Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2024, 11(2):213-221

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Firmware Security with ML: Explore Machine Learning Techniques for the Analysis of Firmware in Semiconductor Devices to Uncover Potential Security Flaws and Backdoors

Rajat Suvra Das

Senior Director, Business Development
L&T Technology Services
Email: rajat.tel@gmail.com

Abstract In recent times, Firmware remains an acute part of the progression of semiconductor devices, and some flaws or mischievous code can stand out as a substantial threat to security. Certain hardware vulnerabilities are difficult to mitigate entirely, deprived of releasing a different generation of mechanisms, whereas others can be secured in firmware. Besides, the low-level program design exists in hardware ICs. Besides, patching is not direct; as a result, such flaws may affect real-time devices constantly. To overcome the potential vulnerabilities that occur during manual firmware processing, many ML algorithms can be explored to enhance firmware security. The presented reviews intend to examine the implementation of Machine Learning (ML) methods in evaluating firmware in semiconductor devices to detect and interpret potential security vulnerabilities with indefinite backdoors. Moreover, it includes the collection of different firmware trial datasets from various semiconductor devices and emergent ML methods to examine and classify the firmware grounded with security features. The existing models can be trained with labeled data to detect designs, variances, and latent indicators for security flaws or backdoors. The presented review provides a valued perception of the potential security threats related to firmware in semiconductor devices. Through leveraging ML methods, it remains predictable, and formerly hidden security flaws and backdoors can be exposed, facilitating positive methods to be engaged to lessen the threats.

Keywords Firmware, Security Flaws, Semiconductor Devices, Vulnerabilities, Potential Security, Threats

1. Introduction

In modern computing, hardware instruments are manufactured at different levels in various locations. Hence, the hardware elements inherited in the mixed-trust computing circumstances are frequently shared between the execution contents of various security levels in a consecutive method. Furthermore, Firmware is a so-called microcode, an instruction set that empowers devices to execute the labeled tasks. Distinct from software generated for particular functions, firmware is accountable for functioning the basic functions of a device. Moreover, it is fundamentally the program design that permits the device to run effortlessly. Through upgrading, the firmware has numerous benefits, such as functionality improvement, which helps the devices to be viable with the different models. Besides, security experts have generated a tremendous determination to mount operative hardware security countermeasures. Moreover, recent advancements in Machine Learning (ML) and Artificial Intelligence (AI) have demonstrated the potential to implement accurate detection solutions [1].

In accordance with the prevailing hardware security threats, namely, hardware Trojan, Covert, side channels, and Reverse Engineering (RE) are consistently arising; recent potent attacks exploit remote, cross-layer and specification-compatible attack surfaces to compromise robust cryptographic primitives, memory protection methodology, isolation mechanism and Deep Neural Networks (DNN). Even with different protection methods for preventing hardware security threats. According to the existing study, the hardware attacks are generally considered to be interfering with a bus on a PCB board or in measuring an Integrated Circuit (IC) power consumption factors such as DPA and SPA to extract its cryptographic keys with injection faults through different methods like laser power glitches, power glitches and EM attacks towards modifying the running firmware [2]. Correspondingly, silicon security remains an essential and current field that desires attention. Besides, attack detection is considered one of the significant features for hardware experts to reduce the number of major defies and difficulties related to hardware attacks. The requirement for a different, modest with fewer detection methods after the Fault Injection Attacks are used to conduct whereas it is difficult to detect [3]. The threat of PCB Trojans has been unknown for several years. However, study in this zone increases its rapidity. Hence, the proposed countermeasures centered on PUFs, run-time side-channel observing, inter-component encryption, and complication. Considerably, the study has made to consider IC-level Trojans. The threat of board-level Trojans and embeds varies as of IC-level Trojans in numerous concerns[4].

Accordingly, the presented study analyzes numerous ML based structures in detecting security flaws and backdoors. Initially, it deliberates the types of datasets and resources used for detecting the flaws in firmware. Besides, it also explains the ML methodologies utilized for analyzing the firmware security flaws in the existing methods. Considerably, it depicts the challenges faced by the researchers in analyzing firmware in semiconductor devices to uncover potential security flaws and backdoors. It also illustrates future directions to support the researchers in enhancing its efficiency and inappropriate requirements.

