A Golden-Free Unsupervised ML-Assisted Security Approach for Detection of IC Hardware Trojans

ASHUTOSH GHIMIRE and MOHAMMED ALKURDI, Wright State University, USA
MD TAUHIDUR RAHMAN, Florida International University, Florida
SARAJU MOHANTY, University of North Texas, USA
FATHI AMSAAD*, Wright State University, USA

Hardware Trojans are deliberate malicious hardware modifications inserted in semiconductor Integrated circuits (ICs) for the purpose of stealing or leaking sensitive information, as well as disrupting critical systems upon activation, underscoring the importance of robust detection mechanisms. Emerging hardware security research highlights the criticality of employing AI for effective detection within the semiconductor IC supply chain. The efficient detection of these malicious trojan circuits is of utmost significance, as it holds paramount importance in cultivating trust within the semiconductor IC supply chain. However, prevailing detection methodologies, predominantly reliant on side-channel analysis, often necessitate the utilization of golden chips for validation. This paper heralds a new era in hardware trojan detection, harnessing the prowess of unsupervised machine learning in conjunction with side-channel analysis to eliminate the need for golden data. Employing unsupervised clustering, the methodology not only showcased a superior false positive rate but also demonstrated a comparable accuracy level when compared to supervised counterparts such as the K-Nearest Neighbors (KNN) classifier, Support Vector Machine (SVM), and Gaussian classifier—methods reliant on the availability of golden data for training. Notably, the proposed model exhibited an impressive accuracy rate of 93%, particularly excelling in pinpointing diminutive trojans triggered by concise events, surpassing the capabilities of preceding techniques. In conclusion, this research advances a groundbreaking paradigm in hardware trojan detection, accentuating its potential in bolstering the integrity of semiconductor IC supply chains.

CCS Concepts: • **Security and privacy → Malicious design modifications**; *Side-channel analysis and countermeasures*; *Tamper-proof and tamper-resistant designs*; • **Computing methodologies → Cluster analysis**; • **Hardware →** *Process, voltage and temperature variations*; **Post-manufacture validation and debug**.

Additional Key Words and Phrases: Hardware Trojan Detection, Unsupervised Machine Learning, Side-Channel Analysis, Semiconductor IC Supply Chain, Anomaly Detection

ACM Reference Format:
Ashutosh Ghimire, Mohammed Alkurdi, Md Tauhidur Rahman, Saraju Mohanty, and Fathi Amsaad. 2018. A Golden-Free Unsupervised ML-Assisted Security Approach for Detection of IC Hardware Trojans. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 19 pages. https://doi.org/XXXXXXX.XXXXXXX

*This author is the corresponding author.

Authors' Contact Information: Ashutosh Ghimire, ghimire@wright.edu; Mohammed Alkurdi, alkurdi.2@wright.edu, Wright State University, Dayton, Ohio, USA; Md Tauhidur Rahman, Florida International University, Miami, Florida, mdtrahma@fiu.edu; Saraju Mohanty, University of North Texas, Denton, Texas, USA; Fathi Amsaad, Wright State University, Dayton, Ohio, USA.

Manuscript submitted to ACM

1 INTRODUCTION

The domain of hardware trojan detection within integrated circuits (ICs) has undergone swift evolution, aiming to fortify reliability and security across the landscape of design, manufacturing, and validation phases. Nevertheless, when the fabrication of ICs is delegated to untrusted entities, the looming prospect of potential adversarial manipulations sparks significant apprehensions. Such manipulations carry the potential to trigger operational breakdowns, expose sensitive data, or compromise overall reliability [13].

Although reverse engineering (RE) stands as a robust mechanism for hardware trojan detection, its feasibility is hampered by the constraints of time and cost, particularly when confronted with a diverse array of ICs. On the other hand, non-destructive methodologies, represented by side-channel analysis (SCA), provide swift results; however, they wrestle with the challenge of differentiating between variations introduced by the manufacturing process and genuine hardware trojans [21].

Recent progressions in SCA have been focusing on ameliorating variability-induced complexities by integrating supervised machine learning, thereby enhancing the efficacy of trojan detection. By establishing a clear distinction between Trojans-free and Trojans-infected chips, this approach holds considerable potential. Nevertheless, this approach is limited by its reliance on trustworthy authenticated ICs (Golden References) [14, 27].

Conventional methodologies employed for hardware trojan detection through side-channel analysis (SCA) have traditionally leaned heavily on the utilization of cost-intensive and challenging-to-access golden ICs (Golden References). Innovations, such as self-referencing techniques, have arisen to overcome this challenge, leveraging side-channel data derived from distinct instances or locations within the same component. Notwithstanding these advancements, these strategies are not exempt from their inherent constraints, often demanding the presence of a minimum of one Trojan-free site and struggling to effectively address combinational Trojans.

The advent of machine learning in recent times has introduced new novel detection paradigms within the purview of side-channel analysis, that liberate themselves from the requirement of golden ICs. A pertinent illustration of this emerging trend is encapsulated within a supervised learning-based approach delineated in [26], which detects anomalies in supply current through simulation models as a response to trojan attacks. Nonetheless, the feasibility of its application to classifying data gleaned from real-world fabricated chips remains an enigma.

Another alternative approach employs a machine-learning model based on human temporal memory (HTM), eliminating the need for a golden chip [5]. Additionally, unsupervised clustering methods, exemplified in [27], use power signatures to partition ICs into clusters, anticipating distinctions between ICs with trojans and their unaltered counterparts. Similarly, [1] explores this terrain using quantum diamond microscope magnetic field images. These models advocate a partial reverse engineering (RE) approach for selected ICs, aiming to reduce time, costs, and expertise required by traditional paradigms relying on golden ICs.

Given the escalating demand for potent and streamlined hardware trojan detection methodologies, this paper introduces an innovative paradigm that amalgamates on-chip sensors with unsupervised machine learning for the purpose of trojan detection. This innovative approach involves the seamless integration of a network of ring oscillators (RO) into the IC's architecture during the design phase. Subsequent to the fabrication process, an extensive analysis of RO data from all potentially suspect ICs is undertaken, harnessing unsupervised clustering techniques to reveal any latent trojans permeating the system.

