Unsupervised IC Security with Machine Learning for Trojan Detection

Ashutosh Ghimire[1], Fathi Amsaad*[1], Tamzidul Hoque[2], Kenneth Hopkinson[3], and Md Tauhidur Rahman[4]

***Abstract*— The detection of Hardware Trojans is crucial for ensuring trust in the semiconductor IC supply chain. However, existing detection methods that rely on side-channel analysis often require golden chips for verification. This paper presents a new approach to Hardware Trojan detection that utilizes unsupervised machine learning with side-channel analysis to eliminate the need for golden data. Trojans of varying sizes were implemented on an FPGA to evaluate the method to perform unsupervised clustering and detect anomalies. The proposed model achieved high accuracy, 93%, and improved the detection of small and short-triggered Trojans compared to competing approaches. The unsupervised learning techniques demonstrated a better false positive rate and similar accuracy to supervised approaches such as the KNN classifier, SVM, and Gaussian classifier which require golden data for training. This research contributes a new approach to Hardware Trojan detection that can improve the trustworthiness of semiconductor IC supply chains.**

Keywords: IC Security, BIRCH, Clustering, Trojan, FPGA, side-channel, unsupervised

I. Introduction

Identifying and preventing hardware Trojans in integrated circuits (ICs) have advanced rapidly to ensure reliability and security in the design, manufacturing, and validation phases. However, when outsourcing IC production to untrusted facilities, there is a concern about potential malicious modifications that could lead to functional failures, data breaches, or other reliability issues [1]. Invasive methods like reverse engineering (RE) based hardware Trojan detection are dependable but not practical for a wide range of ICs due to the significant time and cost involved. Non-destructive techniques, such as side-channel analysis (SCA), are faster but susceptible to manufacturing process variations, making distinguishing between process-induced variations and hardware Trojans challenging [2]. Recent SCA research has focused on addressing variability-related concerns, and supervised machine learning has been implemented to improve the technique by identifying the boundary that separates Trojan-free and Trojan-infected chips. However, the requirement for multiple golden-ICs from various trusted ICs in SCA-based detection remains a significant drawback for the supervised learning-based machine learning method [3], [4].

Side-channel analysis (SCA) techniques for detecting hardware Trojans have traditionally required the use of golden ICs, which are expensive and difficult to obtain. Self-referencing methods have been proposed to overcome this limitation, using side-channel data from the same component at a different time or from different identical locations. However, these methods have their own limitations, such as requiring at least one Trojan-free location and being unable to detect combinational Trojans. Recently, machine learning techniques have been applied to SCA to propose golden-IC-free detection methods. For instance, a supervised learning-based method was proposed in [5] to detect anomalies in supply current resulting from Trojan attacks through simulation modeling. However, its effectiveness for classifying data from fabricated chips is unknown. Alternatively, the human temporal memory (HTM) based machine-learning model has been proposed that uses self-referencing and eliminates the need for the golden chip [6]. Unsupervised clustering methods have also been proposed that eliminate the need for a golden chip for SCA but perform reverse engineering (RE) of a few ICs if an anomaly is detected. In [3], ICs are divided into clusters based on their power signatures, and ICs with and without Trojans are likely to form different clusters. Similarly, [7] explored this approach using quantum diamond microscope magnetic field images. Both techniques suggest full RE of only a few ICs, which reduces the time, cost, and expertise required compared to traditional golden-chip-based approaches.

The need for effective and efficient hardware Trojan detection methods has become increasingly pressing in recent years. In this work, a novel method is introduced for hardware Trojan detection that combines the power of on-chip sensors and unsupervised machine learning. Our proposed method involves introducing a network of ring oscillators (RO) across the integrated circuit (IC) during the design stage. Once the IC is fabricated, RO data of all suspect ICs are analyzed through unsupervised clustering, allowing for the detection of any Trojans present in the system.

The research presented in this paper makes a valuable contribution to the field of hardware Trojan detection. By eliminating the need for destructive reverse engineering or a set of golden chips, the proposed method saves significant time and resources and makes the detection process more accessible to a broader range of stakeholders. In addition, the unsupervised nature of the approach removes the need for labeled data for training the model, which has been a significant challenge for many prior works. The evaluation of the framework on experimental RO data from 32 test chips demonstrates its effectiveness in detecting the presence of Trojans in ICs. The promising results achieved by the proposed method suggest that it has the potential to become a widely adopted approach for detecting hardware Trojans, significantly impacting the field of hardware security.