The major contribution of the presented review is signified in the following:

- To represent the complexity and the diversity faced by the firmware.
- To state the limitations challenged by the prevailing studies in analyzing the firmware with Machine Learning (ML) to support the imminent studies.
- To demonstrate future information about the firmware with ML that contributes to the analysis of semiconductors to expose the security flaws and the backdoors.

1.1 Paper Organization

The paper is organized as follows: section 2 deliberates the analysis technique, section 3 presents the outline of ML in firmware security, and section 4 signifies the dataset and resources used for analyzing the firmware flaws. Consistently, section 5 demonstrates the ML methodologies used in the security analysis in the firmware section.6 addresses the challenges and future directions the existing studies face, and section 7 expresses the complete conclusion of the projected model.

2. Research Method

The systematic literature review is an important methodical analysis that gathers plentiful studies to produce a sufficient appreciative outcome. The existing study was introduced through the search design by accessing Google Scholar using precise keywords such as “semiconductor devices," security flaws," "vulnerabilities," and so on. The projected study is a collection of papers from 2020 to 2024 by current progressions.

Search Strategy

The search strategy of the study is presented by the extensive range of the files. It is planned by deciding on suitable files. The IEEE Access remains an integrative electronic journal that provides the progressions and results of unique studies. The ensuing signifies the sources employed for the projected study.

1. Google Scholar [www.scholar.google.com.au/]
2. IEEE Access [https://ieeeaccess.ieee.org/]

Insertion Criteria and Rejection Criteria

The section represents the inclusion and exclusion criteria utilized to gather the paper for the study. The figure.1 indicates the important factor in selecting the appropriate papers.

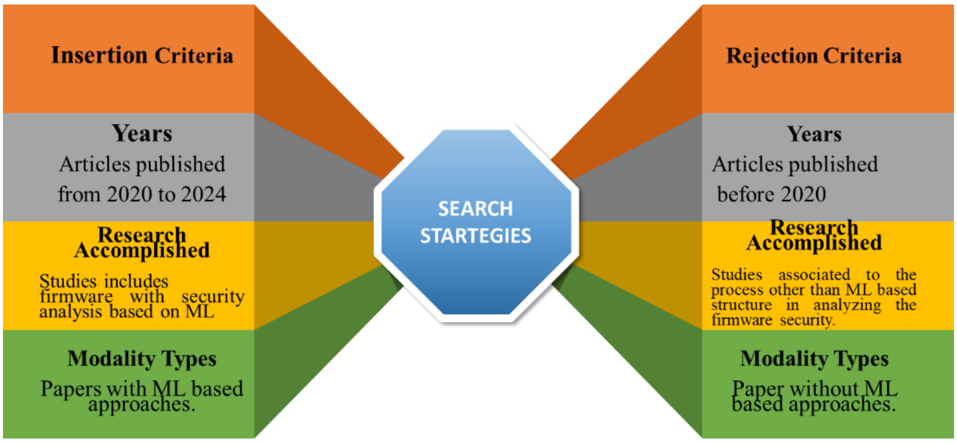
Insertion Criteria
Years
Articles published from 2020 to 2024
Research Accomplished
Studies includes firmware with security analysis based on ML
Modality Types
Papers with ML based approaches.
SEARCH STARTEGIES
Rejection Criteria
Years
Articles published before 2020
Research Accomplished
Studies associated to the process other than ML based structure in analyzing the firmware security.
Modality Types
Paper without ML based approaches.

Figure 1 Insertion and Rejection Criteria

The papers are shortlisted based on the following insertion and rejection criteria.

3. Machine Learning in Firmware Security

ML in firmware security states the application of ML methods to improve the firmware security that remains as software embedded with hardware measures. In leveraging ML procedures, firmware security can be enhanced by identifying and modifying possible vulnerabilities, classifying malicious encryption or behavior, and increasing anomaly detection competencies. Additionally, ML can support the study of firmware updates or reinforcement by robotically detecting potential security risks or compatibility problems. Hence, it can assist in certifying that firmware updates are confident and do not present different vulnerabilities.

However, although ML can be a valued tool in firmware security, it is not a guaranteed solution for difficult problems. It would be utilized and combined with further security methods, namely, consistent security audits, secure coding practices, and firmware reliability checks, to offer complete security in contrast to firmware-based attacks. Analysis, such as static and dynamic analysis, is used for firmware security and is deliberated in the following sub-sections [5].