1.1 Key Contributions of the research

This study introduces a significant stride in the field of hardware trojan detection, aiming to counter deliberate malicious hardware modifications within semiconductor integrated circuits (ICs). These modifications can lead to information leakage and system disruption, necessitating robust detection methods to ensure the security and trustworthiness of semiconductor IC supply chains. Recent trends emphasize the role of artificial intelligence (AI) in effective trojan detection within these supply chains, and this research aligns with that direction.
The principal contributions of this research include:

- A pioneering aspect of this research is the elimination of the need for golden reference data, a bottleneck in traditional detection methods. Through unsupervised machine learning, the proposed approach has broken away from the dependence on authenticated ICs, fostering adaptability to diverse integrated circuits and mitigating limitations associated with the availability of golden chips.
- This study has optimized hardware trojan detection by simultaneously employing non-destructive analysis methods, significantly reducing time and resource requirements, and minimizing reliance on destructive reverse engineering processes. Unlike conventional methods, this streamlined approach enhances efficiency and offers a robust alternative, contributing to cost reduction and streamlining the detection process.
- Achieving a notable accuracy rate of 93%, the research has surpassed existing techniques in identifying hardware Trojans. The use of unsupervised clustering has demonstrated superior false positive rates and comparable accuracy to supervised counterparts. This proficiency is particularly evident in pinpointing diminutive trojans triggered by concise events, showcasing the efficacy of the proposed methodology in detecting subtle malicious modifications.

1.2 Organization of the Paper

The subsequent sections of this document adhere to a well-defined organizational framework. Section II furnishes an encompassing overview of prior research undertakings, the established threat model, and fundamental background elucidation. Section III delves into the exposition of test chip design and the intricate architecture of the ring oscillator network. Hardware trojan implementation is comprehensively addressed in Section IV. In Section V, meticulous attention is devoted to delineating the meticulous process of data collection and overview of the proposed model. Section VI forms the crux of proposed trojan detection methodology, introducing an innovative and impactful unsupervised clustering approach. And, Section VII talks about the related clustering algorithms in details. Subsequently, Section VIII stands as a compelling demonstration platform for the experiments and the subsequent evaluation metrics. In Section IX, a comprehensive examination of results is delved into, and a comprehensive discourse is initiated, underpinned by a comparison with the previous supervised learning-centric approach.

Lastly, Section X encapsulates the work within a conclusive framework, offering an in-depth exploration of its contributions and illuminating potential avenues for future advancement.

2 BACKGROUND

2.1 Hardware Trojan

In the intricate landscape of modern semiconductor design and manufacturing, the emergence of hardware trojans has become an increasingly pressing concern. These stealthy adversaries possess the alarming ability to surreptitiously alter the functionality of integrated circuits, potentially leading to catastrophic outcomes. What sets hardware trojans apart

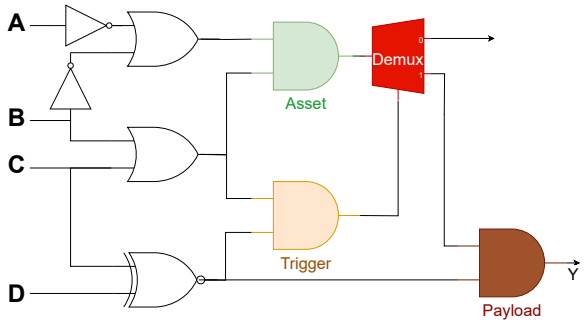


Fig. 1. Hardware Trojan leaks sensitive data through re-routing asset output to Y.

is their insidious nature, enabling them to evade detection through conventional verification and testing mechanisms. A fundamental conceptual model of a hardware trojan encompasses two critical components: The trigger and the payload [12]. The trigger acts as a clandestine initiator, activating the trojan based on specific internal circuit states or external inputs. Once invoked, the payload takes center stage, subtly manipulating the circuit's behavior to achieve the trojan's intended effects. This two-tiered structure of hardware trojans is illustrated in Figure 1. These trojans exhibit a diverse array of activation mechanisms, physical attributes, and payload characteristics, necessitating a multifaceted approach to classification and detection.

2.2 Threat Model

This research operates under the assumption that the trustworthiness of the foundry can be called into question. This implies that an adversarial actor could potentially gain unauthorized access to the integrated circuit (IC) mask layout files, thereby enabling the introduction of malicious modifications [8]. However, the scope of this study intentionally focuses on a specific subset of hardware trojans—those involving the insertion or removal of logic components. This selective approach excludes more intricate attacks such as doping-level trojans and analog circuit-based exploits, as well as non-digital forms of compromise [2, 22]. By defining the threat model this way, detection techniques are specifically developed to identify digital hardware trojans that may be illicitly integrated during the manufacturing process.

2.3 Unsupervised Machine Learning

Within the dynamic landscape of modern computing, machine learning algorithms play a pivotal role in enabling systems to learn from data and make informed decisions without explicit programming [10, 11]. Unsupervised machine learning algorithms, a subset of machine learning techniques, hold particular promise in the context of anomaly detection. Unlike supervised learning, where algorithms are trained on labeled data, unsupervised algorithms explore the inherent structure and patterns within unlabeled datasets. These algorithms excel in scenarios where large datasets require efficient organization and clustering, without the need for prior labeling or categorization of the data points. By leveraging the underlying data distribution, unsupervised machine learning methods can identify subtle deviations, making them well-suited for detecting anomalous behavior indicative of hardware trojans [17].

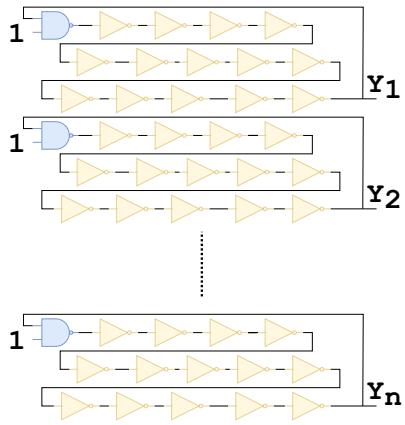


Fig. 2. Schematic of an array of n 15-stage oscillator.

2.4 Related Works

The detection of hardware trojans in pre-fabricated integrated circuit (IC) chips is currently pursued through three distinct methods. These methods encompass invasive approaches such as reverse engineering, as well as non-invasive techniques like logic testing and analysis of side channels [3].

Invasive methods, exemplified by Optical Inspection, involve the gradual removal of successive layers from the surface of the IC chip. This step-by-step deconstruction facilitates the reverse engineering of the chip's original design, as envisioned by the chip manufacturer [23]. However, this methodology is inherently destructive, rendering the tested IC chip unusable. Moreover, its scope is limited to the examination of a solitary chip, lacking the capability to offer conclusive insights into the wider array of manufactured chips. The efficacy of this approach relies heavily upon the availability of a trusted "Golden Reference" chip design, against which the tested chip can be measured for potential hardware trojan contamination [7].

Non-invasive methods, like Logic Testing, are generally more favorably received due to their non-destructive nature, allowing the chip to maintain functionality even after applying hardware trojan detection methods. Logic testing involves devising test vectors that target less frequently used pathways and gates more prone to triggering hardware trojans. However, as IC chip complexity advances, generating test vectors for logic testing becomes increasingly challenging. Despite efforts to mitigate test coverage issues, logic testing remains limited in identifying hardware trojans altering the chip's logic, as it solely detects changes in inputs and outputs [18].

