AI-Enabled Image Processing Approach for Efficient Clustering and Identification of Hardware Trojans*

Ashutosh Ghimire[a], Mohammed Alkurdi[a], Saraju Mohanty[b], Fathi Amsaad[a,*]

[a]*Department of Computer Science and Engineering, Wright State University, 3640 Colonel Glenn Hwy., Dayton, 45435, OH, USA*
[b]*School of Computer Science and Engineering, University of North Texas, 2204 W Prairie St, Denton, 76203, TX, USA*

Abstract

Hardware Trojans are emerging malicious integrated circuit (IC) modifications that pose a significant threat to the integrity of electronics. While existing methods, such as functional testing and reverse engineering, are proposed to identify Trojan anomalies in electronics, their applicability to industrial pipelines is limited. This paper proposes a new image processing technique for efficient clustering and identification of Hardware Trojan insertion in integrated circuits. The uniqueness of the proposed AI-assisted image processing method relies on using real hardware to generate images using side-channel analysis (SCA) before applying unsupervised image classification to identify the impact of hardware Trojans without the need for costly golden references. Leveraging Machine Learning on side-channel data collected from Ring-Oscillator networks, image and digital signal processing are employed to extract features for detection. This research contributes a novel use of side-channel data as images, eliminating the reliance on golden

*This research was funded by a grant 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.

*Corresponding Author

Email addresses: ashutosh.ghimire@wright.edu (Ashutosh Ghimire), alkurdi.2@wright.edu (Mohammed Alkurdi), saraju.mohanty@unt.edu (Saraju Mohanty), fathi.amsaad@wright.edu (Fathi Amsaad)

references, and achieving a remarkable accuracy of 95% in Hardware Trojan detection. In addition to significantly advancing the field and addressing crucial challenges in semiconductor supply chain, making it a significant step toward securing it.

Keywords: Image Processing and Classification, Machine Learning, Very Large Scale Integration, Hardware Trojan Identification, Integrated Circuits, FPGA for Side-channel Analysis.

---

1. Introduction

Industrial manufacturing of integrated circuit (IC) chips shows a worldwide growing dependence on overseas foundries for fabricating IC designs originating in System-on-Chip (SoC) design houses. This shift is a result of the escalating demand for chip quantity and the diminishing feasibility of a single entity undertaking the entire manufacturing process–from design to fabrication to distribution. This challenge emerges as a result of the increasing complexity of the designs and the requirements of specialized equipment that make the cost of having a dedicated foundry in-house impractical and inefficient [1]. This challenge emerges as a result of the increasing complexity of the designs and the need of specialized equipment that makes the cost of having a dedicated foundry in-house impractical and inefficient [1].

This rising dependence on these foundries necessitates the continuous operation of their pipelines to meet customer demand, adhere to production deadlines, and sustain overall production efficiency. An integral aspect of this fabrication process is the assurance of the quality and trustworthiness of the product. This is achieved through multiple means including functional and imaging tests. These tests are significant due to the emergence of hardware Trojans, which are malicious modifications to the IC chip to achieve a purpose not intended by the original designer, which have the potential to cause breakage of operations, critical data leakage, or undermine reliability.

Functional logic testing [2, 3], destructive reverse engineering of the fabricated chip [4], and image processing [5], are examples of proposed methods to detect these modifications. While effective for individual chips, they become impractical for industrial applications, as there is a need for the detection of hardware Trojan presence across a diverse array of ICs in bulk.

Side-channel analysis (SCA) is another method proposed to detect the presence of hardware Trojans that yields swift results. However, its reliance

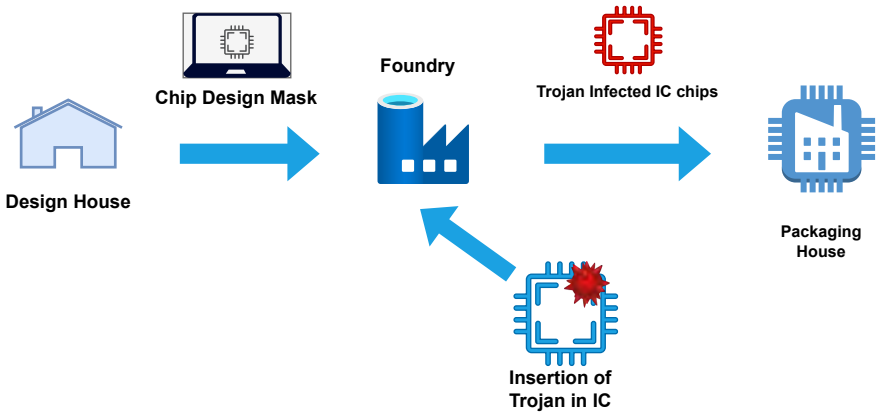


Figure 1: Overview of Malicious Hardware Trojan Insertion in the IC Supply Chain Applications

on physical chip data, causes inaccuracies resulting from noise and production variances [6]. Methods to overcome these inaccuracies are proposed by many researchers in [7, 8, 9]. Additionally, supervised machine learning is utilized with side-channel analysis to facilitate higher accuracy in the detection of hardware Trojans. However, this approach relies on data from reference chips called Golden Reference to construct the labeled dataset for model training, which is infeasible in industrial applications due to cost and accessibility challenges[10].

This paper proposes employing Machine Learning techniques to identify hardware Trojans by analyzing side-channel data obtained from Ring Oscillator Networks, which are design-for-trust components that enable the collection of side-channel data from the IC chip while overcoming noise and production variances. The proposed approach involves utilizing Image and Digital Signal Processing to extract features, which are subsequently utilized in our machine learning algorithms to detect the presence of hardware Trojans.

1.1. Key Contributions

This research advances the domain of AI-enabled image processing by introducing a new technique to classify and identify malicious modifications in IC supply chain applications.

This research's main contributions are:

1. An AI-based image processing technique to enhance classification and

identification of hardware Trojan modifications in semiconductor supply chain is proposed, realized and tested on real hardware (Field Programmable Gate Arrays or FPGAs)

2. Unlike earlier techniques, the proposed method utilizes advanced image and signal processing for feature extraction and employs unsupervised machine learning to overcome the need for golden reference signatures to identify malicious hardware modifications in ICs. This is crucial to solving the issue of the availability of golden chips and building accommodations for various IC chips.
3. The proposed methods reduce the reliance on destructive reverse engineering and cost of expensive tools needed by reverse engineering using non-destructive and efficient image processing techniques.
4. The results show the proposed AI-enabled image processing approach enhances the identification and classification of hardware Trojan presence in semiconductors compared to the state of the art. The proposed model's accuracy is found to be 95%, which outperforms many existing techniques and sets up AI image processing as a strong candidate for efficient identification and classification of anomaly detection in semiconductors.

1.2. Organisation of the paper

The following sections of this paper conform to the standard organizational structure, where the threat model is constructed, and the primary background is clarified in Section II, existing previous work in the field of hardware Trojan detection is surveyed in Section III, the approach followed in this research is presented in Section IV, the research implementation is then discussed in detail from data capturing to machine learning model training in Section V, results are showcased and discussed in Section VI, and this paper is concluded in section VII.

2. Background

2.1. Threat model

As shown in Fig. 1, this research paper assumes that IC chips are fabricated by untrusted third parties, increasing the possibility for the insertion and integration of hidden smart and remote hardware Trojans. This concern arises due to the suspicion that malicious actors might gain access to the

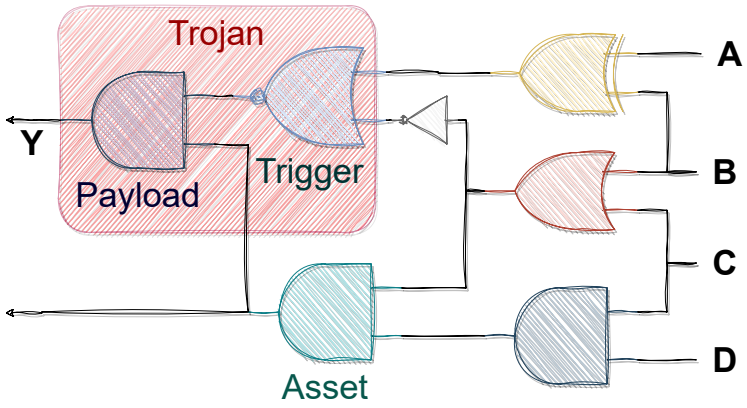


Figure 2: An example of an IC hardware Trojan which is the alteration of the chip during the manufacturing process of a Consumer Electronics Device in a Third Party Fabrication Foundry.

mask layout files of ICs, thereby enabling them to introduce harmful modifications. The focus of this research is specifically on post-manufacturing hardware Trojans, particularly those that involve the inclusion or omission of logical components. It intentionally excludes more complex alterations such as circuit-based analog exploits, doping-level Trojans, and other non-traditional types of compromises. This limitation in the threat model is by design, aimed at developing a method that can accurately detect digital hardware Trojans that are clandestinely incorporated during the fabrication phase.

2.2. Hardware Trojan Design

Recently, hardware Trojans have emerged, raising critical concerns regarding the reliability and trustworthiness of devices in the semiconductor supply chain. These Trojan circuits can be inserted into hardware, leading to altered functionality, deactivation, or destruction of the devices. These malicious hardware components are craftily designed and integrated, making them difficult to detect using conventional testing and verification techniques. Typically, hardware Trojans are modeled on the concept of a Trigger and a Payload [11]. As shown in Fig. 2, the Trigger acts as a concealed catalyst, monitoring the circuit's state and waiting for a specific condition during its operation or for an external signal, whether electrical or radio-based, to activate the Payload. The Payload, as the primary malicious agent, executes

harmful actions such as leaking sensitive data processed by the circuit or causing the circuit to malfunction

2.3. Machine Learning

Machine learning is a subfield of artificial intelligence that allows systems to learn from data and assist in making informed decisions. It does not require the programming of these systems explicitly which is a mark of the rise of machine learning applications in various critical roles [12, 13]. Machine learning addresses the issue of improving a system's performance through the experiences it undergoes. An example is improving the accuracy of a model in predicting a stock market by performing predictions of the stock market and failing, afterwards, utilizing these failures to re-calibrate the model to achieve higher prediction accuracy next time. This is achieved by applying computer science and statistic concepts, through the identification of a problem that can be learned, and the application of statistical methods to infer an approximate answer and to measure how reliable our answer is [14].