The rest of this document follows a particular organizational scheme. Section II gives an overview of prior works, the threat model, and essential background information. In

*Corresponding Authenticator

[1]Department of Computer Science and Engineering, Wright State University {ghimire.18, fathi.amsaad}@wright.edu

[2]Department of Electrical Engineering and Computer Science, University of Kansas hoque.18@ku.edu

[3]Department of Computer Science, Air Force Institute of Technology kenneth.hopkinson@afit.edu

[4]Department of Electrical and Computer Engineering, Florida International University mdtrahma@fiu.edu

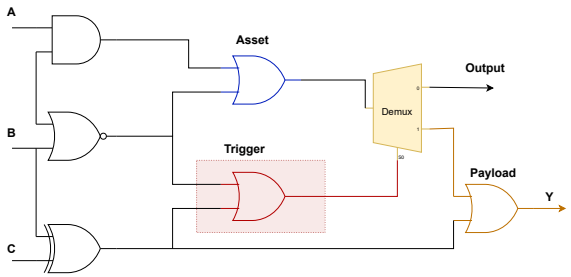

Fig. 1: Hardware trojan leaks sensitive data through the leakage

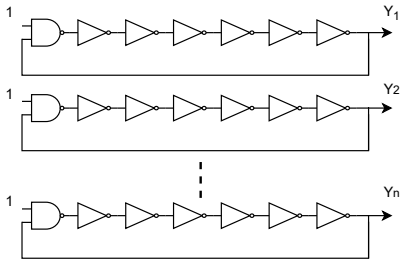

Fig. 2: Schematic of ring oscillator array

Section III, we present our Trojan detection methodology that involves an unsupervised clustering approach. Moving on to Section IV, we demonstrate our experiments and results, including a comparison with the previous approach that relied on supervised learning. Lastly, in Section V, we bring our work to a conclusion by discussing its contributions and potential future directions.

II. Background

A. Hardware Trojan

Hardware Trojans are a growing concern in the semiconductor industry due to their ability to modify circuitry, potentially leading to severe consequences maliciously. Hardware Trojans can be identified by their malicious intent and ability to go undetected by conventional functional verification and test procedures. A commonly recognized hardware Trojan model consists of the payload and the trigger, which are considered as main components. The trigger initiates the Trojan in response to an internal circuit state or input, while the payload affects the circuit's behavior after the Trojan is triggered [8], as shown in Fig. 1. One can classify hardware Trojans by their activation mechanisms, physical properties, and payload properties.

B. Threat Model

This study assumes that the foundry cannot be trusted, meaning that an adversary could potentially access the IC mask layout files and perform malicious modifications. We limit our focus to hardware Trojans that involve adding or removing logic, excluding doping-level Trojans, analog circuit-based Trojans, and other non-digital attacks [9]. By making these assumptions, we can focus our efforts on developing detection techniques specifically designed to identify digital hardware Trojans inserted during the manufacturing process.

C. Unsupervised Machine Learning

Machine learning algorithms, whether supervised or unsupervised, are important because they enable machines to learn from data and make predictions or decisions without being explicitly programmed [10] [11]. Unsupervised machine learning algorithms, like BIRCH, can efficiently cluster large datasets without prior knowledge or labeling of the data [12]. This is particularly useful in applications where data is complex, and its structure is not easily understood. The algorithm consists of four phases, starting with constructing the Clustering Feature (CF) tree, followed by clustering of non-leaf nodes recursively, clustering of leaf nodes using a standard clustering algorithm, and finally, assigning data points to clusters based on the nearest cluster distance metric. The CF Tree data structure reduces computational complexity and is suitable for clustering continuous and discrete attribute data.

D. Related Works

In the realm of hardware security, two fundamental techniques for discovering Trojans embedded in the manufacturing process are logic testing and SCA. Logic testing requires the complete activation of Trojans, which poses a challenge in creating test vectors capable of identifying every Trojan. Additionally, it fails to discover Trojans that cause side-channel leakage, as it only detects Trojans that influence the Integrated Circuit's operation [13]. On the other hand, SCA utilizes physical characteristics like transient current, leakage current, delay, energy, heat generation, or EM radiation to uncover the Trojan horse [14]. Unlike logic testing, SCA-based methods do not require the Trojan to be completely activated for detection. However, process and environmental variations can impact conventional SCA techniques' efficacy, making them less reliable [15].