Treats affect the firmware security.

The threats that affect firmware security are hardware Trojan, Supply-chain vulnerability, Reverse Engineering (RE), and side channel vulnerability, and they are briefly depict**ed** [6].

i) Hardware Trojan is a malicious hardware variation that can leak secret data, reduce the system performance, or lead to denial-of-service.

ii) Supply-chain vulnerability can be classified into three major threats. They are counterfeiting, over-building, and recycling. Besides, IC counterfeiting is a problem that ascends after the worldwide semiconductor supply chain. Moreover, IC overbuilding is a corrupted state where the foundry produces more ICs.

iii) RE of SoC remains an information attack method done by using the backdoors in an SoC, also by defective structure or malicious embedding.

iv) Side channel vulnerabilities arise after the circumstance where the electronic devices unavoidably produce physical discharge in implementation. Besides, it is not restricted to performance period, path-delay power consumption, or electro-magnetic discharge.

3.1 Static analysis

Static analysis [7] is a method utilized to examine the binary or code of firmware deprived of implementation. The static analysis supports classifying potential security defects. For instance, code instillation vulnerabilities, overflows of buffer, and uncertain cryptographic applications are the defects identified in the firmware. By integrating ML such as Neural Network (NN), k-nearest Neighbour (kNN), and Support Vector Machine (SVM) into static analysis, the procedure can be automatic with added effectiveness. In static analysis, ML can also help decrease False Positives (FP) and False Negatives (FN). Moreover, the formal verification, PCB analysis, and heuristic analysis are classified by static analysis.

3.2 Dynamic analysis

Dynamic analysis [8] in firmware security includes examining the performance of firmware in the real world or implementation. Hence, ML can be functional to improve dynamic analysis methods and advance firmware security. Further, algorithms, namely, SVM and kNN, can be utilized in dynamic analysis using training methods to identify the performance of standard firmware. By examining a huge dataset of real firmware implementations, ML algorithms can acquire configurations and features that describe standard performance. Hence, it empowers the recognition of irregularities or abnormalities as predictable comportments that may specify malicious events or prohibited variations.

4. Datasets and Resources

The section deliberates the resources and dataset used in the existing research regarding security flaws analysis within firmware in semiconductor devices. Certain prominent datasets and sources used in the traditional models in this section, such as firmware binaries, open-source tools, and firmware repositories, are included and expressed in Figure 2.

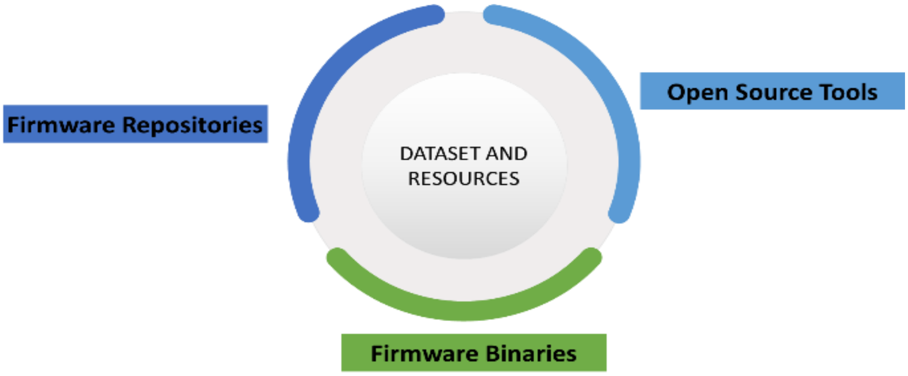

Figure 2: Types of Datasets and Resources

4.1 Firmware Repositories

Firmware repositories are vital in certifying firmware updates' availability, integrity, and security. Efficient Firmware repositories enable the distribution of firmware updates, facilitate collaboration among developers, and help maintain a centralized source of trusted firmware files.

There are various widely obtainable firmware repositories that can be utilized for research determinations. The Security Assessment Framework for Embedded-device Risks SAFER [9], which qualifies a semi-automated threat valuation of IoT devices in several networks, is employed in the existing model. Besides, SAFER incorporates data from network device detection and programmed firmware analysis to evaluate the present threat related to the device. The strength is examined on the network that is implemented in the multi-national organization. Besides, the existing model systematically evaluates the security levels in IoT devices. Hence, the outcomes show that SAFER successfully recognized 531 from 572 devices with a detection rate of 92.83 %, examined 825 firmware imageries, and forecasted the existing and forthcoming security threat against 240 devices.