Machine learning is divided into multiple areas depending on the nature of the data utilized in the training of a machine learning model, example areas are [15]:

2.3.1. Supervised

This area is used with a dataset of labeled data—a dataset of input-output pairs, that requires a machine learning model to learn an algorithm that infers the underlying patterns in these data which links each output with its provided input. This algorithm represented as a machine learning model can be used to predict outputs from future unlabeled data with a high degree of confidence. This area is used when the output of the model is expected and known-—knowledge of the existence of a pattern in the dataset between inputs and outputs. Algorithms used to infer these patterns are numerous, such as linear regression, random forest, K-Nearest Neighbors (KNN), and others...

2.3.2. Unsupervised

This area is used for the discovery of hidden patterns in datasets that appear to not have a relation between its members. This is achieved through discovering similarities between members of the dataset and grouping similar members together, which the algorithm can then highlight hidden patterns

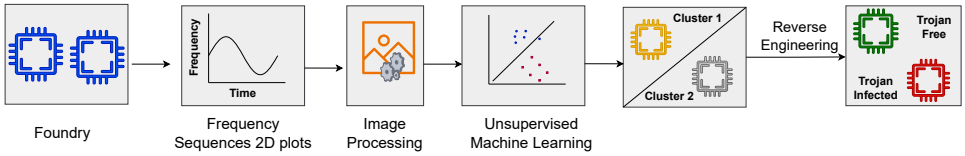


Figure 3: The Proposed AI-Enabled Image-Based Clustering Approach for the Clustering and Identification of Hardware Trojan

that reveal relations previously unpredicted. This area is used when the output of inputs is unknown, and knowledge of existing patterns is uncertain. Algorithms used in these areas are many, such as K-Means, BIRCH, AGNES, and many more.

3. Related Works

The potential for integrated hardware Trojans in IC chips is raising a concern for their potential harm that could reach the users of electronics. This motivates many researchers to delve into researching methods to detect malicious insertions. Many methods are pursued in detecting hardware Trojans in post-fabricated IC chips, three of which are reverse engineering, logic testing, and SCA [1].

The use of invasive or semi-invasive methods like optical imaging for reverse engineering, involves the subsequent scrapping of the IC's layers and obtaining an image for each layer to finally construct the structure of the chip under test and compare it with the original design to obtain a conclusive evidence of the presence or absence of a hardware Trojan. Although this method is conclusive, it is quite expensive and time-consuming for a method that is only for a single chip that also renders the said chip as a non-functional scrap metal and does not help in inferring the status of the other chips in the same production line [16].

Other researchers in [5, 17] propose solutions that utilize imaging techniques to circumvent some of these limitations but they still cover only a small subset of the wide array of ICs, in addition to their still constant reliance on golden reference—a trusted fabricated chip of the design.

Another advantageous method is the use of non-invasive techniques which involve validating the functionality of the chip through Logic Testing. The advantage of this method is facilitating the continued operation of the chip due to the non-destructive application of the detection of hardware Trojans

method. This method involves creating a list of validation operations, inputs, and expected outputs based on chip designs, aiming to cover most execution paths and trigger the Trojan if present [18, 19].

However, the material and time costs of this method are gradually increasing with the increasing IC complexity. This results in this method being progressively challenging for industrial use, in addition to the limited scope of hardware Trojans that this method can detect. This is a result of detecting only hardware Trojans that change the functionality of the IC, while there exists Trojans which leak circuit data to outsiders without the need to change the functionality of the circuit.

An additional non-invasive method is analyzing the side-channel data of the circuit [20]. SCA involves the collection of the myriad characteristics and properties of the chip during its operation, them being generated or emitted. These characteristics and properties encompass a wide variety, such as path delays [21], transient power [22, 23], radiation imaging [24], and others in [16]. The collected data is then processed and transformed by mathematical, image, and signal processing techniques to draw out evidence of a hardware Trojan's existence or nonexistence in the IC. This method can be applied to a wide range of IC chips and offers quick results, however, it struggles with inaccuracies in the presence of production variations and the reliance on golden references.

This resulted in recent researchers proposing the inclusion of a variety of tactics that help circumvent these product variations to acquire more accurate results in detecting hardware Trojans. One such tactic is the inclusion of components that facilitate the collection of data as a design-for-trust methodology. These components facilitate obtaining more accurate measurements which ignore product variations. Ring Oscillator Networks (RONs) [25] is such a component that is integrated in the IC chip during the design stage, which provides the facilities to monitor the power of the circuit by oscillating on a frequency that captures the power currents and voltage of the IC in the area it covers. Utilization of these frequencies, signatures of golden references are constructed, which are then employed to detect variations in the other IC chips' signatures, that result in the eventual detection of the presence or absence of a hardware Trojan.

An additional proposed tactic employed with design-for-trust components is the exploitation of machine learning's ability to glean hidden patterns existing in data. This, combined with side-channel data extracted as features, could lead to higher accuracy in the detection of hardware Trojans while

Table 1: Comparison of State-of-the-Art Hardware Trojan Detection Methods

Method Category	**Golden Reference Free**	**Non-Destructive /Minimal RE**	**Detection Sensitivity**	**Automation and Adaptability**	**Noise-Resilient Visual Encoding**
Reverse Engineering / Optical Imaging [4, 24]	✓	×	High	×	×
Logic Testing / ATPG [2, 18]	✓	✓	Low	×	×
SCA Fingerprinting (Power/Timing) [22, 16]	×	✓	Moderate	×	×
Design-for-Trust (RON) + Supervised ML [10]	×	✓	High	✓	×
Image-based Unsupervised (Proposed Work)	✓	✓	**High**	✓	✓

at the same time circumventing product variation. K. Worley Et al. [10] proposes a method that does just that, he utilizes signatures obtained from different IC chips, some have hardware Trojans inserted in them, as labeled data, which he utilizes in training multiple supervised machine learning models that achieves an impressive 94% accuracy in detecting the presence of hardware Trojans. Although this approach circumvents the issue of production variations and has very high accuracy, it still relies on golden references to construct its set of labeled data.

Building upon these prior efforts, a comparative assessment of state-of-the-art hardware Trojan detection approaches is presented in Table 1. The analysis underscores that while reverse engineering and logic testing provide conclusive or non-destructive evaluations, they either incur prohibitive costs or fail to address subtle Trojans that do not alter functionality. Similarly, side-channel fingerprinting and design-for-trust methodologies, even when enhanced with supervised learning, remain constrained by their reliance on golden references and susceptibility to noise. In contrast, the proposed image-based unsupervised framework introduces a noise-resilient visual encoding of side-channel data that captures multi-dimensional relationships across frequency and time. By transforming raw, high-dimensional traces into compact and interpretable images, the method constructs a machine-

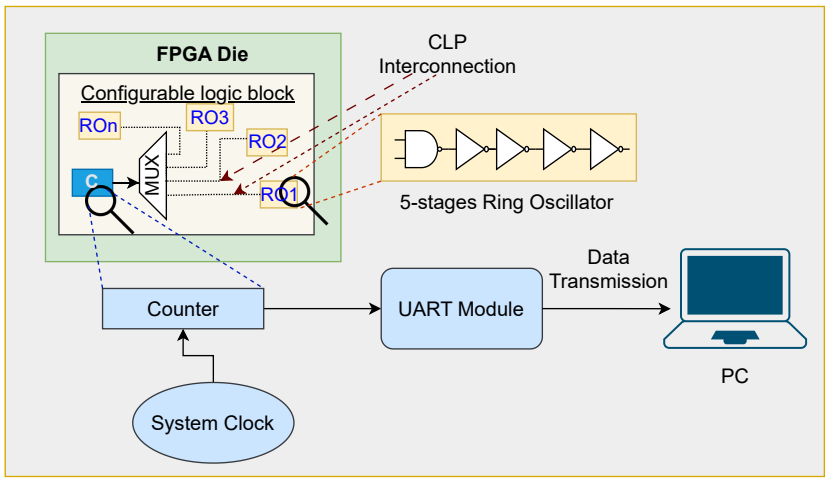


Figure 4: Schematic Diagram of the FPGA Hardware Implementation for Collecting Side Channel Information in AI-Enabled Image Processing for Hardware Trojan Clustering and Identification

learning–friendly feature space where Trojan and clean chips become more separable, thus enabling robust clustering without costly golden references. This automated, reference-free capability positions the proposed approach as a distinctive advancement beyond the limitations of existing methods.

4. Overview of Proposed Methodology

This research aims to employ an image-based unsupervised machine learning approach for the accurate identification of Trojan-infected chips as shown in the Fig. 3. The frequency side channel readings are collected from the chips under test, some are configured as Trojan-infected and others as Trojan-free. The collected data is organized into a sequential format, where each sequence corresponds to the frequency variations recorded from a specific Ring Oscillator (RO) over multiple input instances. This transformation provides a structured representation of the data, capturing indirect power consumption patterns that may indicate the presence of hardware Trojans. By capturing these frequency shifts, which may indicate modifications within the circuit, we can apply analysis to effectively differentiate between Trojan-free and Trojan-infected chips based on their unique frequency response sequences.

The sequences of data are visualized in 2D graphs, allowing for the extraction of features from a visual basis to enhance interpretability. Afterwards,

the pixel data of these plots are categorized using clustering algorithms. The model is optimized to determine the optimal number of clusters for precise categorization. Once clusters are established, a sample from each cluster is reverse engineered to identify whether it is Trojan-free or Trojan-inserted. Subsequently, the entire cluster is labeled accordingly.

In addition to the proposed imaging technology, alternative methods are proposed for extracting spectral-based features, such as frequency after Fourier transformation, amplitude, and wavelet features of the data sequences. These extracted features are then tested with various machine learning models, including both supervised and unsupervised approaches, to compare the performance of the proposed method with conventional state-of-the-art methods.

To ensure the accuracy of the detection process and mitigate the risk of misidentifying Trojan-free ICs as infected, a range of evaluation metrics, including the Area Under the Curve (AUC) value, precision, accuracy, F1-score, and recall, are employed. Further details about the methodology, including data generation, preprocessing, feature extraction, and machine learning analysis, are elaborated in experimental setup and implementation section.

