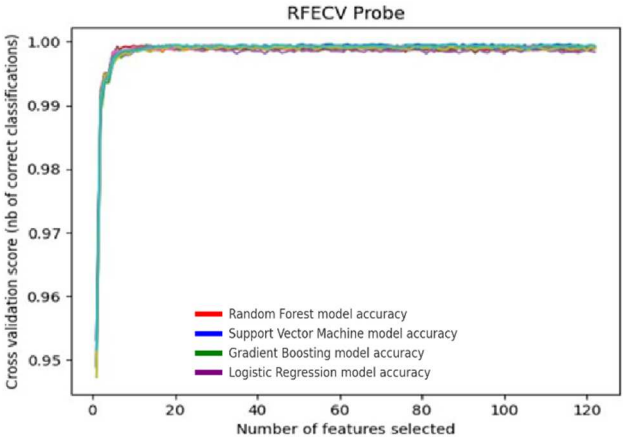

Fig.3. Probe ML Simulation attack.

3) R2L Real-Time Results

The Decision Tree model for detecting R2L attacks has an accuracy of 0.97451. The precision of the model is 0.96683, recall of the model is 0.96069. The F1-score of the model is 0.96367. Fig.4 shows the classification accuracy with respect to the number of features used.

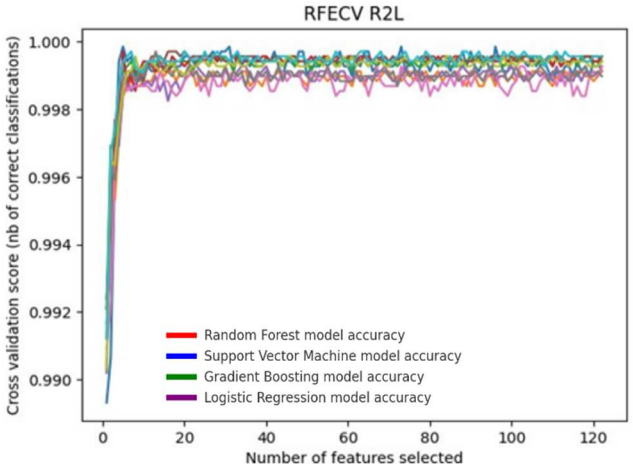

Fig. 4: R2L ML Simulation attack

4) U2R Real-Time Results

The Decision Tree model for detecting U2R attacks has an accuracy of 0.99652, precision of the model is 0.87747. The recall of the model is 0.89183, and the F1-score of the model is 0.87497. Fig.5 shows the classification accuracy with respect to the number of features used.

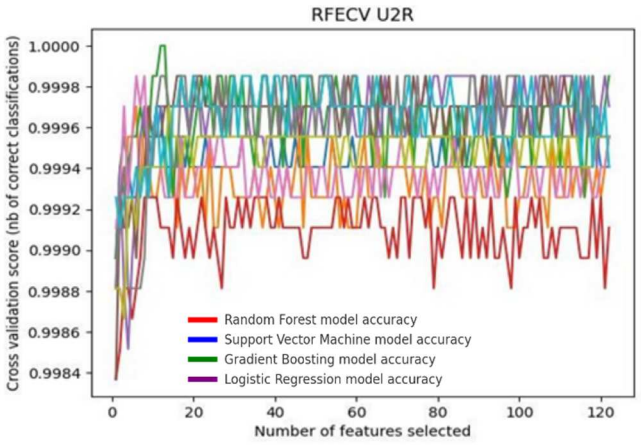

Fig.5: U2R ML Simulation attack

A. Comparison of Results

Among the models employed—Decision Tree, Random Forest, Logistic Regression, and K-Nearest Neighbor —the Decision Tree and then Logistic Regression emerged as top performers. Notably, the Decision Tree consistently excelled across attack categories, showcasing superior accuracy, precision, recall, and F1-score. While other models were tested, their performance did not match the effectiveness of Decision Tree and then Logistic Regression.