Another non-invasive method for detecting hardware trojans is Side Channel Analysis (SCA). This technique collects diverse attributes generated or emitted by the functioning IC chip, including path delays, power fluctuations, leakage, temperature variations, and electromagnetic emissions. The acquired data undergoes transformation using machine learning algorithms or mathematical functions to extract meaningful insights determining the presence of a hardware trojan [4, 25].

Recent research in Side-Channel Analysis includes tactics that can be combined to achieve higher detection accuracy. The Design-for-Trust methodology, integrating trojan detection methods into the IC design, is one such approach. For instance, [15] proposed designing ICs with integrated on-board Ring Oscillator Networks (RONs) that oscillate differently based on circuit power. Monitoring the RONs' oscillating frequencies generates distinct signatures for

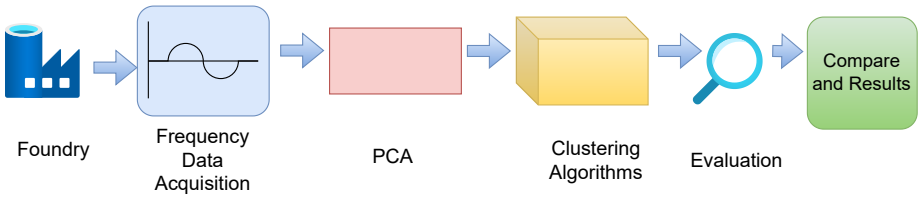


Fig. 3. Architecture of the proposed model for hardware trojan detection.

IC classification, effectively mitigating manufacturing discrepancies interfering with the observation of inherent IC characteristics [25, 28].

Machine Learning, combined with Design-for-Trust, offers an additional Side-Channel Analysis tactic. This involves creating diverse models using signatures from integrated components like Ring Oscillator Networks (RONs) as labeled training data. K. Worley et al. demonstrated this approach in [24], having achieved an impressive 94% accuracy in distinguishing hardware trojans using multiple supervised machine learning models. However, this approach relies on labeled data, necessitating Golden References, which may be costly and inefficient in an industrial application.

3 OVERVIEW OF PROPOSED MODEL

This research is aimed at using three different unsupervised clustering methods to detect trojans in integrated circuits while minimizing false positives. The approach involves collecting data for each chip using a test setup with two Trojan-free samples and 23 samples with inserted Trojans. The data is labeled and preprocessed to remove irrelevant or redundant features.

PCA is used for the feature extraction process as a dimensionality reduction technique to extract informative features from the preprocessed data. 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.

Unsupervised machine learning models are utilized for classification, including K-means, AGNES, and Birch clustering models. Each model is optimized through various techniques such as finding the best number of clusters, linkage criteria, and tuning parameters. The selected models are trained using preprocessed data and evaluated using labeled data with different sample sizes. The evaluation metrics used include accuracy, precision, recall, and F1-score. The methodology ensures the accuracy of the detection process while avoiding discarding Trojan-free ICs that could potentially be mistaken as infected. Figure 3 depicts the process flow of the unsupervised machine learning technique, using the clustering algorithm for detecting hardware trojans in semiconductor IC supply chains.

4 EXPERIMENTAL SETUP

4.1 Test Chip Design

The test chip design is meticulously tailored to facilitate a thorough evaluation of the on-chip sensor's efficacy in detecting hardware trojans within ASICs. Leveraging IBM 90nm technology, the test chip integrates the on-chip sensor structure and facilitates the controlled implementation of $N_T = 23$ pre-inserted hardware trojan designs. These trojans

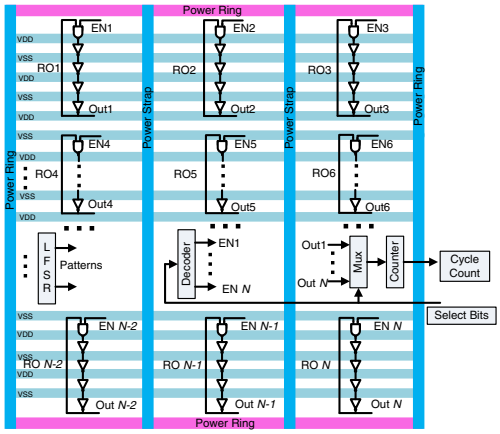


Fig. 4. Configuration of the ring oscillator network employed for trojan detection. The structure may vary based on the power network of the specific integrated circuit (IC) under test, even though N Ring Oscillators (ROs) are utilized in this particular setup [6, 15, 24].

are deliberately varied in terms of partial activity, area, and location to comprehensively assess the sensor's sensitivity to diverse malicious inclusions, enhancing the robustness of the evaluation. The layout of the test chip provides valuable insights into the spatial arrangement and distribution of critical components within the ASIC, including the on-chip sensor structure and strategically inserted hardware Trojans [9].

4.2 Ring Oscillator Network Architecture Integration

The RON architecture integrated into the test chip is meticulously designed to capitalize on the inherent sensitivity of individual ring oscillators (ROs) to power fluctuations induced by circuit switching activity and the presence of hardware trojans as shown in the Figure 4. Comprising N_{ro} = 8 and n = 61-stage ROs, each equipped with one NAND gate and 60 strategically distributed inverters, the RON architecture aims to enhance the coverage of the power distribution network and improve the sensor's sensitivity to power fluctuations induced by hardware Trojans. The deliberate distribution of ROs across different standard cell rows enables the detection of transient power consumption changes, providing a comprehensive assessment of the ASIC's integrity. The RON architecture is meticulously designed to ensure the detection of even the smallest hardware Trojans, thereby enhancing the robustness of the on-chip sensor structure.

The employed RON configuration ensures that the highest observed frequency remains within the 400MHz limit of the 90nm counters used in this design. The spacing between adjacent RO components adheres to a design rule, allowing for up to 10 times the width of the flip-flops, resulting in the utilization of eight ROs. The Linear Feedback Shift Register (LFSR) feedback polynomial utilized is $X^7 + X^3 + 1$. An 8-bit LFSR is employed to generate patterns for the 36-input s9234 benchmark, with a broadcasting technique assigning the 8-bit output to the 36 inputs. RO selection is achieved through an 8-bit decoder and multiplexer. A 16-bit counter, controlled by a timer, measures oscillations during the 500-clock-cycle test duration, considering technology and area overhead.