5. Experimental Setup and Dataset

5.1. Ring Oscillator (RO) Setup

An experiment was conducted to gather data on the frequency fluctuations of individual Ring Oscillators (ROs) across diverse operational settings, with the aim of evaluating their susceptibility to hardware Trojans. The setup for RONs involves the construction of ROs using NOT and NAND gate-based configurations with 5 stages. The number of stages is taken as odd number for the oscillation. Additionally, an 8-bit counter was designed to capture side-channel traces of the ROs, synchronized with the system clock. The Enable signal for the ROs serves a dual purpose: enabling the frequency generation of ROs and counting the number of cycles in the counter. Following the frequency cycle count, the traces were transmitted to connected devices via the UART module, as illustrated in Fig. 4. The RO blocks are strategically positioned at 32 spots across the FPGA die to cover nearly all power rails of the chip. This strategic placement ensures that any hardware Trojan inserted into the chip causing a voltage drop would be detected

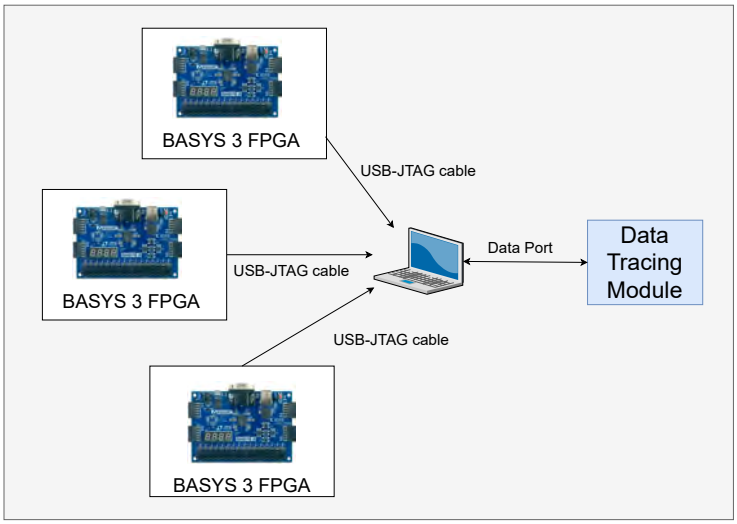


Figure 5: FPGA Setup for Data Collection and Tracing

[25]. The specific additional voltage drop (V_{mali_drop}) affects the oscillation frequency of an N-stage RO, as expressed by the equation 1 [26].

$$f = \frac{\mu_g \times (V_{DD} - V_{threshold} - V_{mali\_drop})^{\alpha}}{2N \times k} \tag{1}$$

In this equation 1, α represents the velocity saturation index, V_{DD} denotes the supply voltage, $V_{threshold}$ is the threshold voltage, μ_g refers to the carrier mobility, and k is a constant that depends on the gate.

5.2. Field-Programmable Gate Arrays (FPGAs) Setup

FPGAs provide a reconfigurable and cost-effective platform for evaluating hardware security mechanisms, including hardware Trojan detection. Their ability to dynamically modify circuit configurations enables rapid prototyping and iterative testing of attack scenarios without requiring expensive fabrication [27]. Compared to Application-Specific Integrated Circuits (ASICs), FPGAs allow precise control over circuit placement and routing, ensuring consistency in experimental conditions for side-channel analysis [28]. Additionally, their real-time power and timing measurement capabilities make them well-suited for evaluating side-channel leakage induced by malicious modifications [29].

The proposed method has been tested on FPGAs due to their suitability for hardware Trojan analysis and detection. The ability to integrate various Trojan designs, collect side-channel traces, and analyze power fluctuations in a controlled environment makes them an ideal choice for this research. Moreover, FPGA-based evaluations provide practical insights that can be extended to security-critical hardware applications [30].

The experiment is conducted using three Basys 3 FPGAs to implement the MicroBlaze microprocessor along with ROs and hardware Trojans. The Basys 3 FPGA development boards feature Xilinx Artix-7 $XC7A35T-1CPG236C$. FPGA chips equipped with USB-JTAG protocols for communication with external devices, such as a laptop in our experiment. ROs are integrated into the FPGA die using Xilinx XDC macro features, ensuring fixed positioning and identical routing.

Fig. 5 provides an overview of the experimental setup, illustrating how the three FPGAs were utilized for feature extraction from side-channel signals. The side-channel traces, collected from ROs, serve as the input data for the AI-enabled image processing approach. These traces capture frequency variations caused by the presence of hardware Trojans, enabling the identification of anomalies. The setup ensures that all FPGA power regions are monitored for potential signal deviations, which are later analyzed using clustering techniques.

In the experimental setup, ROs were integrated alongside an md5 cryptographic core and a MicroBlaze processor. To collect Trojan-free RO data, the setup was maintained with Trojan circuits in a deactivated state. Ensuring a consistent experimental setup, including identical placement and routing, the Trojans were then activated, and data affected by the Trojan was subsequently collected.

The selection of Basys-3 FPGAs is deliberate and motivated by both practicality and methodological rigor. Their cost-effectiveness and widespread availability make them an accessible platform for replicating side-channel experiments, thereby enhancing reproducibility. Furthermore, their reconfigurable Artix-7 architecture provides precise control over placement and routing, ensuring consistent RO-based frequency measurements across devices. This enables the systematic evaluation of Trojan-inserted and Trojan-free chips under comparable conditions. Importantly, demonstrating the efficacy of the proposed approach on a modest FPGA platform confirms that the methodology is not dependent on specialized or high-end hardware, thus reinforcing its broader applicability to industrial-scale scenarios.

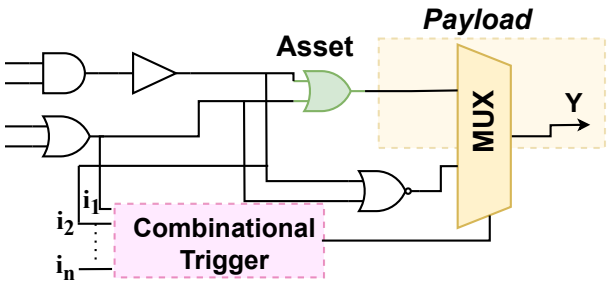


Figure 6: Implemented combinational Trojan with XNOR-based trigger and MUX-controlled payload.

5.3. Trojan Implementation and Mapping of Trigger Lengths

The inserted hardware Trojans used in this work are combinational logic circuits consisting of a Trigger and a MUX-controlled Payload, adapted from publicly available Trojan benchmarks targeting cryptographic circuits. These Trojans emulate real-world hardware threats by activating under specific logic conditions and inducing detectable power anomalies.

The Trigger is an n-bit equality detector that monitors selected internal signals (such as control/status lines or data-bus bits from the MicroBlaze/md5 core) and asserts when the observed n-bit vector matches a predefined secret pattern. In our FPGA implementation, this equality detector is realized using a chain of per-bit XNOR gates, followed by an n-input AND gate:

$$T = \bigwedge_{k=1}^{n} \mathrm{XNOR}(i_k, p_k)$$

The Trigger output T controls a Payload implemented via a MUX. When activated, the MUX injects a high-activity signal into local logic, causing a transient increase in dynamic power. To capture this effect, a Ring Oscillator (RO) is placed adjacent to the MUX switching nodes. This setup ensures the localized power perturbation induced by the Trojan activation can be captured through side-channel measurements. The overall structure is shown in Fig. 6.

Multiple trigger bit-lengths ($n = \{4, 8, 16, 32\}$) were evaluated to simulate Trojans of varying complexity and stealth. The trigger length directly determines the probability of accidental activation in normal operation: $P_{\text{accidental}} = 2^{-n}$. Hence, larger triggers such as 32-bit variants are stealthier

(2^{-32}) compared to simpler 4-bit triggers (2^{-4}). This variation enables robust evaluation of detection accuracy across different stealth regimes.

All Trojan variants were synthesized with identical placement, routing constraints, and Payload circuitry. The only variable across experiments was the trigger length, ensuring a controlled comparison of Trojan stealth and detection effectiveness. Furthermore, the intentional alteration in power profiles during Trojan activation forms the basis for the proposed detection method, which relies on clustering side-channel features without requiring a golden reference chip.

5.4. Data Collection and Preparation

The data collection process involved capturing frequency counts from each of the 32 Ring Oscillators (ROs) embedded in an FPGA under controlled input conditions. Each sample represents the frequency measurement from all 32 ROs for a given input instance. For example, a sample input for the first instance was recorded as 1.11111101110001E+113. The input vector does not represent a floating-point measurement but rather a compact scientific notation of a large integer that corresponds to the binary stimulus applied to the circuit. When expanded, this value yields the full binary input vector (e.g., `111111110000110110...`), which is then monitored by the Trojan trigger logic. Thus, the "Inputs" column provides the n-bit binary patterns ($n = 4, 8, 16, 32$ depending on the trigger length under study) that drive the DUT, while the remaining columns (`RO0`–`RO31`) contain the measured frequency responses. The FPGA was tested with 1001 distinct input instances across 18 different chips, of which 9 were Trojan-free while the remaining 9 were Trojan-infected. Consequently, for each chip, the dataset comprises 32 frequency sequences, each containing 1001 instances, yielding a total of 18×32 sequences referred to as *data sequences* in this paper.

It is essential to explicitly define what these data sequences represent to facilitate accurate interpretation. Each sequence corresponds to the frequency variations observed in a specific RO over multiple input instances. These sequences serve as indirect representations of power consumption patterns influenced by hardware Trojans. Since frequency shifts are indicative of underlying circuit modifications, clustering analysis is employed to differentiate between Trojan-free and Trojan-infected chips based on these data sequences.