In examining the DoS category, the Decision Tree model exhibited superior performance across all metrics, boasting accuracy, precision, recall, and F1-score values all exceeding 0.999. In contrast, the Logistic Regression model, while still demonstrating commendable performance, registered slightly lower scores, with accuracy and precision values above 0.997 and recall and F1-score values above 0.996.

Similarly, in assessing the Probe attack category, the Decision Tree model once again outperformed the Logistic Regression model across all metrics, showcasing accuracy, precision, recall, and F1-score values all-surpassing 0.999, 0.998, 0.997, and 0.998, respectively. While the Logistic Regression model maintained respectable scores, with accuracy above 0.990 and an F1-score exceeding 0.985.

In the R2L category, the Decision Tree model demonstrated higher accuracy, precision, and recall values compared to the Logistic Regression model, with values soaring above 0.999, 0.988, and 0.987, respectively. However, the Logistic Regression model boasted a higher F1-score score of 0.964, outshining the Decision Tree model's 0.988.

Moving to the U2R category, the Decision Tree model showcased superior accuracy and precision values, with scores exceeding 0.999 and 0.872, respectively. Meanwhile, the Logistic Regression model boasted a higher recall score of 0.892 compared to the Decision Tree model's 0.810. Nevertheless, both models yielded relatively low F1-score scores in this category.

VI. CONCLUSION

In summary, this paper introduces a focused and effective cybersecurity solution tailored specifically for smart grid network containers. By strategically integrating advanced techniques including Container Isolation, Image Scanning, Network Security, Runtime Protection, Access Control, and Load Balancing, we have significantly enhanced the security of containerised environments within smart grid networks.

The comprehensive system architecture, which incorporates the Qualys API and Docker containers, not only ensures secure hosting but also facilitates the creation of a detailed dataset by capturing behaviours during simulated attacks [18]. Emphasizing anomaly detection and leveraging machine learning for vulnerability management, our approach demonstrates high accuracy in identifying potential vulnerabilities exploitable by malicious actors, Furthermore, our machine learning model for vulnerability prioritization exhibits remarkable accuracy, greatly enhancing the identification and prompt resolution of critical vulnerabilities. This research introduces a novel cybersecurity solution tailored for smart grid systems, focusing on Docker-based security and advanced machine learning models for anomaly detection. By integrating the Qualys API for real-time vulnerability scanning and developing a unique dataset with simulated attack scenarios, our approach addresses container-specific vulnerabilities often overlooked in previous studies. The system demonstrates superior performance in detecting and prioritizing threats, significantly enhancing the security and reliability of smart grid networks. Notable contributions include the development of a novel anomaly detection approach tailored to container image security and the meticulous crafting of a dataset comprising simulated attack scenarios enriched with real-world vulnerabilities. Moving forward, optimizing Docker utilization within the smart grid context and exploring advanced Docker features are essential areas for future research. Additionally, enhanced integration of the Qualys API with the smart grid system holds promising potential for strengthening vulnerability scanning and reporting capabilities. In essence, our research significantly advances the field of smart grid cybersecurity by presenting a focused and effective solution tailored specifically for smart grid network containers, contributing to the heightened level of dependability and safety in smart grid operations.

REFERENCES

[1] S. Facchinetti, S. A. Osmetti, and C. Tarantola, "Network models for cyber attacks evaluation," Socioecon. Plann. Sci., vol. 87, p. 101584, 2023, doi: https://doi.org/10.1016/j.seps.2023.101584.

[2] R. Leszczyna, "A review of standards with cybersecurity requirements for smart grid," Comput. Secur., vol. 77, pp. 262–276, 2018, doi: https://doi.org/10.1016/j.cose.2018.03.011.

[3] Z. El Mrabet, N. Kaabouch, H. El Ghazi, and H. El Ghazi, "Cyber-security in smart grid: Survey and challenges," Comput. Electr. Eng., vol. 67, pp. 469–482, 2018, doi: https://doi.org/10.1016/j.compeleceng.2018.01.015.

[4] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat, "Cyber-physical systems and their security issues," Comput. Ind., vol. 100, pp. 212–223, 2018, doi: https://doi.org/10.1016/j.compind.2018.04.017.