Most SCA-based methods require a set of Trojan-free or golden Integrated Circuits (ICs) to recover the reference side channel signature. However, new SCA procedures that are free of golden-chip dependencies have been introduced by utilizing the suspicious devices' EM and light emission properties [16], [17]. The research uses laser probing to detect hardware Trojans by analyzing the reflected light of a circuit to detect any variations in behavior [16]. Similarly, the On-Chip EM Sensors technique involves monitoring the electromagnetic radiation emitted by the circuit and analyzing any deviations from the expected behavior [17]. Both techniques have been shown to detect previously undetected Trojans with high accuracy on several benchmarks. However, further research is needed to assess their practicality and scalability in real-world applications.

To replace the requirement for golden integrated circuits, researchers have also looked into machine learning approaches. For example, some studies have proposed supervised learning-based detection using supply current where the training data has been procured through simulation. However, these techniques are yet to be tested on Trojan-infected supply current from fabricated chips [5].

III. Methodology

A. Test Chip and Hardware Trojan Design

In order to evaluate RON structure and hardware Trojan designs, the 32 test chips were fabricated using IBM 90nm technology. The test chips featured the RON architecture

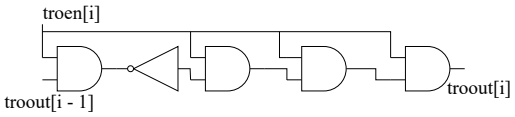

Fig. 3: A hardware Trojan design stage T_i

inserted into the ISCAS s9234 benchmark design, with 8-ring oscillator stages (ROs) and additional circuit components. Each chip contained seven hardware Trojan designs that could be deactivated, allowing for extensive testing and isolation of process variations from Trojan-inserted RO characteristic frequencies [18]. Figure 3 illustrates the gate-level representation of a Trojan stage.

B. Data Collection and Preparation

In this study, the data preparation process was crucial for achieving reliable unsupervised clustering results. Data for each chip included "golden" or Trojan-free samples and samples with Trojans, labeled only for evaluation purposes. Preprocessing steps removed irrelevant/redundant features, normalized feature values, and checked for missing values/outliers using imputation and interquartile range. These steps created a suitable dataset for feature extraction and clustering analysis, described in more detail in the following section.

C. Feature Extraction

In this study, PCA was used for the feature extraction process as a dimensionality reduction technique to extract informative features from the preprocessed data. The two components that explained the maximum variance in the data were identified through exploratory data analysis. The optimal number of PCA components was selected by choosing the two components that explained 80% of the total variance in the dataset.

The effectiveness of the PCA components was evaluated by visualizing the data in a reduced feature space using a 2D scatter plot. The scatter plot showed clear clusters of Trojan-free and Trojan-infected samples, indicating the effectiveness of the PCA components in extracting informative features for clustering analysis.

The use of PCA for feature extraction allowed for a reduction in the dimensionality of the feature space while preserving the most relevant information. Two PCA components were identified as optimal for the analysis, and they were effective in distinguishing between Trojan-free and Trojan-infected samples.

D. Clustering Model

The preprocessed and feature-extracted data, which consisted of both trojan-infected and trojan-free samples, were clustered using the BIRCH algorithm.

The BIRCH algorithm has two tuning parameters: the branching factor and the threshold, which were optimized to ensure the accurate detection of hardware trojans. The branching factor sets the maximum number of subclusters that can be combined into a larger cluster, while the threshold sets the maximum distance between a data point and its assigned subcluster centroid. These parameters were optimized using a validation set or cross-validation. The BIRCH

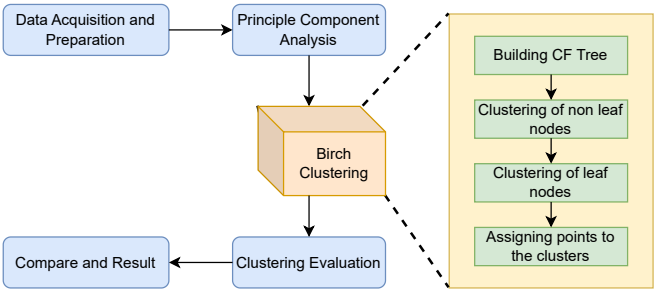

Fig. 4: Architecture of the BIRCH clustering model for hardware trojan detection

algorithm defines the distance between a data point and a subcluster centroid using the equation 1.

$$d_{i,j} = \sqrt{\sum_{k=1}^{p}(x_{i,k} - x_{j,k})^2} \tag{1}$$

where data points are given as j and i, p is the count of features, and $x_{i,k}$ and $x_{j,k}$ are the k^{th} feature values for data points i and j, respectively. This approach enabled us to detect the subtle differences between trojan and non-trojan cases, leading to accurate clustering and classification.