A summary of the dataset is presented in Table 2:

Table 2: Summary of Dataset Characteristics

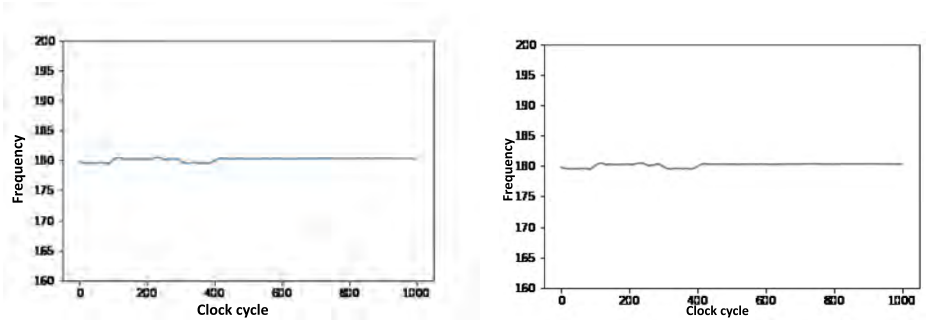
Parameter	Value
Number of Chips	18
Trojan-Free Chips	9
Trojan-Infected Chips	9
Number of ROs per chip	32
Instances per chip	1001
Total Data Sequences	576 (18 $\times$32)

The preprocessing of the dataset plays a crucial role in ensuring the quality and reliability of the subsequent analysis. Preprocessing techniques, including imputation and interquartile range analysis, were employed to remove redundant features, standardize feature values, and rectify missing data or outliers. These approaches facilitated the creation of a dataset suitable for feature extraction and clustering analysis.

Outliers, which are data points significantly deviating from the majority, can distort clustering results and impact overall accuracy. To address this, a rigorous outlier removal process was implemented. Statistical methods such as Z-score analysis were utilized to identify data points exceeding predefined thresholds, which were subsequently removed from the dataset.

Noise, often arising from random variations or measurement errors, introduces unwanted variability into the dataset. To enhance the signal-to-noise ratio, a moving average filter was applied. This filter smoothens the data by averaging values within a specified window, effectively reducing high-frequency fluctuations. The selection of an appropriate window size was guided by the characteristics of the data and the desired level of noise reduction.

By incorporating these preprocessing steps, the dataset was refined to accurately represent the underlying frequency variations, ensuring that clustering and classification models operate on high-quality input data. The refined dataset provides a robust foundation for identifying anomalies indicative of hardware Trojans and improving the overall detection accuracy.



(a) Plotting the processed data sequences (b) Grayscale

Figure 7: Preprocessing images to prepare for the image based clustering

5.5. Feature Extractions

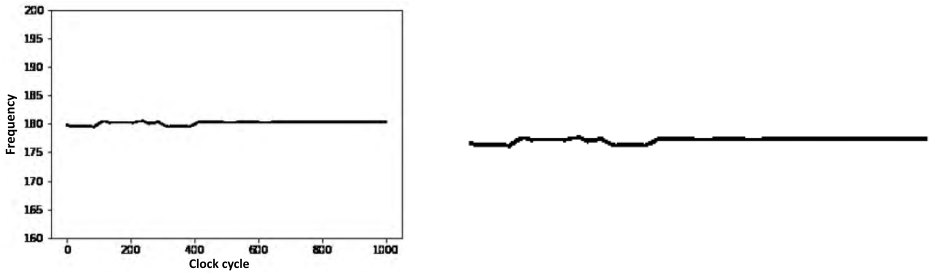
5.5.1. Imaging Technology

In the context of this study, imaging technology serves as a pivotal component in the feature extraction process, providing crucial insights into the behavior of ICs. The raw data sequences, representative of the circuit's response to RO frequencies and potential hardware Trojans, undergo a sequence of imaging procedures to facilitate subsequent analysis.

Frequency side-channel data is captured from FPGA-based Ring Oscillators (ROs), resulting in data sequences where each sequence represents the frequency response of a single RO across 1001 test instances. Each data sequence, a series of frequency values, is then plotted as a 2D graph. The x-axis of this graph represents the instance number (out of 1001), and the y-axis represents the corresponding frequency value measured from the RO. This process is repeated for each of the 32 RO data sequences, generating 32 distinct 2D graphs, each visually depicting the frequency variations of a specific RO over the tested input instances. These graphs form the basis for subsequent image processing.

The initial step involves the transformation of preprocessed data sequences into visual representations through a 2D graph, as shown in Fig. 7 (a). This graphical representation allows for a comprehensive visualization of the circuit's characteristics and aids in identifying patterns associated with RO frequencies and potential Trojan insertions. The 2D plots serve as the foundation for the subsequent imaging techniques.

To enhance the interpretability and usability of these images, the raw frequency response data is mapped to intensity values in grayscale format.



(a) Equalized Image (b) Cropped Image

Figure 8: Preprocessing images to prepare for the image based clustering

The grayscale conversion of the 2D plots is employed to simplify the data while retaining essential information, as given in Fig. 7 (b). Grayscale images reduce the complexity of color-based information, focusing on intensity variations [31]. This transformation facilitates the extraction of meaningful frequency patterns that may indicate the presence of a hardware Trojan.

Following grayscale conversion, the images undergo equalization to normalize the intensity distribution, as shown in Fig. 8 (a). Equalization enhances the contrast of the images, ensuring that features with varying intensities are more discernible. This step contributes to the overall robustness of the imaging process by standardizing the representation of different circuit characteristics.

In addition to equalization, frequency-domain transformations such as Fast Fourier Transform (FFT) are applied to extract latent features embedded within the raw data sequences. The transformed images provide an additional perspective on signal variations, highlighting subtle anomalies that may not be immediately visible in the time domain.

Fig. 8 (b) depicts the final step in which the imaging process involves cropping the equalized images to eliminate non-informative regions such as axis labels, borders, and indices. The cropped images, represented as 2D arrays of pixels, are then ready for further analysis and feature extraction.

The 2D arrays of pixels obtained from the imaging process serve as the input for feature extraction using clustering machine learning. Each pixel in the 2D array contributes to the feature vector, and k-means clustering is applied to group pixels with similar characteristics. This approach leverages spatial correlations within the images to differentiate between Trojan-free and Trojan-infected circuits effectively. The transformed images provide a

powerful visual representation of side-channel anomalies, ensuring a more reliable and interpretable analysis of hardware security threats.

5.5.2. Spectral-based features

In conjunction with the proposed imaging technology, spectral-based features offer an alternative method for extracting critical insights into the identification of Trojan-free and Trojan-inserted chips in semiconductors. Unlike the image-based approach, which visualizes the sequences of data, spectral-based features focus on analyzing the frequency, amplitude, and wavelet characteristics directly, providing a distinct perspective in the feature extraction process.

For a detailed exploration of the distribution of the extracted spectral-based features, please refer to the distribution plot shown in Fig. 9. This exploratory data analysis provides valuable insights into the characteristics of the features before they are utilized in the subsequent stages of the methodology.

The motivation behind incorporating spectral-based features lies in their ability to offer complementary information to the image-based approach. By considering both visual and frequency-based characteristics, this methodology aims to enhance the accuracy and reliability of Trojan detection in semiconductors. Further details about the experimental setup and implementation of the spectral-based features are provided in the subsequent sections.

Frequency Domain Features. The procedure entails utilizing Python, notably the NumPy and Matplotlib tools, to execute Fourier transformation on ring oscillator data. The procedure consists of using NumPy for numerical operations and Matplotlib for visualization. The steps are as follows:

(i) Import the FPGA chip data that is programmed with a ring oscillator logic, which contains time-domain samples.
(ii) Use the Fourier Transform to translate the time-domain signal into the frequency domain.
(iii) Analyze the frequency spectrum to detect dominant frequencies or frequency ranges in the ring oscillator signal.

This analysis provides insights into certain traits or behaviors of the oscillator.

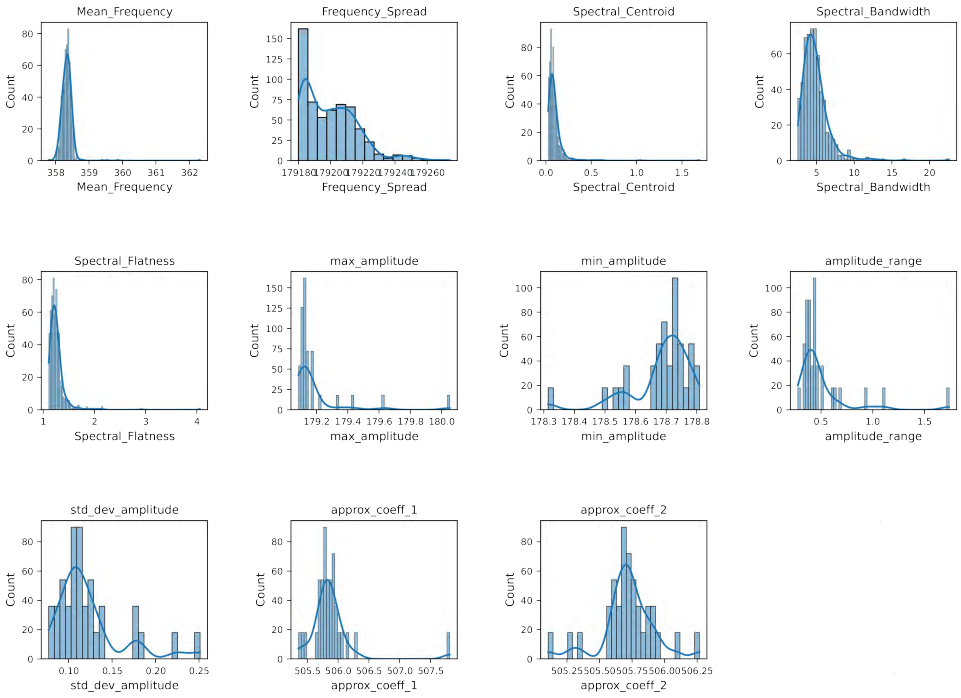


Figure 9: Distribution plot of spectral based features for exploratory image processing analysis

To calculate the frequency domain features, various mathematical techniques are applied. These features include mean frequency, frequency spread, spectral centroid, spectral bandwidth, and spectral flatness. The frequency domain features along with the mathematical equations used to generate these features are given below.

Let's assume x is the row column value.

- **Mean Frequency**: The mean ($\bar{x}$) of a set of n values $x_1, x_2, \ldots, x_n$ is calculated using the equation 2.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2}$$

- **Frequency Spread**: The mathematical phrase "max() - min()" represents the difference between the maximum value and the minimum

value of x as given in equation 3.

$$\bar{x}_{\max}() - x_{\min}() \tag{3}$$

The symbol () denotes the interval of values within a given set. This feature defines the range of the set of values. In our context, it is called frequency spread.