4.3 Hardware Trojan Implementation

Each integrated circuit (IC) incorporates seven combinational hardware trojan designs, all potentially deactivatable. In 90nm CMOS technology, static power dissipation and side-channel contributions are negligible when trojans are

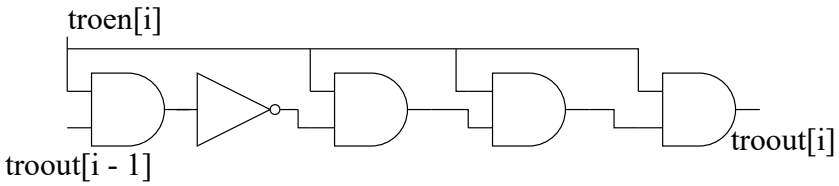


Fig. 5. Stage T_i hardware trojan.

inactive. Adopting a single-IC multiple-Trojan approach enables extensive trojan impact testing and isolates process variation effects from Trojan-induced influences on RO characteristic frequencies.

The gate-level structure of a trojan stage is illustrated in Fig. 5. troout$[i]$ represents the output of the i-th trojan stage, troout$[i-1]$ denotes the previous trojan stage output, and troen$[i]$ is the enabling signal for the i-th stage, activating prior stages as well. Trojan T_i encompasses i stages, each composed of $i \times$ (4AND + 1INV) gates. The first Trojan, T_1, is driven by a 200MHz clock signal.

5 DATA AND MATERIALS

5.1 Data Collection

Data collection for the test chip involves a meticulous measurement process aimed at capturing the frequency variations of individual ring oscillators under different conditions. The measurement setup facilitates the precise assessment of measurement noise and the impact of hardware trojans on the frequency characteristics of the on-chip oscillators. Each trial involves measuring the frequency of a single ring oscillator on each chip 10 times, with each trial lasting 500 clock cycles. This rigorous data collection process enables the calculation of measurement noise for each chip, providing valuable insights into the impact of hardware trojans on the frequency characteristics of the on-chip sensors. The comprehensive data collection process serves as a foundation for evaluating the effectiveness of the on-chip sensor in detecting fluctuations induced by the presence of hardware Trojans, thereby contributing to a thorough understanding of the sensor's performance characteristics.

5.2 Data Preparation

To prepare the collected data for analysis, a series of preprocessing steps are performed to remove irrelevant/redundant features, normalize feature values, and check for missing values/outliers using imputation and interquartile range. The resulting dataset includes "golden" or Trojan-free samples and samples with Trojans, labeled only for evaluation purposes. This dataset serves as the foundation for feature extraction and clustering analysis, which are described in more detail in the Table 1.

6 ANALYSIS METHOD USING UNSUPERVISED LEARNING

The preprocessed and feature-extracted data, which consists of both trojan-infected and trojan-free samples, are clustered using the various clustering algorithms. The clustering model is applied to the preprocessed and feature-extracted data, resulting in different clustering outcomes. The clustering model enables 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.

Table 1. Table Showing Summary of Datasets

Entities	number
Number of Chips	32
Number of ROs in each chip	8
Trojan infected instances in each chip	23
Trojan free instances in each chip	2
Total number of Instances	800

6.1 BIRCH Clustering model

The BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) algorithm is a hierarchical clustering technique that excels in efficiently processing large datasets while effectively discovering underlying patterns and structures within the data. Its design principles and mechanisms make it an appealing choice for the intricate task of hardware trojan detection, where identifying subtle variations in integrated circuits is paramount.

Algorithm 1 outlines the BIRCH clustering process, wherein data points are initially grouped based on a specified branching factor (B) and threshold (T). Subsequently, a global clustering phase merges these sub-clusters into larger clusters, forming a hierarchical structure that captures the interrelationships among data points. The algorithm may further undergo a cluster refining phase to rectify any inaccuracies introduced during the initial clustering. BIRCH relies on a distance metric, commonly the Euclidean distance defined in Equation 1, to gauge the similarity between data points and subcluster centroids. The procedural steps are visualized in Figure 6. The Euclidean distance is favored for its simplicity and effectiveness in quantifying dissimilarity between data points.

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

Parameter optimization is a critical aspect of the BIRCH algorithm, involving the selection of appropriate tuning parameters such as the branching factor and the threshold to strike a balance between capturing intricate variations indicative of hardware trojans and avoiding the inclusion of normal variations that might lead to false positives . The cluster size selection process involved the "adjust branching factor and threshold values" technique within BIRCH clustering, allowing for effective control over the granularity of clustering outcomes. Through rigorous experimentation, the optimal number of clusters consistently converged to 4 .

6.2 AGNES Clustering

The AGNES (Agglomerative Nesting) algorithm is known for its ability to create hierarchical cluster structures, providing an avenue to unveil inherent patterns within the data, which is essential for hardware trojan detection. The algorithm 2 describes how AGNES works by iteratively merging the most similar data points or clusters until a stopping criterion is met, resulting in a hierarchical clustering structure.

This process forms a hierarchy of clusters, which is visualized as a dendrogram as shown in Figure 7. The height of the dendrogram at each merging step reflects the similarity level at which clusters are combined.

Parameter optimization for AGNES clustering involves the extraction of a specific number of clusters that best capture the inherent patterns within the data. By strategically pruning the dendrogram at an appropriate height, the optimal cluster count was determined to be 4, ensuring that the resultant clusters are meaningful and representative

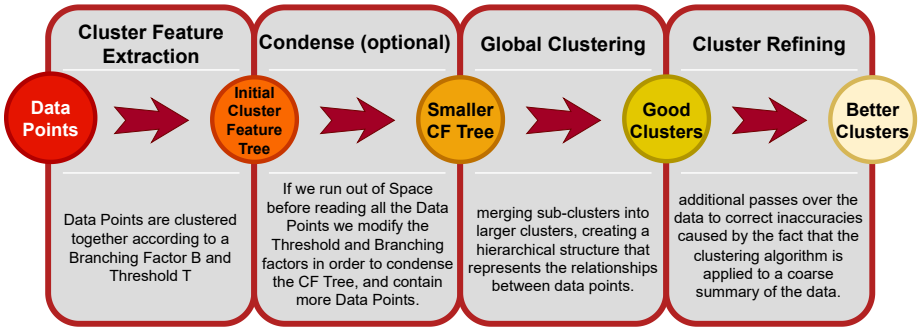


Fig. 6. Stepwise Progression of the Birch Clustering Model.