Correspondingly, the potentials are particularly advantageous in open-air settings, with inadequate connectivity and challenging regions to connect. Also, it has to manage the encrypted connections that protect the extended sensor networks with different environments. In the existing method, the algorithms that deliver the greatest compatibility within the existing structures are performed to be associated with the Wire guard, IKEV1-L2TP, and OpenVPN after the experimentations. Hence, the VPN has been installed with various security levels on similar hardware; the research demonstrates that OpenVPN, by TLS 1.3 through various data encryption procedures, then OpenWRT router produced better results when compared with the experimental results [10].

4.2 Firmware Binaries

Firmware binaries state the collected and executable code explicitly kept in non-volatile memory in a hardware device. Moreover, it remains a machine-readable form of the firmware, to be precise, the software that regulates the device functionality.

An open-source automatic static analysis tool extract [11] that extracts security-related structured data against exposed IoT peripheral firmware is deployed in the existing model. Object binaries, which implement the ARM Cortex-M structure, are used here due to its increasing regard between IoT peripherals. Besides, argXtract hinders the tasks related to exposing the Cortex-M study and can recover and progress the arguments towards security-related controller and function calls, empowering automatic bulk study of firmware collections. The outcomes of the classical method expose an extensive lack of protected data and varying data access controls with isolated vulnerabilities. The suggested argXtract method makes the automated security studies of exposed IoT binaries easier.

Similarly, an ML-based technique has been employed in the conventional model to categorize IoT firmware and greatest performance models. Besides, a wide-range assessment is carried out in the existing model based on specific procedures such as Logistic Regression, Random Forest classifiers, and Gradient Boosting. The tested approaches produce the outcomes for detecting every type of malicious cryptography. Moreover, a procedure such as The Matthews Correlation Coefficient (MCC) or Phi coefficient is utilized to generate more appropriate candidates for organization and resistance to precarious IoT structures [12].

4.3 Open Source Tools

In general, the features can be utilized to make implanting and train ML models towards robotically detect source code requires that comprise vulnerability resolutions. Although it shows isolation and is not different in a statistical manner, it also utilizes the tools to make an ML pipeline, which achieves outcomes equivalent to the advanced model. Besides, it is found that grouping the recent technique with commit2vec signifies a perceptible enhancement over the advancement in the automated detection of constraints that resolve vulnerabilities.

Moreover, the ML techniques that are constructed with commit2vec are consistent, and it is not accurate. The conventional ML models [13] centered on classifiers and ensemble methods are trained by combining the features. The commit2vec is interrelated with the other model and has a lesser prognostic efficiency to forecast a wider variety of commits. Hence, a typical model integrates both methods to attain better results than the results acquired by the methods individually.

Furthermore, a general overview of IoT has been provided, Viable IoT, with its design and the Internet Engineering Task Force (IETF) procedural set. Consequently, the open-source tools are explored, and datasets are aimed at the propagation in the existing studies with the development of IoT. Hence, a comprehensive classification of attacks related to numerous exposures is obtainable in the manuscript [14].

5. ML Methodologies

The ML methodologies can be engaged to improve firmware security in semiconductor devices. Some methods, such as supervised learning, unsupervised learning, and Open-source tools, are utilized in the firmware security in semiconductor devices.

5.1 Supervised Learning

Supervised learning (Supervised ML) remains a subsection of ML and AI. It is described by using labeled datasets to train processes, which are used to classify data or forecast results precisely. For instance, the input data is provided into the model and regulates the weights up to the structure that has been fixed properly. Additionally, it ensues as a portion of the cross-validation method. Supervised learning assists groups in resolving real-time difficulties at measure, namely, classifying spam in a distinct file after the inbox.

Introducing different accessible structures for the semiconductor engineering Final Test (FT) returns the extrapolation of leveraging ML methods. Besides, the classical method used classifiers such as SVM, kNN, and LR for training and validating the proposed model. Later, the above-mentioned structure can forecast FT yield at the wafer production phase. Therefore, FT that yields fewer difficulties can be gathered at the previous fabrication phase and is associated with the preceding research [15]. To improve model performance for both binary and multi-class classification, model selection and model ensemble using the F1-macro method are demonstrated. The existing structure has been deployed on the wide production of products through various wafer frameworks. It has been applied to three mass-production products with different wafer techniques with engineering flows. Hence, the existing method has attained a greater F1-macro test score for the conventional structure.