The clustering model was applied to the preprocessed and feature-extracted data, resulting in different clustering outcomes. The clustering model enabled us to distinguish between trojan and non-trojan cases, contributing to the development of a more reliable and effective approach for hardware trojan detection without the need for golden data.

E. Model Training and Evaluation

The clustering model was trained using the preprocessed data and evaluated using the labeled data. The evaluation metrics used for the models were the AUC value, F1-score, accuracy, G-mean, FNR, precision, FPR, and Recall. We repeated the model training and evaluation process for different sample sizes, including 6 samples, 12 samples, and 24 samples.

IV. Experiments and Results

A. Ring Oscillator Network Architecture

The ring oscillator network (RON) architecture was used in the studies in this paper to find Trojans in integrated circuits. Eight FPGA boards, notably the Nexys4 DDR development board, were used in the research. Each FPGA board was separated into four distinct areas, each of which was treated as a separate IC and Trojan in order to increase the sample size. While using different Trojan benchmarks from Trusthub, combinations and sequential Trojans were randomly distributed in one section of each integrated circuit (IC). To prevent one part or individual IC from interfering with another, the RON architecture was only deployed one piece at a time. Eight 41-stage ROs were found in each part of the IC. The average RO frequency was measured for 50 measurements with and without Trojans at room temperature and nominal operating voltage to eliminate measurement noise. Fig. 2 shows the schematic of the RO array. For the RON architecture-based hardware Trojan detection in integrated circuits, this experimental design ensured accurate and trustworthy results.

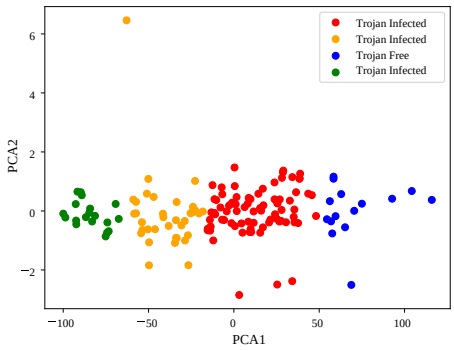

Fig. 5: Scatter plot depicting the distribution of clusters for a sample size of 6

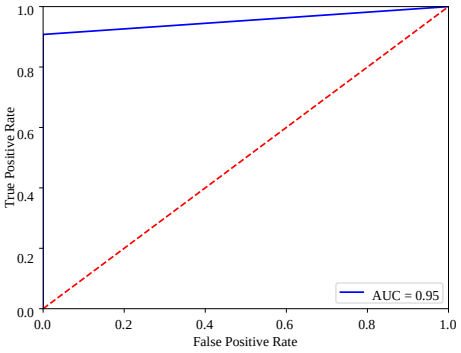

Fig. 6: ROC curve plot for 6 sample size with BIRCH clustering

The IC is mounted on a prototyping board and connected to a Xilinx Spartan-6 FPGA on a Nexys 4 board via a high-density serial connector. The FPGA controls the test sequence and transmits IC outputs to a computer using a USB-UART module. The setup includes voltage conversion and a voltage divider to supply the IC with 1.875V and the core with 0.9V. The FPGA's state machine manages the data collection process, including selecting counter outputs and transmitting them as hex digits. Each IC undergoes 10 trials with a 500-clock cycle duration. The ICs contain pre-inserted hardware Trojan designs, but during data collection, the Trojans are disabled for some samples [18].

B. Cluster Size selection

Our machine learning experiment employs the "adjust branching factor and threshold values" strategy to determine the ideal number of clusters that control the granularity of the clustering and can be used to obtain the optimized number of clusters. The branching factor determines the maximum number of subclusters in each node, while the threshold determines the radius of the subcluster. By setting these parameters appropriately, the size and number of the resulting clusters are controlled. We experimented with different parameter values and evaluated the resulting clusters using silhouette score, which measures the quality of the clustering, to select the best clustering solution.

C. Cross-Validation Approach

In machine learning and data analysis, cross-validation is a frequently used technique to increase the accuracy and generalizability of the model. This method involves dividing the dataset into subsets, with one subset reserved for testing and the others used for training. In BIRCH clustering, the dataset is divided into several folds for cross-validation. The validation set for each fold is used only once, with the remaining folds being used for training. The method is executed numerous times, with the validation set changing for each run. Performance metrics such as mean squared error are used to evaluate the algorithm, and optimal tuning parameter values are chosen based on the best performance.