Algorithm 1: BIRCH Clustering Algorithm

Data: Dataset D, Branching factor B, Threshold T
Result: Cluster Feature Tree (CF Tree)
Initialization;
Initialize CF Tree as empty;
for *each data point in* D **do**
  if *CF Tree is empty* **then**
    Create a new subcluster with the data point as its centroid;
  end
  else
    Find the closest subcluster to the data point based on the distance metric;
    if *Closest subcluster is within distance* T **then**
      Add the data point to that subcluster and update its centroid;
    end
    else if *Number of subclusters* $< B$ **then**
      Create a new subcluster with the data point as its centroid;
    end
    else if *Adding the data point causes subcluster size to exceed threshold* **then**
      Split the subcluster into two subclusters;
    end
  end
end
Global Clustering (GC) Phase;
Merge subclusters into larger clusters to create a hierarchical structure;
Create a hierarchical clustering tree representing the relationships between data points;
Cluster Feature Tree Refining Phase (Optional);
Run additional passes over the data to correct any inaccuracies;
Parameter Optimization;
Select optimal values for B and T;

. The AGNES algorithm's performance is also evaluated across different metrics and sample sizes, demonstrating adaptability to varying sample sizes with notable performance in true positive rate (TPR) for moderate samples . The

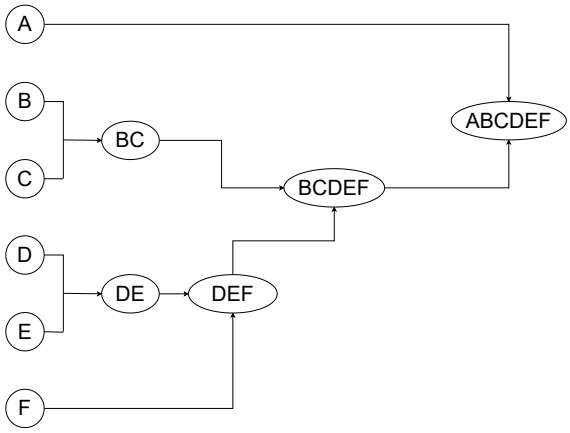


Fig. 7. An AGNES Dendogram showing the order in which the points were merged according to the Similarity between each other.

Algorithm 2: AGNES Clustering Algorithm

Data: Dataset D, Distance metric, Linkage criteria
Result: Hierarchical clustering tree
Initialization;
Initialize each data point as its own initial cluster;
Compute pairwise distances between data points;
while *Number of clusters* > 1 **do**
  Merge the two closest clusters based on the computed pairwise distances and linkage criteria;
  Update the hierarchical clustering tree;
end

Algorithm 3: K-means Clustering Algorithm

Data: Dataset D, Number of clusters k
Result: Set of cluster centroids
Initialization;
Randomly initialize k cluster centroids;
while *Centroids are changing* **do**
  Assign each data point to the nearest centroid;
  Update each centroid to be the mean of the data points assigned to it;
end

ROC curve for AGNES revealed a respectable area under the curve (AUC) of 0.94, further validating its aptitude in delineating between true and false positives.

6.3 K-Means Clustering Model

K-Means clustering offers an efficient approach to hardware trojan detection by grouping similar data points. KMeans clustering is known for its stability and balanced performance across sample sizes and configurations . The steps given in algorithm 3 describes how trojan-induced patterns are uncovered. This is accomplished by assigning the data points

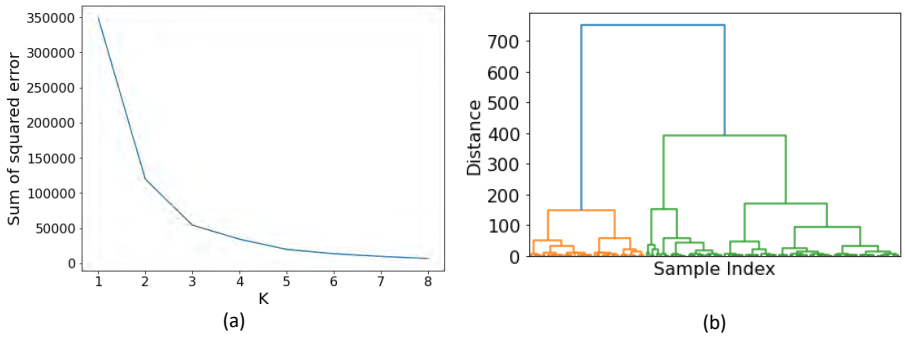


Fig. 8. Techniques for number of cluster section: (a) Elbow Plot and (b) Dendogram for hierarchical clustering.

iteratively to the nearest cluster centroid, afterwards, the centroids are recalculated based on the mean of the assigned data points.

Given a dataset $\mathbf{X}$ with n data points and $\mathbf{x}_i \in \mathbb{R}^d$, the goal of K-Means is to partition dataset $\mathbf{X}$ into k clusters $\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_k$, where each cluster is represented by a centroid $\mathbf{c}_j \in \mathbb{R}^d$. The data points $\mathbf{x}_i$ are clustered according to their proximity to the cluster centroids, where the point get clustered with the closest cluster centroid's cluster, and the proximity is calculated according to the algorithm which minimizes the objective function:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} r_{ij} \|\mathbf{x}_i - \mathbf{c}_j\|^2 \tag{2}$$

where r_{ij} is an indicator variable that equals 1 if data point $\mathbf{x}_i$ is assigned to cluster $\mathbf{C}_j$, and 0 otherwise.

Parameter optimization for KMeans clustering involves determining the ideal number of clusters through systematic analysis, with the sum of squared error (SSE) being a key metric used to quantify the variability within clusters. The formula for SSE is given by:

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{k} (x_{ij} - \mu_j)^2 \text{ where:} \tag{3}$$

n is the number of data points, k is the number of clusters, x_{ij} is the j^{th} feature of the i^{th} data point, μ_j is mean of j^{th} feature across all datapoints in the cluster.

The Elbow Method, a graphical technique, involves plotting the SSE values for different values of k and identifying the "elbow point" as shown in Figure 8. (a). This point represents the optimal cluster count where further increasing clusters provides diminishing returns in reducing SSE. Remarkably, in our analysis, the Elbow Method consistently points to an optimal cluster count of 4. This cross-validated result reinforces the significance of this cluster count in revealing meaningful patterns within the data.

7 EVALUATION METHOD

The evaluation employs three distinct sample sizes—6 chips, 12 chips, and 24 chips—where each experimental trial involves training and testing machine learning models on datasets corresponding to these chip quantities. This variation in sample sizes allows for an investigation into how the performance of unsupervised machine learning algorithms

is influenced by the quantity of data available for training and testing. This assessment provides insights into the scalability and robustness of the models across varying sample sizes, offering valuable implications for the practical applicability of the detection methods in real-world scenarios with diverse data availability. Moreover, this approach facilitates a direct comparison of results with the supervised learning techniques introduced by [24], employing the same evaluation method and dataset. Each sample size underwent 20 trials, and metrics including average accuracy, false positive rate (FPR), false negative rate (FNR), true negative rate (TNR), and true positive rate (TPR) were calculated and recorded. Additional metrics such as Area under the Curve (AUC), F1 score, and G-mean were also considered. Repeating the experiments multiple times and calculating average results enhanced the reliability and accuracy of the assessment for each classifier [20], [19], [16].