In data communication and implementation with IIoT, we have updated Predictive Maintenance (PdM) to preserve tools and feature control in engineering procedures. Besides, the PdM is valuable in scheme, capability, and Total Quality Management (TQM) fields. Moreover, PdM centered on cloud or edge computing has transformed smart engineering methods utilized in the existing method. Similarly, it advances ensemble-learning procedures by adaptive learning to create an advanced decision tree further intellectual. The procedure forecasts foremost PdM problems in the prevailing method, namely, failure of a product or definite industrial tools in progress. In addition, Adaptive Boost Decision Trees (ABDT) implemented in the model improve the efficacy of the computational performance. Semiconductor and eruption packing machine data are utilized distinctly in engineering data analytics [16].

5.2 Unsupervised Learning

Unsupervised learning is also identified as unsupervised ML, which utilizes procedures to examine and group unlabelled datasets. The procedures determine unknown designs or data groups deprived of human intervention. The capability to determine comparisons and variances in data variety is the perfect result for Exploratory Data Analysis (EDA), cross-selling approaches, consumer segmentation, and image recognition.

Furthermore, in unsupervised learning, clustering can be engaged to advance firmware security in semiconductor engineering. Hence, clustering procedures can support comprising unified firmware models centered on the structures, which can assist in categorizing designs, variations, and threats in potential security. By relating unsupervised learning methods to firmware models, semiconductor corporations may progress perceptions from the resemblances and variances among various firmware forms or variations. It may support detecting general vulnerabilities or otherwise malevolent code designs, which can occur through several firmware models.

Due to emerging IoT devices, firmware detection remains a critical challenge. To overcome the problem, the existing method analyses a firmware detection method to analyze webpage data based on feeble code words, and it identifies device type and variety. Through the usage of classification and page segmentation, the structure and firmware version of the device are detected. Hence, 74,307 devices are evaluated through SVM, NN, and RF algorithms to extract the firmware information. Besides, it improves the efficiency of the existing method. Investigational outcomes demonstrate that the existing technique attains an accuracy of 95.97%, greater than the former methods [17].

5.3 Reinforcement learning

Reinforcement Learning (RL) is the facts of decision-making, and it is roughly learning the optimum activities in a background to acquire determined compensation. The data remains gathered in RL after ML systems, which utilize a trial-and-error technique.

Moreover, RL can be employed for dynamic analysis along with modification of firmware threats. In the framework, RL is where an agent learns to sort decisions in addition to proceeds activities in the background to exploit a reward sign. When applying RL towards firmware threat mitigation, an agent can be trained to analyze dynamic firmware activities and make results to lessen potential threats. Besides, the agent interrelates with the firmware background, detects its portion, and takes activities to minimize the effect of threats.

Through the implementation of SARSA RL [18], the attack graph includes the set of probable attack settings achieved in contrast to the structure that prevailed in the study. The agent effectively established the finest path that may source the structure to maximum destruction. The existing results exposed that subsystem is mostly uncovered to cyber-attacks. Besides, the outcomes effectively exposed the worst-case attack situation by an entire reward of 26.9 and recognized the most severely impaired subsystems.

Similarly, implementing standard security measures is not constantly in effect, specifically with resource-controlled IoT devices. Consequently, here is a requisite to conduct perception analysis at the level of IoT systems. The existing method presented an automated penetration analysis structure [19] that employs RL based on the Q-Learning Network to assist the prevailing process. Besides, the structure can be utilized for defense training by reconstructing counterfeit attacks in the training background. Optimum outcomes demonstrate that

Q-Learning’s accuracy in hostile the optimum attack path by 86%. Further, the procedure can be measured as a pretend attack to detect some available vulnerabilities and a security breach in firmware.

6. Challenges and Future Direction

The extensive implementation of ML to resolve a huge set of real-world difficulties derived with requisite to gather and progress outsized sizes of data, and it is specified to consider individual and complex problems about data security. Hence, Privacy-enhancing technologies (PETs) are often designated. As a result, to defend individual data and attain overall dependability as of the requirement of recent EU procedures on data security along with AI. However, a standard implementation of PETs remains inadequate to confirm data security [20]. Some of the challenges are mentioned, and it is depicted.