- **Spectral Centroid**: To compute the weighted average of the index values of x, we multiply each index value by its corresponding value in the dataset. The equation can be represented in equation 4.

$$\frac{\sum |x.\text{index} \times x|}{\sum |x|} \tag{4}$$

The same procedure is applied to each column of the RO time series values and calculate the spectral centroid, which is a weighted average of the values in the time series.

- **Spectral Bandwidth**: The given expression computes the standard deviation, also known as the root mean square deviation, of the indices x derived from a spectral centroid. This is mathematically represented as in equation 5

$$\sqrt{\frac{\sum (x.\text{index} - \text{spectral\_centroid})^2 \cdot |x|}{\sum |x|}} \tag{5}$$

This formula computes the square root of the average squared difference between the indices and the spectral centroid. The calculation is weighted by the absolute values of the items in x.

- **Spectral Flatness**: The given phrase entails the computation of the geometric mean of the absolute values of the components in a series. The equation can be represented as equation 6.

$$\exp\left(\frac{1}{n}\sum \log(1 + |\text{x}|)\right) \tag{6}$$

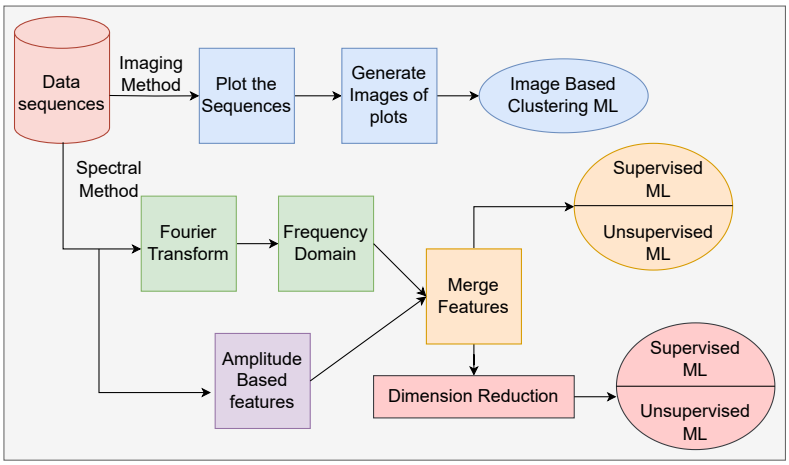


Figure 10: Feature Extraction techniques for various analysis ML models

Amplitude Domain Features. Amplitude domain features have significance when studying signals according to their amplitude properties. It mainly refers to properties or representations that express a signal in terms of its amplitude or magnitude changes. This is often contrasted with frequency-domain features, which concentrate on the flow of signal energy across various frequencies. By analyzing the highest and lowest points in these signals, behavioral changes between a Trojan-infected chip and a Trojan-free chip can be identified.

Following are some amplitude features extracted using max, min, and other basic mathematical expressions to extract more features:

- **Max Amplitude**: This feature involves identifying the maximum amplitude range across all the different Ring Oscillators (ROs) configured within the FPGA.

  Lets assume each RO column as Rc, This equation 7 represents max amplitude.

$$\mathtt{Rc}_{\max}() \tag{7}$$

- **Min Amplitude**: This feature entails the calculation of minimum amplitude range for all different ROs programmed in the FPGA.

  Lets assume each RO column is Rc, This equation goes as follows:

$$\texttt{Rc}_{\min}() \tag{8}$$

- **Amplitude Range**: The amplitude range (*Rc_range*) is the difference between the maximum amplitude and the minimum amplitude as given in Equation 9. The *max_amplitude* and *min_amplitude* values correspond to the highest and lowest frequency measurements recorded in a specific RO's data sequence. The *Rc_range* then quantifies the total spread or variability of the frequency signal. A Trojan-infected chip would likely exhibit a much wider amplitude range compared to a Trojan-free chip under the same test conditions, because the physical power anomaly from the malicious circuit significantly perturbs the normal frequency signature.

$$\texttt{Rc\_range} = \text{max\_amplitude} - \text{min\_amplitude} \tag{9}$$

Wavelet Domain Features. Wavelets are functions of math that may be utilized for analyzing data in the frequency as well as time domains concurrently [32]. The wavelet transformation breaks down a signal into a set of wavelet functions, allowing for the examination of distinct frequency elements at different scales.

We apply the operation of wavelet decomposition. It is applied to the dataset signals at level 3 using the wavelet type 'db1. It decomposes the signal into approximation coefficients and detail coefficients at different scales or levels. We are specifying a decomposition level of 3.

```
pywt.wavedec(df[col], 'db1', level=3)
```

The above is the Python code for wavelet transformation, where we are using the Python library 'pywt' with a wavelet type "db1" and a decomposition level of 3. Here "db1" is first-order Daubechies wavelet which means it has the shortest filter length (2 coefficients) within Daubechies family. We take both of the coefficients as additional 2 features from our data generated from RO. Db1 are considered to have good feature importance while distinguishing Trojan free and Trojan inserted because it is effective in capturing sharp transitions and high-frequency details in signals and also, it ensures perfect signal reconstruction [33].

6. ML-Assisted Analysis Methods

6.1. Image based Clustering Model

The input for feature extraction relies on the 2D pixel arrays derived from the imaging process. Employing the k-means clustering algorithm, an influential unsupervised learning technique, the data points in these arrays are systematically grouped into distinct clusters according to their similarities. In our study, individual pixels within the 2D array collectively form the feature vector, and the k-means clustering methodology is employed to categorize pixels exhibiting comparable characteristics.

The choice of k, the number of clusters, is critical and is determined based on the inherent structure of the data. The resulting clusters represent distinct patterns or features in the circuit images. Each cluster centroid encapsulates the characteristics of the pixels it represents, serving as a condensed representation of the original image.

The feature vectors extracted through k-means clustering encapsulate spatial patterns and intensity variations within the circuit images. These features are instrumental in discerning subtle differences related to RO frequencies and potential hardware Trojan insertions. The clustering results contribute valuable information for subsequent analysis, aiding in the identification and classification of ICs based on their unique characteristics.

The integration of imaging technology and k-means clustering in our feature extraction process enhances the overall effectiveness of the unsupervised clustering analysis, providing a comprehensive understanding of the underlying patterns within the dataset.

6.2. Spectral Feature-Based Machine Learning Models

The spectral-based features extracted from the frequency domain, amplitude domain, and wavelet domain provide valuable information for detecting potential hardware Trojans in ICs. To harness this information, we employ both supervised and unsupervised machine learning models. Initially, these features undergo dimension reduction techniques, and their performance is evaluated. Subsequently, the spectral features are utilized without dimension reduction, and the results are compared to determine the most effective approach.

6.2.1. Dimension Reduction

Dimension reduction is an essential step in handling high-dimensional data, especially when dealing with spectral-based features. In our study, we focus on reducing the dimensionality of the feature space containing 11 spectral-based features. The primary goal is to emphasize the most influential features while discarding less informative ones, enhancing the efficiency and interpretability of our machine learning models.

Principal Component Analysis (PCA). PCA is applied to the spectral feature space in order to convert the original features into a new collection of orthogonal variables called principle components. These components are ordered by their variance, allowing us to retain a subset of the most significant components that capture the majority of the variability in the data. By selecting a reduced number of principal components, we aim to concentrate on the critical aspects of the spectral features, improving model performance and interpretability.

t-Distributed Stochastic Neighbor Embedding (t-SNE). While t-SNE is a powerful tool for visualizing data with numerous dimensions, in our situation, it is not applied directly to images but rather to the spectral feature space. t-SNE aids in revealing relationships and patterns within the spectral features by projecting them into a lower-dimensional space while preserving pairwise similarities. This assists in better understanding the intrinsic structure of the feature space and can be particularly useful for identifying clusters and outliers.

Linear Discriminant Analysis (LDA). LDA, although often associated with classification tasks, is utilized here to find a linear combination of spectral features that maximally separates different classes. By focusing on the features that contribute most to class discrimination, LDA aids in highlighting the aspects of the spectral data crucial for detecting variations in RO frequencies and potential hardware Trojans.

6.2.2. Machine Learning Models

Supervised Machine Learning Models. We utilize several well-established supervised machine learning models to individually assess their performance in classifying ICs based on their spectral features. These models include ADABoost, Random Forest, XGBoost, LightGBM, K-Nearest Neighbors (KNN) and NGBoost.

ADABoost, Random Forest, XGBoost, LightGBM: These models are employed separately to evaluate their individual classification capabilities. Each model is trained and tested independently to understand its effectiveness in discerning patterns related to RO frequencies and potential hardware Trojan insertions. The extracted features were also evaluated using NGBoost, a recently developed supervised learning technique which is also applicable for anomaly detection [34].

K-Nearest Neighbors (KNN): KNN is applied separately as a non-parametric, instance-based learning technique to classify RO frequency data according to a majority class of closest neighbors. This approach enables for a targeted study of its effectiveness in collecting local patterns inside the spectral-based feature space.

Unsupervised Machine Learning Models. For unsupervised learning, we employ clustering algorithms to identify patterns and groupings within the dataset. The models include K-Means, K-Means++, AGNES, Spectral Clustering, and Self-Organizing Maps (SOM).

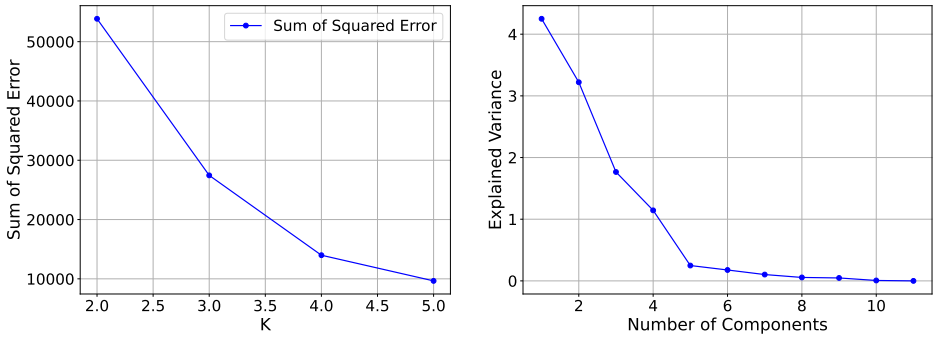
K-Means, K-Means++, AGNES, Spectral Clustering: These clustering algorithms are applied individually to group similar data points together, aiding in the identification of inherent structures and patterns. They are particularly valuable in unsupervised scenarios where the class labels are unknown.