8 RESULTS AND DISCUSSIONS

In the comprehensive experimental analysis, the performance of three distinct clustering algorithms: BIRCH, AGNES, and KMeans, is systematically investigated, each subjected to rigorous scrutiny across varying configurations and sample sizes. BIRCH clustering not only demonstrates high accuracy and runtime efficiency in clustering the dataset, but it also produces informative visualizations for a better understanding of the clustering results. The scatter plot presented in Fig. 9 illustrates how the data points are distributed among the clusters formed using the various clustering algorithm for a sample size of 6. It is apparent from the plot that the clusters are distinctly separated. The labels in legends are given as per the evaluation of the acquired results. In all three models, second cluster from the left is identified as trojan free and others are found to be trojan infected data.

8.1 Analysis of Varied Model Performances Across Different Sample Sizes

Starting with the BIRCH algorithm, a meticulous evaluation unveils its remarkable capacity to accurately cluster the dataset. The default configuration yieldes an impressive clustering accuracy of 92.5%. Fine-tuning the branching factor parameter to 100 resultes in a marginal accuracy improvement to 93.1% for 24 number of samples, though at the expense of increased runtime. Notably, a further increment in the branching factor leads to escalating runtime without commensurate accuracy gains. The algorithm exhibits notable scalability, efficiently handling substantial datasets even with relatively modest sample sizes, affirming its efficacy for large-scale data clustering. Moreover, for the 6 sample sized experiment the receiver operating characteristic (ROC) curve underscores the algorithm's robustness, featuring an area under the curve (AUC) value of 0.95, a testament to its adeptness in distinguishing true and false positives. The results is illustrated in the Table 2.

Shifting focus to the AGNES algorithm, the findings highlight its nuanced performance trends which is showcased in Table 3. With varying sample sizes, the false negative rate (FNR) demonstrates fluctuations, ranging from 0.096 to 0.128. The true negative rate (TNR) exhibits a similar oscillating pattern, with values of 1.0, 0.89, and 1.0 for sample sizes 6, 12, and 24, respectively. While TPR decreases slightly from 0.903 to 0.87, FPR shows variations across sample sizes. The accuracy trend exhibits a modest decline from 0.91 to 0.88. Notably, the F1 Score showcases consistent improvement from 0.95 to 0.931. The AGNES algorithm's ROC curve reveals a respectable AUC of 0.94, further validating its aptitude in delineating between true and false positives.

Furthermore, the KMeans algorithm exhibits robustness across sample sizes and configurations as illustrated in Table 4. Noteworthy observations include minimal variations in FNR, ranging from 0.064 to 0.12 to 0.093 for sample sizes 6, 12, and 24, respectively. TNR demonstrates an ascending trajectory, ranging from 0.90 to 1.0 to 0.934. TPR shows slight variability, with values of 0.935, 0.879, and 0.906, and FPR experiences fluctuations. Accuracy displays subtle changes,

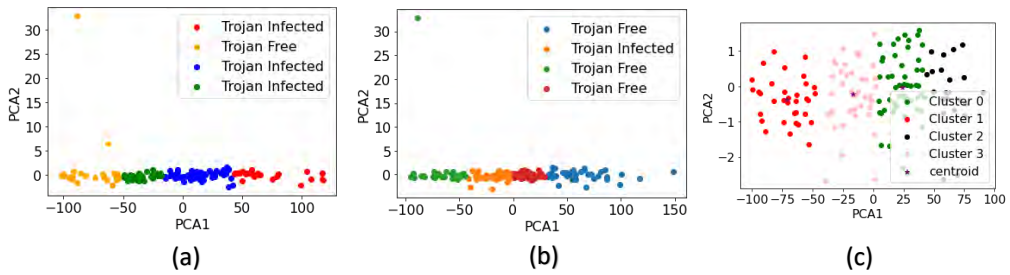


Fig. 9. Scatter plot depicting the distribution of clusters for a sample size of 6: (a) BIRCH model (b) AGNES model and (c) Kmeans model.

Table 2. 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

Table 3. Table showing AGNES Clustering Results

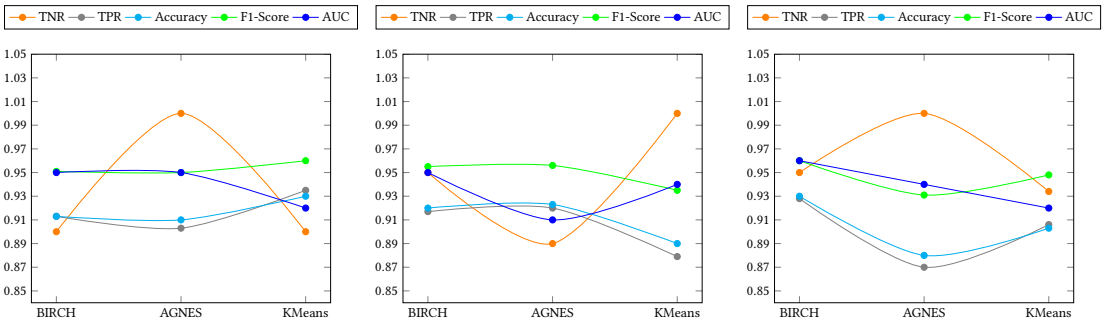
Metric	Sample Size		
	6	12	24
FNR	0.096	0.073	0.128
TNR	1.0	0.89	1.0
TPR	0.903	0.92	0.87
FPR	0.0	0.107	0.0
Accuracy	0.91	0.923	0.88
F1 Score	0.95	0.956	0.931
AUC	0.95	0.91	0.94

ranging from 0.93 to 0.89 to 0.903. F1 Score exhibits a consistent positive trend, ascending from 0.96 to 0.935 to 0.948. The KMeans algorithm's ROC curve features an AUC of 0.93, affirming its prowess in discerning between true and false positives.

In culmination, the comprehensive and methodical experimentation unveils a tapestry of unique characteristics inherent to each clustering algorithm. The triad of BIRCH, AGNES, and KMeans are meticulously evaluated, with their distinct attributes and performance nuances laid bare through a symphony of rigorous analyses and interpretive

Table 4. Table showing KMeans Clustering Results

Metric	Sample Size		
	6	12	24
FNR	0.064	0.12	0.093
TNR	0.90	1.0	0.934
TPR	0.935	0.879	0.906
FPR	0.09	0.0	0.065
Accuracy	0.93	0.89	0.903
F1 Score	0.96	0.935	0.948
AUC	0.92	0.94	0.92



(a) Sample Size 6 (b) Sample Size 12 (c) Sample Size 24

Fig. 10. Comparative Analysis of Clustering Algorithm Performance for different sample sizes.

