TOWARDS UNCERTAINTY-AWARE HARDWARE TROJAN DETECTION

A THESIS

Presented to the Department of Computer Engineering and Computer Science

California State University, Long Beach

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

Committee Members:

Amin Rezaei, Ph.D. (Chair)
Ava Hedayatipour, Ph.D.
Arash Sarshar, Ph.D.

College Designee:

Praveen Shankar, Ph.D.

By Rahul Deo Vishwakarma

B.Tech., 2009, SRM Institute of Science and Technology, India

May 2024

ABSTRACT

As the semiconductor industry moves towards a fabless paradigm, the risk of hardware Trojans being inserted at various production stages has increased. Recently, there has been a trend towards using machine learning solutions to detect hardware Trojans more effectively, with a focus on model accuracy as an evaluation metric. However, in a high-risk and sensitive domain, even a small misclassification is unacceptable. Additionally, expecting an ideal model, especially when Trojans evolve over time, is unrealistic. Thus, there is a need for metrics to assess the reliability of detected Trojans and a mechanism to simulate unseen ones.

In this thesis, we generate evolving hardware Trojans using conformalized generative adversarial networks and offer an approach to detecting them based on a non-intrusive statistical inference framework, leveraging the Mondrian conformal predictor. This approach acts as a wrapper over any machine learning model, providing predictions accompanied by uncertainty quantification for each identified Trojan, facilitating more resilient decision-making. In cases where a NULL set emerges, indicative of instances where the prediction set is empty, we discuss an approach to reject the decision while providing explainability.

Moreover, while the focus has been on statistical or deep learning approaches, the limited number of Trojan-infected benchmarks affects detection accuracy and hampers the ability to detect zero-day Trojans. To mitigate this shortfall, we employ generative adversarial networks to augment our data in two alternative representation modalities: graph and tabular, ensuring a representative dataset with different modalities. Additionally, we propose a multimodal deep learning methodology for hardware Trojan detection and assess outcomes from both early and late fusion strategies. We also evaluate the uncertainty quantification metrics of each prediction to facilitate risk-aware decision-making. The findings affirm the efficacy of our proposed hardware Trojan detection technique and pave the way for future research in multi-modality and uncertainty quantification to address broader hardware security concerns.

The practical application of the proposed approach lies in enhancing hardware Trojan detection

and evaluation through an uncertainty-aware approach within the semiconductor industry. Validation of the approach on synthetic and real chip-level benchmarks underscores its effectiveness and opens avenues for future investigations in multi-modality and uncertainty quantification to address broader hardware security concerns in the semiconductor domain.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation and gratitude to Dr. Amin Rezaei, my esteemed thesis advisor, for his invaluable guidance and unwavering support throughout the research journey. Dr. Rezaei played a pivotal role in shaping my research experience, and his patience and dedication have been instrumental in bringing this thesis to fruition.

I also express my heartfelt gratitude to Dr. Ava Hedayatipour and Dr. Arash Sarshar for their invaluable input and for graciously serving on my thesis defense committee. Their expertise and constructive feedback have significantly contributed to the quality of this work.

To my dear friend, especially Guō Yīmò, whose mentorship, understanding, encouragement, and camaraderie has played a significant role in my ability to persevere and succeed.

Lastly, I am eternally indebted to my loving parents. You have instilled in me the values of hard work and perseverance, teaching me that our determination and courage shape our destiny. Your guidance and support have been the foundation of the person I am today, and I aspire to make you proud.

This material is based upon work supported by the National Science Foundation under Grant No. 2245247. Chapter 2 is based on the paper entitled ”Risk-Aware and Explainable Framework for Ensuring Guaranteed Coverage in Evolving Hardware Trojan Detection” which has been published at the *Proceedings of 42nd International Conference on Computer Aided Design (ICCAD)*. Chapter 3 is based on the paper entitled ”Uncertainty-Aware Hardware Trojan Detection Using Multimodal Deep Learning” which has been published at the *Proceedings of 27th Design, Automation & Test in Europe Conference & Exhibition (DATE)*.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

> Security wins many battles but loses the security war. We are definitely going backwards in computer security.
>
> — Adi Shamir, A.M. Turing Award Laureate

Hardware Trojan (HT) insertion involves a malicious alteration to a hardware component's design, posing risks such as device malfunction, data leakage, or physical damage [1]. With the semiconductor industry adopting a fabless model, the potential for HT insertion at various manufacturing stages grows, posing a substantial security threat. Traditional detection methods, like signature-based approaches [2], prove ineffective against evolving HT attacks, prompting a shift towards machine learning (ML) solutions. However, existing ML approaches often lack information on datasets, struggle with concept drift, and require additional evaluation metrics [3]. A recent study in [4] questions the universality of ML in addressing hardware security concerns [5].

ML methods face challenges in production due to the evolving nature of a real-time dataset characterized by the concept drift caused by intelligent modifications to HT insertion techniques. The thesis introduces PALETTE, an algorithm-agnostic framework for detecting the evolving hardware Trojan in a circuit, utilizing conformal prediction [6]. This provides a theoretical guaranteed for each of the predictions made for the new data points [7]. Non-invasive and implementable as a wrapper over existing ML models, PALETTE offers set predictions, ensuring the correct class is included 90% of the time on average (i.e., $\alpha = 0.1$).