Self-Organizing Maps (SOM): SOM is employed individually as a neural network-based approach that organizes data points into a grid, preserving the topological relationships between them. This approach allows for an in-depth analysis of its effectiveness in capturing complex relationships within high-dimensional data and revealing underlying structures.

By evaluating each model independently, we aim to gain insights into their individual strengths and weaknesses in the context of spectral-based features, providing a comprehensive approach for detecting hardware Trojans and understanding the distinct characteristics associated with different RO frequencies.

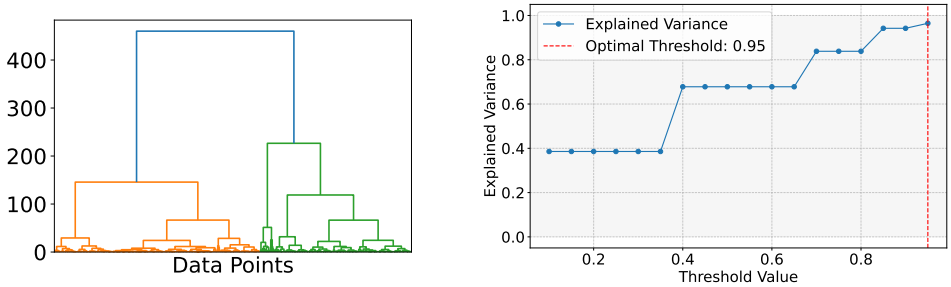
6.3. Optimization of Machine Learning Models

In our study, optimization holds a crucial role in enhancing the performance of various machine learning models deployed for hardware Trojan detection. Diverse optimization techniques are employed and tailored to the



(a) Number of Kmeans Clusters (b) PCA components

Figure 11: Elbow plot for optimized parameters



(a) AGNES dendrogram plot (b) PCA threshold optimization

Figure 12: Optimization for PCA and AGNES

specific characteristics of each model, ensuring that they operate at peak efficiency.

6.3.1. K-Means Clustering Optimization

Determining the optimal number of clusters was crucial for the K-Means clustering algorithm. Leveraging the elbow technique, the ideal point is determined by running K-Means with varied cluster numbers and graphing the within-cluster sum of squares (WCSS) against the number of clusters. The best number of clusters, detected at the elbow point, was found to be 4, striking a balance between decreasing within-cluster variation and avoiding overfitting as seen in Fig. 11(a).

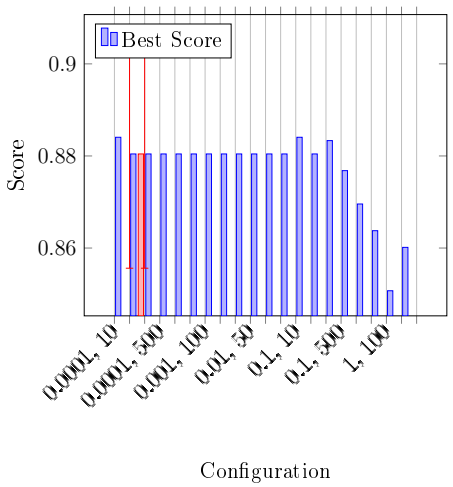


Figure 13: Grid Search Results for Parameter Optimizing for ADABoost

6.3.2. AGNES Hierarchical Clustering Optimization

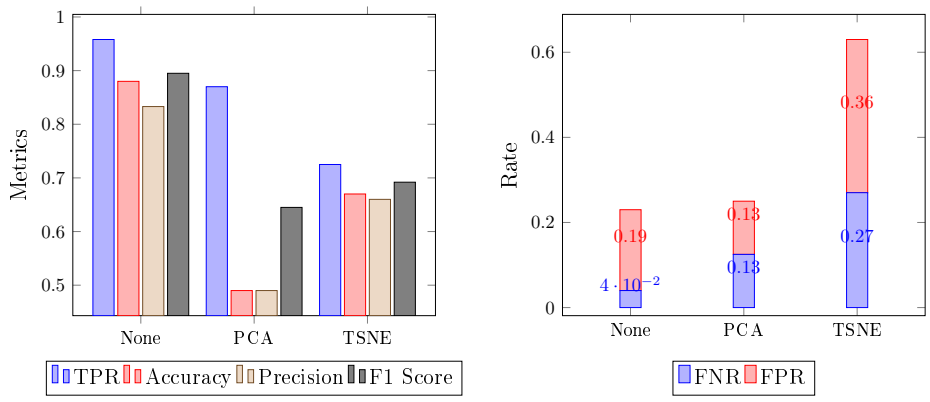
AGNES (Agglomerative Nesting) relied on hierarchical clustering, and optimization involved constructing a dendrogram. The hierarchical structure and cluster relationships were visualized, aiding in the determination of the optimal number of clusters, which, in our case, was identified as 4 as shown in Fig. 12(a).

6.3.3. PCA and Dimension Reduction Optimization

Optimizing Principal Component Analysis (PCA) involved two steps. First, the optimal threshold is determined for retaining components by systematically exploring various threshold values which was 0.95 as depicted by Fig. 12(b). The goal is to maximize explained variance while minimizing the number of components. Additionally, the elbow method is applied to find the optimal number of components, which, in our case, was identified as 4 as shown in Fig. 11(b). This step ensurs that the retained components provided a balance between information preservation and dimensionality reduction.

6.3.4. ADABoost Classifier Parameter Optimization

To enhance the ADABoost classifier's performance in hardware Trojan detection, an extensive grid search optimization is conducted as given in Fig. 13. This procedure required exploring a range of hyperparameter configurations, specifically the learning rate and the number of estimators. The ideal



(a) Performance comparison of different methods

(b) Error metrics with different techniques

Figure 14: Comparison of different feature dimension reduction techniques with Kmeans Clustering model

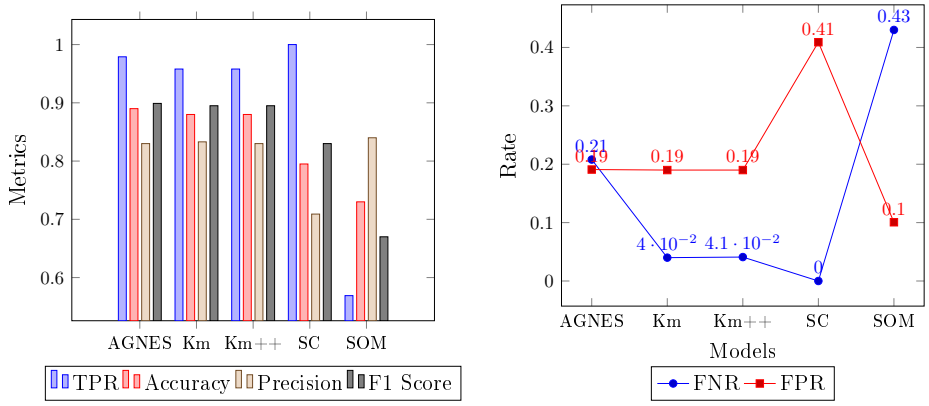
configuration learning_rate = 0.01, number of estimators = 500 is determined by grid search, resulting in a score of 0.884058. This comprehensive optimization improved the ADABoost classifier's efficacy, showcasing a balanced trade-off between the learning rate and the number of estimators.

7. Results and Discussion

In this research paper, multiple approaches is explored for classifying hardware Trojans using the provided data. Initially, an unsupervised approach is employed, specifically Kmeans Clustering. A comparative analysis is conducted by employing different feature dimension reduction techniques and without reducing the feature dimensions. Interestingly, the best results are achieved without reducing the feature dimensions.

Similarly, in the realm of supervised learning, ADABoost is utilized and observed that the optimal results are obtained without employing feature dimension reduction. Our comparisons are extended to various unsupervised models utilizing spectral-based features directly, without reducing the dimension. The same methodology is repeated for various supervised learning models.

Ultimately, a thorough comparison is carried out between the top-performing unsupervised model and the top-performing supervised models, integrating



(a) Performance comparison of different methods

(b) Error metrics with different techniques

Figure 15: Comparison of different unsupervised models

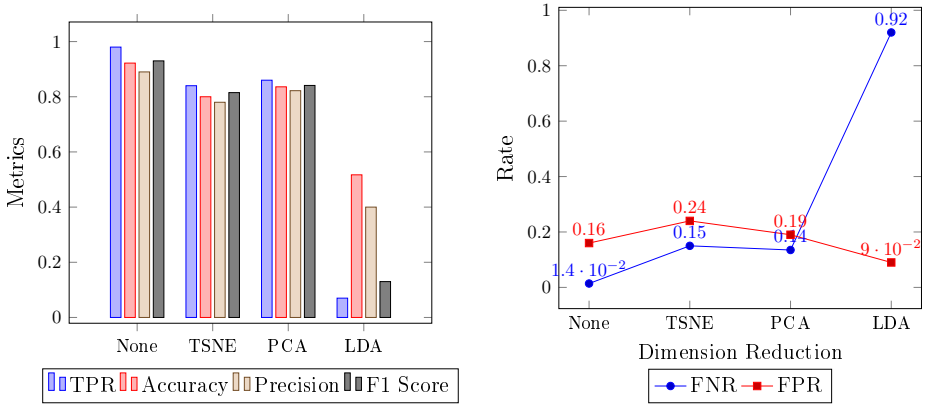
the most efficient feature extraction approach. A clustering algorithm is incoporated based on images into the comparison. Our results suggest that the image-based clustering strategy performed better than the other studies. Detailed results are reported in the ensuing subsections.