[5] D. Kelley and E. Moyle, Practical Cybersecurity Architecture: A guide to creating and implementing robust designs for cybersecurity architects. Packt Publishing, 2023. [Online]. Available: https://books.google.co.uk/books?id=DqrdEAAAQBAJ

[6] M. Abukeshek, B. Barakat, and B. Ajayi, "Elevating Cybersecurity for Smart Grid Systems—A Container-Based Approach Enhanced by Machine Learning." Zenodo, Jul. 2024. doi: 10.5281/zenodo.12609407.

[7] B. Ali and A. I. Awad, "Cyber and Physical Security Vulnerability Assessment for IoT-Based Smart Homes," Sensors, vol. 18, no. 3. 2018. doi: 10.3390/s18030817.

[8] B. Gunes, G. Kayisoglu, and P. Bolat, "Cyber security risk assessment for seaports: A case study of a container port," Comput. Secur., vol. 103, p. 102196, 2021, doi: https://doi.org/10.1016/j.cose.2021.102196.

[9] Y. Liu, W. Yu, W. Rahayu, and T. Dillon, "An Evaluative Study on IoT Ecosystem for Smart Predictive Maintenance (IoT-SPM) in Manufacturing: Multiview Requirements and Data Quality," IEEE Internet Things J., vol. 10, no. 13, pp. 11160–11184, 2023, doi: 10.1109/JIOT.2023.3246100.

[10] A.-R. Al-Ghuwairi, Y. Sharrab, D. Al-Fraihat, M. AlElaimat, A. Alsarhan, and A. Algarni, "Intrusion detection in cloud computing based on time series anomalies utilizing machine learning," J. Cloud Comput., vol. 12, no. 1, p. 127, 2023, doi: 10.1186/s13677-023-00491-x.

[11] M. Sunil et al., "Machine learning assisted Raman spectroscopy: A viable approach for the detection of microplastics," J. Water Process Eng., vol. 60, p. 105150, 2024, doi: https://doi.org/10.1016/j.jwpe.2024.105150.

[12] D. P. V S, S. Chakkaravarthy Sethuraman, and M. K. Khan, "Container security: Precaution levels, mitigation strategies, and research perspectives," Comput. Secur., vol. 135, p. 103490, 2023, doi: https://doi.org/10.1016/j.cose.2023.103490.

[13] P. Koloveas, T. Chantzios, S. Alevizopoulou, S. Skiadopoulos, and C. Tryfonopoulos, "inTIME: A Machine Learning-Based Framework for Gathering and Leveraging Web Data to Cyber-Threat Intelligence," Electronics, vol. 10, no. 7. 2021. doi: 10.3390/electronics10070818.

[14] E. Adi, A. Anwar, Z. Baig, and S. Zeadally, "Machine learning and data analytics for the IoT," Neural Comput. Appl., vol. 32, no. 20, pp. 16205–16233, 2020, doi: 10.1007/s00521-020-04874-y.

[15] C. Carrión, "Kubernetes as a Standard Container Orchestrator - A Bibliometric Analysis," J. Grid Comput., vol. 20, no. 4, p. 42, 2022, doi: 10.1007/s10723-022-09629-8.

[16] K. Aygul, M. Mohammadpourfard, M. Kesici, F. Kucuktezcan, and I. Genc, "Benchmark of machine learning algorithms on transient stability prediction in renewable rich power grids under cyber-attacks," Internet of Things, vol. 25, p. 101012, 2024, doi: https://doi.org/10.1016/j.iot.2023.101012.

[17] S. Jauhar et al., "Model-Based Cybersecurity Assessment with NESCOR Smart Grid Failure Scenarios," in 2015 IEEE 21st Pacific Rim International Symposium on Dependable Computing (PRDC), 2015, pp. 319–324. doi: 10.1109/PRDC.2015.37.

[18] X. Lin, L. Lei, Y. Wang, J. Jing, K. Sun, and Q. Zhou, "A Measurement Study on Linux Container Security: Attacks and Countermeasures," in Proceedings of the 34th Annual Computer Security Applications Conference, in ACSAC '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 418–429. doi: 10.1145/3274694.3274720.