D. Results

BIRCH clustering not only demonstrated high accuracy and runtime efficiency in clustering the dataset, but it also produced informative visualizations for a better understanding of the clustering results. The scatter plot presented in Fig. 5 illustrates how the data points are distributed among the clusters formed using the BIRCH clustering algorithm for a sample size of 6. It is apparent from the plot that the clusters are distinctly separated.

TABLE I: Table showing BIRCH Clustering Results

Metric	**Sample Size**		
	6	**12**	**24**
FNR	0.086	0.082	0.07
TNR	0.9	0.95	0.95
TPR	0.913	0.917	0.928
FPR	0.09	0.045	0.04
Accuracy	0.913	0.92	0.93
F1 Score	0.951	0.955	0.96
AUC	0.95	0.95	0.96

In our experiments, the algorithm was able to cluster the dataset accurately with a clustering accuracy of 92.5% with the default configuration. Increasing the branching factor parameter to a value of 100, we observed a marginal increase in accuracy to 93.1%. However, this came at the cost of an increased runtime. Further increasing the branching factor resulted in a significant increase in runtime with no improvement in accuracy. Additionally, we observed that the BIRCH clustering algorithm was able to scale well with large datasets. Even with a dataset size of a small number of samples, the algorithm was able to cluster the data within a reasonable amount of time. Overall, BIRCH clustering showed great potential as a fast and accurate clustering algorithm for large datasets. Additionally, as shown in Fig. 6 curve, a receiver operating characteristic (ROC) curve was produced to assess the clustering performance and the trade-off between the rate of true positives and the rate of false positives. Strong clustering performance was demonstrated by ROC curve, which had a high area under the curve value called AUC of 0.95.

In this study, we compared the performance of clustering technique to several ensemble models of different supervised methods including Naive Bayes (NB), Support Vector Machine (SVM), and K-nearest neighbors (KNN) classifiers. The goal was to evaluate whether clustering technique can perform well without relying on training labels.

The results of our experiments are summarized in the Table II. It shows the accuracy and G-mean for each model across three sample sizes: 6, 12, and 24.

TABLE II: Comparison of clustering techniques and prior supervised models performance

Model	Accuracy			G-mean		
	Sample Size			Sample Size		
	6	**12**	**24**	**6**	**12**	**24**
BIRCH Cluster	0.913	0.92	0.93	0.906	0.933	0.94
Ensemble - SVM + KNN + NB [19]	0.922	0.92	0.94	0.85	0.86	0.926
Ensemble – SVM + NB [19]	0.88	0.879	0.88	0.90	0.91	0.933
Ensemble – KNN + NB [19]	0.886	0.883	0.873	0.92	0.93	0.92

As shown in Table II, the clustering technique generally outperformed the ensemble models in terms of accuracy and G-mean across all sample sizes. BIRCH Clustering achieved an accuracy of 0.913, 0.92, and 0.93 for the three sample sizes, respectively. In contrast, the ensemble model of SVM, NB, and KNN classifiers achieved an accuracy of 0.922, 0.92, and 0.94 for the largest sample size of 24. The G-mean for this ensemble model was 0.926 for the largest sample size.

Overall, the results suggest that clustering techniques can be effective in solving classification problems, particularly in cases where training labels may be limited or difficult to obtain.

V. Conclusion

The research work presented in this paper demonstrates the effectiveness of unsupervised machine learning with side-channel analysis for Hardware Trojan detection. Our proposed approach has the potential to enhance the trustworthiness of semiconductor ICs and improve the confidence of end-users in the authenticity and security of the devices they use. However, there are potential future directions for this research. For example, further studies can be conducted to investigate the robustness of the proposed approach against attacks, such as adversarial examples. Additionally, the proposed approach can be extended to multi-chip systems to improve the security of the entire system.

In conclusion, our research presents a new approach to Hardware Trojan detection that utilizes unsupervised machine learning with side-channel analysis. The proposed method achieved high accuracy and outperformed competing approaches in detecting small and short-triggered Trojans. The results suggest that clustering techniques can effectively solve classification problems, and this finding has implications for other applications where training data may be scarce or expensive to collect. Overall, our proposed approach provides a valuable contribution to the semiconductor IC supply chain security field and has significant potential for future research and development.