6.1 Data Privacy and Ethics

Data privacy and ethics [21] are critical concerns when approaching firmware security with ML. Here are certain strategic facts to address the concerns:

- Data Privacy: ML models utilized for firmware security need access to complex data, like firmware encryption with system performance. It is needed to handle the data with maximum attention and confirm acquiescence by significant confidentiality guidelines.
- Informed Consent: After gathering firmware data for ML training, attaining informed consent against users remains vital. Hence, the users can be informed about the determination of data collection, conventional methodologies, and the involvement of some potential threats.
- Data Minimization: Gathering the needed data for ML training is ethical. Hence, minimalizing the number of Personally Identifiable Information (PII) with complex data collected decreases the data interruption threat with potential exploitation.
- Bias and Fairness: ML methods utilized in firmware security can be trained on varied and demonstrative datasets to avoid bias and confirm fairness.
- Constant Monitoring and Improvement: Observing ML methods for potential biases, mistakes, and unintentional concerns is critical.

6.2 Evolving Threat Landscape

As per advancements in technology, threats [22] are converted into further refinement, and it is critical to set ethical considerations and defend user data. Moreover, there is a trial in the gathering and storing individual data. Besides, the sets need to confirm that they have appropriate consent mechanisms in position and observe related data security principles. Moreover, applying robust security trials to protect data against illegal admittance is necessary.

An additional difficulty is the accountable usage of data as per ML procedures depending on a huge volume of data. Similarly, it is significant to confirm that data utilized for training methods is descriptive, impartial, and acquired by appropriate consent. Hence, the sets would be apparent about utilized data and take periods to lessen whichever potential biases otherwise biased results.

6.3 Explainability and Interpretability

Explainability and interpretability [23] are essential firmware security features after using ML systems. As ML methods become more difficult and influential, consider how they generate resolutions in security-critical fields like firmware. Besides, firmware security encompasses attentive vulnerability or threat detected and supports security designers to increase perceptions in the internal mechanisms of ML methods.

Similarly, interpretability emphasizes accepting the fundamental aspects and structures that have been added to the ML model's decision-making process. Besides, it comprises classifying the detailed firmware characteristics or designs that the techniques depend on vulnerability detection or threat classification. Moreover, it permits security professionals to increase their consideration of the ML representation's activities and supports detecting potential blind spots or regions aimed at upgrading.

7. Conclusion

Firmware security is important in protecting the reliability and secrecy of embedded structures. Significantly, adopting a general method for firmware security syndicates with automated security methods, consistent updates, and constant observation of potential vulnerabilities. Besides, the firmware is added with automated tools, static code, and dynamic analysis implements that can assist in detecting vulnerabilities and updating the security valuation method. The presented study could delve into the realm of ML and the implementation in evaluating firmware towards detecting vulnerabilities along with backdoors. Through training procedures on huge datasets of real firmware, models have created and captured the estimated performance of firmware. As a result of leveraging ML in firmware security, there is an enhancement in analyzing the firmware updates and reinforcements. Hence, the present study has analyzed the prevailing datasets and the methodologies used by the firmware in semiconductor devices are evaluated. To conclude, the presented paper presents an overview of the ML in firmware security. It can effectually examine firmware, expose vulnerabilities and backdoors, and subsidize it to generate a safer digital ecosystem. Besides, it is significant to note that Deep Learning (DL) can be combined with additional security methods to deliver complete security compared to firmware-based attacks.

References

[1]. W. Hu, C.-H. Chang, A. Sengupta, S. Bhunia, R. Kastner, and H. Li, "An overview of hardware security and trust: Threats, countermeasures, and design tools," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, pp. 1010-1038, 2020.

[2]. Z. Kenjar, T. Frassetto, D. Gens, M. Franz, and A.-R. Sadeghi, "{V0LTpwn}: Attacking x86 processor integrity from software," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 1445-1461.

[3]. S. Kaur, B. Singh, and H. Kaur, "Silicon Based Security for Protection Against Hardware Vulnerabilities," *Silicon*, pp. 1-7, 2021.

[4]. J. Harrison, N. Asadizanjani, and M. Tehranipoor, "On malicious implants in PCBs throughout the supply chain," *Integration*, vol. 79, pp. 12-22, 2021.