7.1. Comparison with unsupervised learning

Figure 14 (a) represents a comparative analysis of feature performance with different dimension reduction techniques in conjunction with the Kmeans Clustering model. It is evident that not employing any dimension reduction technique (None) yields the highest True Positive Rate(TPR) (0.958) and F-measure(F1 score) (0.895). However, PCA performs well in terms of TPR (0.87) and is competitive in Accuracy (0.645). TSNE, although not leading in any specific metric, demonstrates balanced performance across all metrics.

Figure 14 (b) compares the False Negative Rate (FNR) and False Positive Rate (FPR) for three different feature dimension reduction techniques: None, PCA, and TSNE. The chart shows that without any dimensional reduction, it gives the lowest FNR at 0.004 and the highest FPR at 0.23. PCA has a balanced performance with both FNR and FPR at around 0.13 and 0.25, respectively, while TSNE has an increased FNR of 0.27 and an FPR of 0.63. The chart is generated using the Kmeans Clustering model.

Figure 15 (a) evaluates the performance metrics of five unsupervised machine learning models: AGNES, Kmeans (Km), Kmeans (Km++), Spectral



(a) Performance comparison with different dimension reduction techniques

(b) Error comparison with different dimension reduction techniques

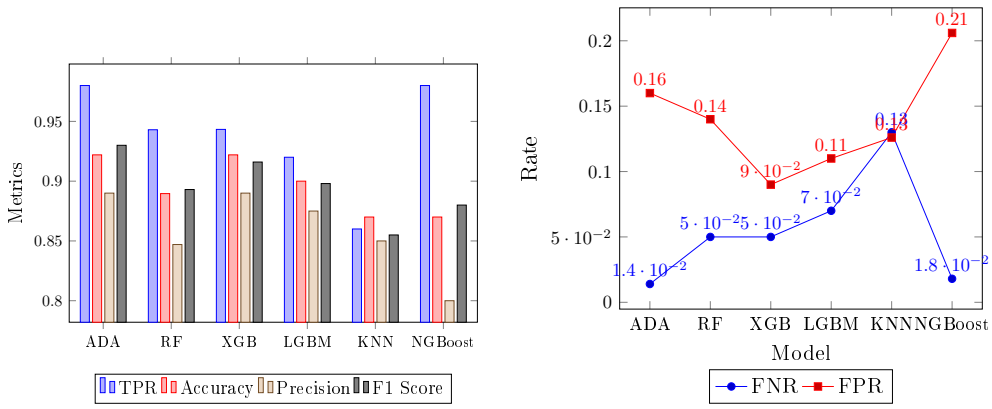
Figure 16: Comparison of different feature dimension reduction techniques with ADABoost model

Clustering (SC), and SOM. The chart shows that AGNES has the highest TPR and Accuracy of 0.89 but a lower Precision of around 0.83, and an F1 Score of 0.899, close to 0.9. Km has moderate values in all four metrics ranging from approximately 0.833 to 0.95. Km++ shows results similar to Km with almost identical results. SC has the lowest Precision of 0.709 and an F1 score around 0.83 but a maximum TPR of 1 and Accuracy around 0.795. SOM exhibits low values in all four metrics, with none exceeding 0.85.

In Figure 15 (b), FNR and FPR are compared across five unsupervised models: AGNES, Km, Km++, SC, and SOM. Lower values indicate higher performance. The SOM model excels in minimizing false positives at the expense of higher false negatives. AGNES has FNR at 0.208, and the lower FPR is at 0.1909. The SOM model performs well with a reduced FNR of 0.43 and a remarkable dip in FPR to just above zero at approximately 0.1.

7.2. Comparison with supervised models

Figure 16 (a) compares the performance of the ADABoost model against four dimension reduction techniques: None, TSNE, PCA, and LDA. The chart's bars reflect the metric values for each dimension reduction approach. It is found that eliminating any dimension reduction approach (None) resulted in the best performance across all criteria. TSNE likewise worked



(a) Performance comparison of Accuracy, (b) Performance comparison of FNR and TPR, Precision, and F1 score of supervised FPR of supervised models
models

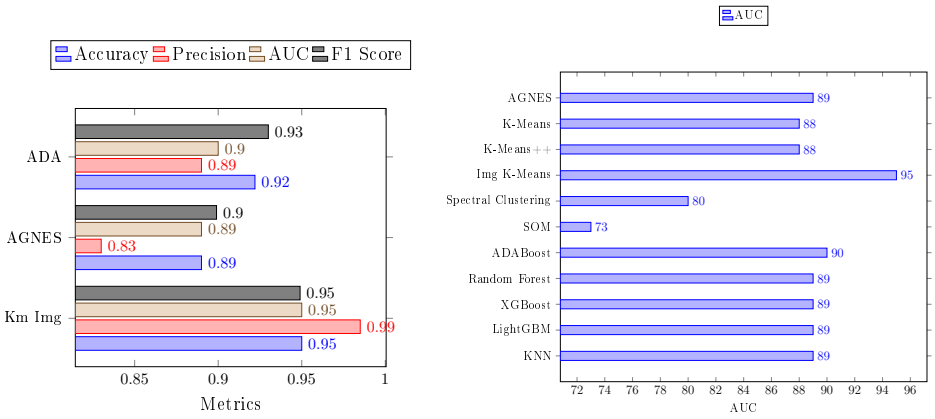
Figure 17: Comparison of performance metrics for supervised models

admirably but was somewhat inferior to no dimension reduction. LDA has the lowest performance across all metrics.

In Figure 16 (b), ADABoost model performance is compared across four dimension reduction techniques (None, TSNE, PCA, LDA), assessing false positive rate (FPR) and false negative rate (FNR), where lower values indicate higher performance. Shifting from no dimension reduction to TSNE, PCA, and LDA leads to a notable decrease in FPR, but an opposite trend is evident in FNR, particularly with LDA rising to 0.92. The data highlights the diverse impact of dimension reduction strategies on model performance indicators.

Figure 17 (a) compares the performance of five distinct supervised machine learning models: ADA, RF, XGB, LGBM, and KNN. The ADA model demonstrats greatest performance in terms of TPR, with a score approaching 0.98 and accuracy of 0.92. The RF model shows moderate results with high TPR of 0.943 and Accuracy and F1 score of 0.8896 and 0.893 respectively. The F1 Score is reasonably stable across all models, with the exception of KNN, which lagged somewhat behind.

Figure 17 (b) evaluted the performance of five supervised machine learning models using FPR and FNR metrics. In this graph lower the value is considered higher performance. The chart shows that ADA exhibited the highest FPR at 0.16 but maintained a low FNR of 1.4 x $10^{-}2$. Conversely,



(a) Performance comparison of different clustering models

(b) AUC comparison of different clustering models

Figure 18: Comparison of Image Processing method with best performing supervised and unsupervised models

KNN demonstrats the heightest elvation FNR at 0.126 while having an elevated FPR of 0.12. The other models - RF, XGB, and LGBM - balanced both rates moderately with no extreme variations observed.

7.3. Comprehensive comparison of best models with image based clustering technique

Figure 18 (a) presents a comparison of the performance of image-based Kmeans clustering with spectral features-based Kmeans, AGNES, and ADABoost. The Image K-means model is the most effective in categorizing Trojan inserted data and Trojan free data, with the greatest accuracy of 95% and F1 Score of 94.9% compared to other models. In contrast, K-means with spectral feature based exhibits comparatively lower performance with an accuracy of 89% and the lowest F1 Score at 89%. Supervised model ADABoost is better performing than the unsupervised spectral feature-based clustering but still image based clustering outperformed ADABoost model. The robust performance of Image K-means can be attributed to its efficient identification and grouping of similar patterns in image data, contributing to higher accuracy and F1 Score.

In Figure 18 (b), eleven distinct segments illustrate various machine learning models, each identified by its name and corresponding AUC (Area Under the Curve) percentage. The AUC, derived from the Receiver Operating

Characteristic (ROC) curve analysis, serves as a metric for assessing the model's ability to discriminate between classes. The comparison highlights Image K-means as the standout performer, boasting a remarkable AUC of 95%. Notably, K-means, K-means++, Spectral Clustering, SOM, and Img Kmeans exhibit comparatively lower AUC values. The conclusion drawn is that Image K-means excels in discriminating between clusters in image data, indicating its effectiveness in capturing intricate patterns. This superior discriminative ability contributes to its overall strong performance in the image-based clustering task.

These findings clarify why the proposed image-based approach achieves higher accuracy than traditional side-channel analysis. The reported 95% detection accuracy stems not from side-channel analysis alone, but from a synergistic, noise-resilient pipeline. Traditional SCA methods rely on raw signal traces, which are highly susceptible to process noise and manufacturing variability which are the factors that reduce the separability between Trojan-infected and clean chips. In contrast, our approach begins with RONs that capture stable frequency variations, which are then transformed into structured visual representations through time–frequency imaging. This encoding enhances the signal-to-noise ratio and exposes multi-dimensional dependencies that are difficult to detect in one-dimensional traces. Finally, unsupervised clustering models operate on these enriched representations to achieve robust anomaly detection. Together, this integrated process explains the superior performance of our image-based method compared to conventional SCA techniques.

7.4. Comparative Analysis of Trojan-Free and Trojan-Infected Circuits

A key challenge in side-channel based detection is to ensure that the observed frequency perturbations are uniquely attributable to Trojan activation and not merely to variations caused by executing a more computationally intensive task. To address this concern, our experimental framework leverages the dense network of 32 ROs distributed across the FPGA fabric. This network enables a localized on-chip golden reference: in Trojan-infected chips, the majority of ROs continue to exhibit nominal behavior that closely matches Trojan-free chips. Consequently, deviations can be spatially and statistically isolated to ROs located in the vicinity of the Trojan payload.

Baseline Stability. In both Trojan-free chips and in non-triggering input instances of Trojan-infected chips, the measured frequencies remain tightly

clustered around ∼180 MHz across the input sequence. This stability is evident in Figure 7, where the latter half of the RO measurements maintain consistent operation. This occurs because the Trojan was already activated briefly and deactivated during the first half of the applied input sequence. If the anomaly were caused by a global voltage drop or by a computationally intensive workload, the perturbation would appear uniformly across the entire input sequence and across all ROs, which is not observed.