ACKNOWLEDGMENT

The authors gratefully acknowledge the technical and financial support provided by the Air Force Research Lab (AFRL) through the Assured and Trusted Digital Microelectronics Ecosystem (ADMETE) grant, BAA-FA8650-18-S-1201, which was awarded to Wright State University, Dayton, Ohio, USA. This project was carried out under CAGE Number: 4B991 and DUNS number: 047814256.

References

[1] A. Jain, Z. Zhou, and U. Guin, "Survey of recent developments for hardware trojan detection," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2021, pp. 1–5.

[2] M. A. Rahman, M. T. Rahman, M. Kisacikoglu, and K. Akkaya, "Intrusion detection systems-enabled power electronics for unmanned aerial vehicles," in *2020 IEEE CyberPELS (CyberPELS)*. IEEE, 2020, pp. 1–5.

[3] S. Yang, P. Chakraborty, P. SLPSK, and S. Bhunia, "Trusted electronic systems with untrusted cots," in *2021 22nd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2021, pp. 198–203.

[4] N. Karimian, F. Tehranipoor, M. T. Rahman, S. Kelly, and D. Forte, "Genetic algorithm for hardware trojan detection with ring oscillator network (ron)," in *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*. IEEE, 2015, pp. 1–6.

[5] M. Xue, J. Wang, and A. Hux, "An enhanced classification-based golden chips-free hardware trojan detection technique," in *Hardware-Oriented Security and Trust, IEEE Asian*. IEEE, 2016, pp. 1–6.

[6] S. Faezi, R. Yasaei, A. Barua, and M. A. A. Faruque, "Brain-inspired golden chip free hardware trojan detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2697–2708, 2021.

[7] M. Ashok, M. J. Turner, R. L. Walsworth, E. V. Levine, and A. P. Chandrakasan, "Hardware trojan detection using unsupervised deep learning on quantum diamond microscope magnetic field images," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 2022.

[8] J. R. Hamlet, J. R. Mayo, and V. G. Kammler, "Targeted modification of hardware trojans," *Journal of Hardware and Systems Security*, vol. 3, no. 2, pp. 189–197, 2019.

[9] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant-level hardware trojans: extended version," *Journal of Cryptographic Engineering*, vol. 4, no. 1, pp. 19–31, 2014.

[10] A. Ghimire, H. Tayara, Z. Xuan, and K. T. Chong, "Csatdta: Prediction of drug–target binding affinity using convolution model with self-attention," *International journal of molecular sciences*, vol. 23, no. 15, p. 8453, 2022.

[11] A. Ghimire, A. Chapagain, U. Bhattarai, and A. Jaiswal, "Nepali handwriting recognition using convolution neural network," *International Research Journal of Innovations in Engineering and Technology*, vol. 4, no. 5, p. 5, 2020.

[12] B. Lorbeer, A. Kosareva, B. Deva, D. Softić, P. Ruppel, and A. Küpper, "Variations on the clustering algorithm birch," *Big data research*, vol. 11, pp. 44–53, 2018.

[13] Z. Pan and P. Mishra, "A survey on hardware vulnerability analysis using machine learning," *IEEE Access*, vol. 10, pp. 49 508–49 527, 2022.

[14] S. Kelly, X. Zhang, M. Tehranipoor, and A. Ferraiuolo, "Detecting hardware trojans using on-chip sensors in an asic design," *Journal of electronic testing*, vol. 31, no. 1, pp. 11–26, 2015.

[15] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware trojan attacks: threat analysis and countermeasures," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1229–1247, 2014.

[16] G. V. Cataloni, P. S. Sawyer, and S. R. Hasan, "A primer on hardware trojans including platform specific attacks and machine learning for detection," in *SoutheastCon 2022*. IEEE, 2022, pp. 479–486.

[17] J. He, X. Guo, H. Ma, Y. Liu, Y. Zhao, and Y. Jin, "Runtime trust evaluation and hardware trojan detection using on-chip em sensors," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2020, pp. 1–6.

[18] A. Ferraiuolo, X. Zhang, and M. Tehranipoor, "Experimental analysis of a ring oscillator network for hardware trojan detection in a 90nm asic," in *Proceedings of the International Conference on Computer-Aided Design*, 2012, pp. 37–42.

[19] K. Worley and M. T. Rahman, "Supervised machine learning techniques for trojan detection with ring oscillator network," in *2019 SoutheastCon*. IEEE, 2019, pp. 1–7.