[5]. N. Chawla, A. Singh, H. Kumar, M. Kar, and S. Mukhopadhyay, "Securing iot devices using dynamic power management: Machine learning approach," *IEEE Internet of Things Journal*, vol. 8, pp. 16379-16394, 2020.

[6]. Z. Pan and P. Mishra, "A survey on hardware vulnerability analysis using machine learning," *IEEE Access*, vol. 10, pp. 49508-49527, 2022.

[7]. H. Liang, Z. Xie, Y. Chen, H. Ning, and J. Wang, "FIT: Inspect vulnerabilities in cross-architecture firmware by deep learning and bipartite matching," *Computers & Security*, vol. 99, p. 102032, 2020.

[8]. M. Kim, D. Kim, E. Kim, S. Kim, Y. Jang, and Y. Kim, "Firmae: Towards large-scale emulation of iot firmware for dynamic analysis," in *Annual computer security applications conference*, 2020, pp. 733-745.

[9]. P. Oser, R. W. van der Heijden, S. Lüders, and F. Kargl, "Risk prediction of IoT devices based on vulnerability analysis," *ACM Transactions on Privacy and Security*, vol. 25, pp. 1-36, 2022.

[10]. A. F. Gentile, D. Macrì, F. De Rango, M. Tropea, and E. Greco, "A VPN performances analysis of constrained hardware open source infrastructure deploy in IoT environment," *Future Internet*, vol. 14, p. 264, 2022.

[11]. P. Sivakumaran and J. Blasco, "argXtract: Deriving IoT Security Configurations via Automated Static Analysis of Stripped ARM Binaries," *arXiv preprint arXiv:2105.03135*, 2021.

[12]. E. Larsen, K. MacVittie, and J. Lilly, "A Survey of Machine Learning Algorithms for Detecting Malware in IoT Firmware," *arXiv preprint arXiv:2111.02388*, 2021.

[13]. T. Fehrer, R. C. Lozoya, A. Sabetta, D. Di Nucci, and D. A. Tamburri, "Detecting security fixes in open-source repositories using static code analyzers," *arXiv preprint arXiv:2105.03346,* 2021.

[14]. P. Anand, Y. Singh, A. Selwal, M. Alazab, S. Tanwar, and N. Kumar, "IoT vulnerability assessment for sustainable computing: threats, current solutions, and open challenges," *IEEE Access*, vol. 8, pp. 168825-168853, 2020.

[15]. D. Jiang, W. Lin, and N. Raghavan, "A novel framework for semiconductor manufacturing final test yield classification using machine learning techniques," *Ieee Access*, vol. 8, pp. 197885-197895, 2020.

[16]. Y. H. Hung, "Improved ensemble-learning algorithm for predictive maintenance in the manufacturing process," *Applied Sciences,* vol. 11, p. 6832, 2021.

[17]. D. Yu, L. Zhang, Y. Chen, Y. Ma, and J. Chen, "Large-scale IoT devices firmware identification based on weak password," *IEEE Access,* vol. 8, pp. 7981-7992, 2020.

[18]. M. Ibrahim and R. Elhafiz, "Security Analysis of Cyber-Physical Systems Using Reinforcement Learning," *Sensors,* vol. 23, p. 1634, 2023.

[19]. J. P. A. Yaacoub, H. N. Noura, O. Salman, and A. Chehab, "Ethical hacking for IoT: Security issues, challenges, solutions and recommendations," *Internet of Things and Cyber-Physical Systems,* vol. 3, pp. 280-308, 2023.

[20]. S. Z. El Mestari, G. Lenzini, and H. Demirci, "Preserving data privacy in machine learning systems," *Computers & Security,* vol. 137, p. 103605, 2024.

[21]. A. Karale, "The challenges of IoT addressing security, ethics, privacy, and laws," *Internet of Things,* vol. 15, p. 100420, 2021.

[22]. M. Anisetti, C. Ardagna, M. Cremonini, E. Damiani, J. Sessa, and L. Costa, "Security Threat Landscape," *White Paper Security Threats,* 2020.

[23]. S. Wolf, R. Foster, J. Haile, and M. Borowczak, "Data-driven suitability analysis to enable machine learning explainability and security," in *2021 Resilience Week (RWS)*, 2021, pp. 1-7.