Localized Anomalies. In contrast, ROs located in close physical proximity to the Trojan payload (e.g., R1 in Chip 18) exhibit distinct frequency drops during specific input ranges (e.g., inputs 1–100 and 301–400 in our dataset). This pattern is restricted and repeatable, aligning with the expected behavior of a stealthy, combinationally-triggered Trojan.

Statistical Differentiation. Table 3 summarizes the statistical comparison between representative Trojan-free and Trojan-infected ROs. While the mean frequency across cases remains same, the standard deviation of the Trojan-adjacent RO is significantly higher, confirming the presence of localized perturbations that cannot be explained by normal computational load.

Table 3: Comparative Statistical Summary of Ring Oscillator Frequencies (MHz)

RO Identifier	N (Inputs)	Min	Max	Mean	SD
Chip 1, R1 (Trojan-Free Baseline)	1001	180.36	181.25	180.68	0.10
Chip 18, R1 (Closest to Trojan)	1001	176.81	181.20	180.68	0.53
Chip 18, R18 (Farther from Trojan)	1001	180.18	180.92	180.68	0.14

These findings confirm that the anomalous frequency drops are both *localized* and *statistically distinct*, providing strong evidence that the perturbations originate from Trojan activation rather than from ordinary workload-induced voltage fluctuations.

8. Conclusion

The manufacturing process of the semiconductor supply chain comprises several stages, from designing chips to their production, which involves fabrication, assembly, packaging, and quality assurance tests, culminating in the delivery of reliable and secure electronic products to the final consumer. By processing image data extracted using reprogrammable hardware devices

in the form of 2D images, our proposed AI-enabled image processing approach extracts features and not only outperforms traditional methods but also achieves an impressive 95% accuracy in Hardware Trojan detection without relying on costly golden references.

The comprehensive comparison of unsupervised and supervised models underscores the superiority of the image-based clustering technique, especially Image K-means, in terms of accuracy, precision, and F1 Score. The results show that the proposed image-based unsupervised learning approach exceeds the performance of traditional supervised learning methods, challenging the widespread assumption that supervised models typically demonstrate superior performance. This innovative research is oriented towards a golden-reference-free methodology for hardware Trojan detection, utilizing an image-based unsupervised learning technique.

The proposed approach improves efficiency over traditional methods while reducing dependence on golden data for anomaly detection in consumer electronics. This breakthrough conclusion redefines the landscape of Hardware Trojan detection, offering a cost-effective and highly accurate alternative suitable for securing IC chips in consumer electronics. The integration of unsupervised learning techniques not only advances detection accuracy but also optimizes resource utilization, marking a significant stride in ensuring the security of consumer electronic devices.

Declarations

Ethical Approval

This research did not involve the use of human or animal subjects, sensitive data, or any other ethical considerations that would require approval from an ethics committee or institutional review board.

Competing interests

The authors declare no competing interests that could bias the results or interpretation of this research. This study was conducted without any financial, personal, or other relationships that could be perceived as a conflict of interest.

Authors' contributions

A.G. was responsible for the conception and design of the research, oversaw the experimental setup, led the development of machine learning models, implemented the algorithms, handled data preprocessing, and performed

comparative analyses. M.A. wrote the initial draft of the manuscript and contributed to the experiments and methodological framework. F.A. (the corresponding author) provided guidance on research design and methodology. F.A., N.Z.J., and S.M. supervised the project. All authors reviewed, revised, and approved the final manuscript for publication.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used GPT-4 in order to enhance the grammar and proofreading. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

[1] A. Jain, Z. Zhou, 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] S. Jha, S. K. Jha, Randomization based probabilistic approach to detect trojan circuits, in: 2008 11th IEEE High Assurance Systems Engineering Symposium, 2008, pp. 117–124. doi:10.1109/HASE.2008.37.

[3] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, S. Bhunia, Mero: A statistical approach for hardware trojan detection, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer, 2009, pp. 396–410.

[4] J. Kumagai, Chip detectives [reverse engineering], IEEE Spectrum 37 (11) (2000) 43–48. doi:10.1109/6.880953.

[5] A. Stern, D. Mehta, S. Tajik, U. Guin, F. Farahmandi, M. Tehranipoor, Sparta-cots: A laser probing approach for sequential trojan detection in cots integrated circuits, in: 2020 IEEE Physical Assurance and Inspection of Electronics (PAINE), IEEE, 2020, pp. 1–6.

[6] J. Francq, F. Frick, Introduction to hardware trojan detection methods, in: 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2015, pp. 770–775.

[7] S. Narasimhan, X. Wang, D. Du, R. S. Chakraborty, S. Bhunia, Tesr: A robust temporal self-referencing approach for hardware trojan detection, in: 2011 IEEE International Symposium on Hardware-Oriented Security and Trust, IEEE, 2011, pp. 71–74.

[8] Y. Jin, Y. Makris, Hardware trojan detection using path delay fingerprint, in: 2008 IEEE International workshop on hardware-oriented security and trust, IEEE, 2008, pp. 51–57.

[9] R. Rad, J. Plusquellic, M. Tehranipoor, Sensitivity analysis to hardware trojans using power supply transient signals, in: 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, IEEE, 2008, pp. 3–7.

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

[11] H. Salmani, M. Tehranipoor, R. Karri, On design vulnerability analysis and trust benchmarks development, in: 2013 IEEE 31st international conference on computer design (ICCD), IEEE, 2013, pp. 471–474.

[12] A. Ghimire, A. N. Asiri, B. Hildebrand, F. Amsaad, Implementation of secure and privacy-aware ai hardware using distributed federated learning, in: 2023 IEEE 16th Dallas Circuits and Systems Conference (DCAS), IEEE, 2023, pp. 1–6.

[13] A. Ghimire, V. V. Baligodugula, F. Amsaad, Power analysis side-channel attacks on same and cross-device settings: A survey of machine learning techniques, in: IFIP International Internet of Things Conference, Springer, 2023, pp. 357–367.

[14] K. Das, R. N. Behera, A survey on machine learning: concept, algorithms and applications, International Journal of Innovative Research in Computer and Communication Engineering 5 (2) (2017) 1301–1309.

[15] B. Mahesh, Machine learning algorithms-a review, International Journal of Science and Research (IJSR).[Internet] 9 (1) (2020) 381–386.

[16] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, M. Tehranipoor, Hardware trojans: Lessons learned after one decade of research, ACM Transactions on Design Automation of Electronic Systems (TODAES) 22 (1) (2016) 1–23.

[17] S. Bhasin, J.-L. Danger, S. Guilley, X. T. Ngo, L. Sauvage, Hardware trojan horses in cryptographic ip cores, in: 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, IEEE, 2013, pp. 15–29.

[18] J. Cruz, F. Farahmandi, A. Ahmed, P. Mishra, Hardware trojan detection using atpg and model checking, in: 2018 31st international conference on VLSI design and 2018 17th international conference on embedded systems (VLSID), IEEE, 2018, pp. 91–96.

[19] S. K. Haider, C. Jin, M. Ahmad, D. M. Shila, O. Khan, M. van Dijk, Advancing the state-of-the-art in hardware trojans detection, IEEE Transactions on Dependable and Secure Computing 16 (1) (2017) 18–32.

[20] S. Kaur, B. Singh, H. Kaur, Stratification of hardware attacks: Side channel attacks and fault injection techniques, SN Computer Science 2 (3) (2021) 183.

[21] K. Xiao, X. Zhang, M. Tehranipoor, A clock sweeping technique for detecting hardware trojans impacting circuits delay, IEEE Design & Test 30 (2) (2013) 26–34.

[22] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar, Trojan detection using ic fingerprinting, in: 2007 IEEE Symposium on Security and Privacy (SP'07), IEEE, 2007, pp. 296–310.

[23] Z. Zhang, J. Dofe, P. Yellu, Q. Yu, Comprehensive analysis on hardware trojans in 3d ics: Characterization and experimental impact assessment, SN Computer Science 1 (4) (2020) 233.

[24] B. Zhou, R. Adato, M. Zangeneh, T. Yang, A. Uyar, B. Goldberg, S. Unlu, A. Joshi, Detecting hardware trojans using backside optical imaging of embedded watermarks, in: Proceedings of the 52nd Annual Design Automation Conference, DAC '15, Association for Computing Machinery, New York, NY, USA, 2015. doi:10.1145/2744769.2744822.
URL https://doi.org/10.1145/2744769.2744822

[25] X. Zhang, M. Tehranipoor, Ron: An on-chip ring oscillator network for hardware trojan detection, in: 2011 Design, Automation & Test in Europe, IEEE, 2011, pp. 1–6.

[26] A. Ferraiuolo, X. Zhang, 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.

[27] I. Kuon, R. Tessier, J. Rose, FPGA Architecture: Survey and Challenges, Vol. 2, Foundations and Trends® in Electronic Design Automation, 2008. doi:10.1561/1000000005.
URL http://dx.doi.org/10.1561/1000000005

[28] S. Mangard, E. Oswald, T. Popp, Power analysis attacks: Revealing the secrets of smart cards, Vol. 31, Springer Science & Business Media, 2008.

[29] S. Waybhase, P. Adakane, Data security using advanced encryption standard (aes), International Journal of Engineering Research & Technology (IJERT) 11 (06) (2022).

[30] A. P. Fournaris, L. Pyrgas, P. Kitsos, An efficient multi-parameter approach for fpga hardware trojan detection, Microprocessors and Microsystems 71 (2019) 102863.

[31] A. Ghimire, A. Chapagain, U. Bhattarai, A. Jaiswal, Nepali handwriting recognition using convolution neural network, International Research Journal of Innovations in Engineering and Technology 4 (5) (2020) 5.

[32] P. Brémaud, Mathematical principles of signal processing: Fourier and wavelet analysis, Springer, 2002.

[33] S. Bahri, L. Awalushaumi, M. Susanto, The approximation of nonlinear function using daubechies and symlets wavelets, in: Proceeding of The First International Conference on Mathematics and Islam, 2018, pp. 300–306.

[34] W. Wang, W. Liu, X. Kong, W. Ding, M. Chen, M. Li, M. Li, T. Qin, L. Zhu, Research on cigarette moisture anomaly risk identification based

on an improved ngboost algorithm, in: International Workshop on Automation, Control, and Communication Engineering (IWACCE 2024), Vol. 13394, SPIE, 2024, pp. 296–306.