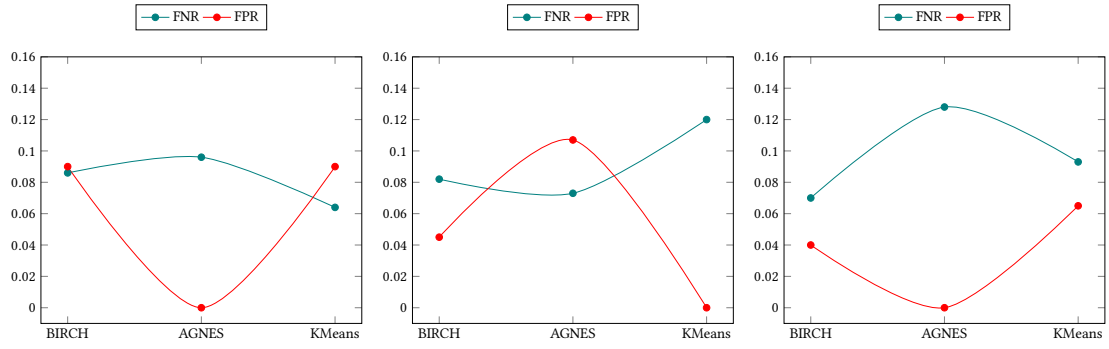
insights. Figures 10 and 11, a visual symposium of data-driven narratives, enriches this summation by providing a line graph representation of their comparisons.

Each line graph (10a, 11a, 10b, 11b, 10c, 11c) corresponds to a specific sample size (6, 12, and 24 respectively), providing a vivid portrait of how the algorithms perform under different conditions. The line graphs unveils how BIRCH, AGNES, and KMeans perform across different metrics and sample sizes, as Figure 10 shows, as the metrics are higher the corresponding algorithm offers more accurate results, and as Figure 11 shows, as the metrics go down, the Algorithms output inaccuracies less.

BIRCH consistently excels, maintaining high accuracy, F1 Score, and AUC, as shown in Figure 10. AGNES adapts well to varying sample sizes, with notable performance in TPR for moderate samples, as shown in Figure 10b, but faces challenges with larger datasets, as the FNR and FPR shows in Figure 11c. KMeans remains stable and balanced, showcasing steady F1 Score and TNR as shown in Figure 10. Overall, BIRCH stands out for its robust performance, while AGNES demonstrates adaptability and KMeans maintains stability.

8.2 Comparison with existing work

This section delves into a comprehensive analysis that juxtaposes the performance of the proposed unsupervised clustering techniques against existing supervised models, illuminating both the results and the broader implications for real-world applications. The presented findings, outlined in Table 5, provides a multifaceted perspective on the efficacy



(a) Sample Size 6 (b) Sample Size 12 (c) Sample Size 24

Fig. 11. Comparative Analysis of False Negative Rate and False Positive Rate for Clustering Algorithm Performance for different sample sizes.

of different models in the context of the trojan detection problem. The accuracy and G-mean scores stand as reliable metrics to assess the models' classification prowess. As the sample sizes of 6, 12, and 24 are traversed, intricate trends begin to emerge.

Starting with the unsupervised BIRCH Cluster, its accuracy demonstrates consistent growth as the sample size expands. Specifically, accuracy values of 0.913, 0.92, and 0.93 for sample sizes 6, 12, and 24, respectively, underscore its remarkable ability to adapt and improve its trojan detection accuracy. Likewise, the G-mean scores of 0.906, 0.933, and 0.94 for the corresponding sample sizes mirror this positive trend. The AGNES Cluster, while slightly varying in performance, also showcases its capability to contend with trojan detection. Notably, accuracy values of 0.91, 0.923, and 0.88 for sample sizes 6, 12, and 24, respectively, provide insights into its potential. This is mirrored in the G-mean scores of 0.95, 0.909, and 0.933 for the same sample sizes. The KMeans Cluster, although displaying a dip in accuracy for the largest sample size, maintains a competitive stance with accuracy values of 0.926, 0.92, and 0.9 for sample sizes 6, 12, and 24. G-mean scores of 0.922, 0.937, and 0.92 indicate its consistent capability in trojan detection across varying sample sizes.

Contrasting the unsupervised techniques, the ensemble models of SVM, KNN, and NB, as well as SVM and NB, and KNN and NB, provide valuable benchmarks. These supervised models, while demonstrating competitive performance, often fall short of the unsupervised techniques in terms of both accuracy and G-mean, especially when faced with smaller sample sizes.

Importantly, these results hold significant implications for real-world trojan detection scenarios. The challenge of acquiring accurate and comprehensive golden reference data, especially for rare or evolving trojans, is a pervasive concern. Here, the unsupervised techniques, by showcasing remarkable performance across sample sizes, effectively address the dearth of labeled data. This underscores the potential of leveraging unsupervised clustering techniques as potent tools for trojan detection, particularly in scenarios where labeled data collection is prohibitively expensive, time-consuming, or impractical.

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. In essence, the remarkable performance of the proposed unsupervised clustering techniques signifies a paradigm shift in trojan detection methodologies. By harnessing the inherent patterns within the data, these techniques demonstrate the capacity to rival or even surpass existing

Table 5. 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
AGNES Cluster	0.91	0.923	0.88	0.95	0.909	0.933
KMeans Cluster	0.926	0.92	0.9	0.922	0.937	0.92
Ensemble - SVM + KNN + NB [24]	0.922	0.92	0.94	0.85	0.86	0.926
Ensemble – SVM + NB [24]	0.88	0.879	0.88	0.90	0.91	0.933
Ensemble – KNN + NB [24]	0.886	0.883	0.873	0.92	0.93	0.92

supervised models, effectively circumventing the reliance on extensive labeled data and offering robust solutions for real-world trojan detection challenges.

9 CONCLUSION

In addressing the persistent threat of malicious hardware trojan insertions in supply chain attacks, this research introduces an innovative hardware security approach leveraging unsupervised machine learning and side-channel analysis. The results demonstrate the effectiveness and resilience of this method, offering a valuable addition to the hardware security toolkit. Beyond practical application, the transformative potential of our approach is underscored, suggesting a paradigm shift in semiconductor integrated circuits (ICs) security. By contributing to the ongoing discourse, this research plays a substantive role in advancing secure and trustworthy semiconductor ICs within the intricate landscape of digital systems.

ACKNOWLEDGMENT

This research is 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.

REFERENCES

[1] Maitreyi Ashok, Matthew J Turner, Ronald L Walsworth, Edlyn V Levine, and Anantha P Chandrakasan. 2022. Hardware trojan detection using unsupervised deep learning on quantum diamond microscope magnetic field images. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 18, 4 (2022), 1–25.

[2] Georg T Becker, Francesco Regazzoni, Christof Paar, and Wayne P Burleson. 2014. Stealthy dopant-level hardware trojans: extended version. *Journal of Cryptographic Engineering* 4 (2014), 19–31.

[3] Shivam Bhasin and Francesco Regazzoni. 2015. A survey on hardware trojan detection techniques. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2021–2024.

[4] Xiaotong Cui, Elnaz Koopahi, Kaijie Wu, and Ramesh Karri. 2018. Hardware Trojan Detection Using the Order of Path Delay. *J. Emerg. Technol. Comput. Syst.* 14, 3, Article 33 (oct 2018), 23 pages. https://doi.org/10.1145/3229050

[5] Sina Faezi, Rozhin Yasaei, Anomadarshi Barua, and Mohammad Abdullah Al Faruque. 2021. Brain-inspired golden chip free hardware trojan detection. *IEEE Transactions on Information Forensics and Security* 16 (2021), 2697–2708.

[6] Andrew Ferraiuolo, Xuehui Zhang, and Mohammad Tehranipoor. 2012. 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*. 37–42.

[7] Julien Francq and Florian Frick. 2015. Introduction to hardware Trojan detection methods. In *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 770–775.

[8] Ashutosh Ghimire, Mahommed Alkurdi, and Fathi Amsaad. 2024. Enhancing Hardware Trojan Security through Reference-Free Clustering using Representatives. In *2024 37th International Conference on VLSI Design and 2024 23rd International Conference on Embedded Systems (VLSID)*. 467–473. https://doi.org/10.1109/VLSID60093.2024.00084

[9] Ashutosh Ghimire, Fathi Amsaad, Tamzidul Hoque, Kenneth Hopkinson, and Md Tauhidur Rahman. 2023. Unsupervised IC Security with Machine Learning for Trojan Detection. In *2023 IEEE 66th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 20–24.

[10] Ashutosh Ghimire, Ahmad Nasser Asiri, Brian Hildebrand, and Fathi Amsaad. 2023. Implementation of Secure and Privacy-aware AI Hardware using Distributed Federated Learning. In *2023 IEEE 16th Dallas Circuits and Systems Conference (DCAS)*. IEEE, 1–6.

[11] Ashutosh Ghimire, Vishnu Vardhan Baligodugula, and Fathi Amsaad. 2023. 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, 357–367.

[12] Jason R Hamlet, Jackson R Mayo, and Vivian G Kammler. 2019. Targeted modification of hardware trojans. *Journal of Hardware and Systems Security* 3 (2019), 189–197.

[13] Ayush Jain, Ziqi Zhou, and Ujjwal Guin. 2021. Survey of recent developments for hardware trojan detection. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–5.

[14] Nima Karimian, Fatemeh Tehranipoor, Md Tauhidur Rahman, Shane Kelly, and Domenic Forte. 2015. Genetic algorithm for hardware Trojan detection with ring oscillator network (RON). In *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*. IEEE, 1–6.

[15] Shane Kelly, Xuehui Zhang, Mohammed Tehranipoor, and Andrew Ferraiuolo. 2015. Detecting hardware trojans using on-chip sensors in an asic design. *Journal of electronic testing* 31 (2015), 11–26.

[16] Vincent Labatut and Hocine Cherifi. 2011. Evaluation of performance measures for classifiers comparison. *arXiv preprint arXiv:1112.4133* (2011).

[17] Boris Lorbeer, Ana Kosareva, Bersant Deva, Dženan Softić, Peter Ruppel, and Axel Küpper. 2018. Variations on the clustering algorithm BIRCH. *Big data research* 11 (2018), 44–53.

[18] Anindan Mondal, Debasish Kalita, Archisman Ghosh, Suchismita Roy, and Bibhash Sen. 2023. Toward the Generation of Test Vectors for the Detection of Hardware Trojan Targeting Effective Switching Activity. *J. Emerg. Technol. Comput. Syst.* 19, 4, Article 29 (sep 2023), 16 pages. https://doi.org/10.1145/3597497

[19] Sarang Narkhede. 2018. Understanding auc-roc curve. *Towards Data Science* 26, 1 (2018), 220–227.

[20] Jake Olivier, William D Johnson, and Gailen D Marshall. 2008. The logarithmic transformation and the geometric mean in reporting experimental IgE results: what are they and when and why to use them? *Annals of Allergy, Asthma & Immunology* 100, 4 (2008), 333–337.

[21] Mohammad Ashiqur Rahman, Md Tauhidur Rahman, Mithat Kisacikoglu, and Kemal Akkaya. 2020. Intrusion detection systems-enabled power electronics for unmanned aerial vehicles. In *2020 IEEE CyberPELS (CyberPELS)*. IEEE, 1–5.

[22] Saswat Kumar Ram, Sauvagya Ranjan Sahoo, Banee Bandana Das, Kamalakanta Mahapatra, and Saraju P. Mohanty. 2023. Eternal-thing 2.0: Analog-Trojan-resilient Ripple-less Solar Harvesting System for Sustainable IoT. *J. Emerg. Technol. Comput. Syst.* 19, 2, Article 12 (mar 2023), 25 pages. https://doi.org/10.1145/3575800

[23] Hassan Salmani. 2016. COTD: Reference-free hardware trojan detection and recovery based on controllability and observability in gate-level netlist. *IEEE Transactions on Information Forensics and Security* 12, 2 (2016), 338–350.

[24] Kyle Worley and Md Tauhidur Rahman. 2019. Supervised machine learning techniques for trojan detection with ring oscillator network. In *2019 SoutheastCon*. IEEE, 1–7.

[25] Kan Xiao, Domenic Forte, Yier Jin, Ramesh Karri, Swarup Bhunia, and Mohammad Tehranipoor. 2016. Hardware trojans: Lessons learned after one decade of research. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 22, 1 (2016), 1–23.

[26] Mingfu Xue, Jian Wang, and Aiqun Hu. 2016. An enhanced classification-based golden chips-free hardware Trojan detection technique. In *2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST)*. IEEE, 1–6.

[27] Shuo Yang, Prabuddha Chakraborty, Patanjali SLPSK, and Swarup Bhunia. 2021. Trusted electronic systems with untrusted cots. In *2021 22nd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 198–203.

[28] Xuehui Zhang, Andrew Ferraiuolo, and Mohammad Tehranipoor. 2013. Detection of trojans using a combined ring oscillator network and off-chip transient power analysis. 9, 3, Article 25 (oct 2013), 20 pages. https://doi.org/10.1145/2491677