HT insertion is a concern in fabless semiconductor manufacturing, with potential attacks at different stages [8–11]. Vulnerabilities span design phases, EDA processes, and post-production stages, necessitating robust security measures [12–16]. Comprehensive approaches, while crucial, come with drawbacks, leading to the relevance of ML in countering HTs. Challenges like

resource-intensive training, adversarial attacks, and interpretability are acknowledged.

Recently, machine learning has surfaced as a formidable tool for HT detection in fabless semiconductor manufacturing [17–21]. Challenges include acquiring diverse datasets, susceptibility to adversarial attacks [22], and the need for interpretability and explainability [23, 24]. NOODLE, proposed in this thesis, is an uncertainty-aware hardware Trojan detection using multimodal deep learning with graph (AST) representation and tabular data, performing binary classification.

Research Gap

The shared identification of these gaps in research sets the stage for a thorough exploration plan, underscoring the particular areas that demand deeper investigation and focused development in the field of multimodal learning for detecting hardware trojans. A few of them are discussed below:

Dataset Transparency and Class Distribution: The lack of transparency in revealing comprehensive dataset details, especially concerning class distribution differences, highlights a research gap. There is a need for scholarly attention to address transparency issues and gain a nuanced understanding of dataset characteristics in the presence of significant class distribution variations.

Concept Drift in Model Evaluation: The insufficiency in considering "concept drift" stemming from adversaries' evolving insertion techniques introduces a research gap. Coping strategies are required to effectively handle concept drift, indicating the necessity for further exploration and validation in this area.

Custom metrics for Model Evaluation: The inadequacy of conventional metrics for model evaluation underscores a research gap. There is a need for additional measures to strengthen traditional evaluation methods, ensuring reliable decision-making and comprehensive prediction coverage.

Uncertainty-Aware Multimodal Learning: The limited exploration of diverse multimodal approaches, particularly beyond prevalent graph representation and tabular data, is identified

as a research gap. Scholarly inquiry and investigation into various modality combinations in multimodal learning for hardware trojan detection are needed.

Interpretability and Explainability: The acknowledgment of the significance of interpretability and explainability in multimodal models highlights a research gap. The development of interpretable multimodal hardware trojan detection models is necessary to enhance trust and understanding in the domain.

Preliminaries

Calibrated Prediction

Calibration in predictive modeling is a fundamental process aimed at ensuring the alignment between a model's confidence scores and the actual probabilities of correctness in its predictions [25].

Let X denote the input data and Y the corresponding output label. Within a training dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, the objective is to derive a function f that can effectively predict the correct output label y for a given input x. The model's output for a specific input x is represented by $f(x)$, while the actual probability of prediction correctness is denoted as $P(y = 1|x)$.

A calibrated model is characterized by its ability to produce a confidence score $g(x)$ that accurately reflects the true probability of correctness for each prediction. The central tenet of calibration lies in ensuring the alignment of the confidence score $g(x)$ with the actual probabilities, as described by the condition $P(y = 1|g(x) = p) = p$ for all p within the interval $[0, 1]$. To formalize this, let's define the calibration error for a set of predictions. For a given set of m predictions $\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, where y_i is the true label and $g(x_i)$ is the confidence score produced by the model for input x_i, the calibration error is computed as the mean squared difference between the confidence scores and the true probabilities and the reduction of this error constitutes the fundamental objective of calibration techniques.:

$$\text{Calibration Error} = \frac{1}{m} \sum_{i=1}^{m} (g(x_i) - P(y_i = 1|x_i))^2$$

Why We Need Calibration?

In the context of HT detection, calibration plays a pivotal role in evaluating the likelihood of Trojan presence within a circuit, thereby guiding crucial decision-making processes. A properly calibrated model offers reliable confidence scores, aiding in discerning situations where the circuit is unlikely to harbor Trojans despite high confidence scores. Conversely, low confidence scores coupled with a high likelihood of Trojan presence necessitate thorough scrutiny and potential mitigation measures.

By rigorously calibrating predictive models, researchers and practitioners can enhance the reliability and interpretability of predictions, thereby fortifying decision-making frameworks in critical domains like HT detection.

Conformal Prediction

Conformal prediction, introduced by Shafer and Vovk [6], is a machine learning paradigm designed to quantify prediction uncertainty through the generation of prediction sets. This framework serves to augment the inference capabilities of traditional models, ensuring robust validity and facilitating the estimation of confidence levels for individual predictions. Within the domain of HT detection, the notion of label-conditional validity assumes importance, particularly when confronted with imbalanced datasets characterized by disparities in label proportions. This relevance is accentuated by the inherent rarity of encountering a Trojan within a circuit. Notably, the absence of label-conditional validity tends to disproportionately impact minority classes, exacerbating the potential for biases in predictions [26]. Nonetheless, the mitigation of such biases can be achieved through the establishment of label-conditional validity, which guarantees that the error rate, even for minority classes, will ultimately converge to the designated significance level over time.

In certain instances, conformal prediction may yield uncertain predictions, indicated by prediction sets containing multiple values. This scenario arises when none of the labels can be confidently rejected at the specified significance level.

In the application of conformal prediction, the conventional confusion matrix undergoes

modification due to the characteristic of *prediction sets*, which encompass multiple values rather than a singular prediction. In binary classification scenarios, it becomes imperative to account for the number of accurately predicted instances, wherein the prediction set exclusively contains the correct label, alongside the tally of inaccurately predicted instances, where the prediction set encompasses solely the incorrect label. Also, due consideration must be given to instances of inconclusive predictions, manifesting when the prediction set encompasses both labels, as well as instances characterized by an *empty prediction set*. There are circumstances wherein furnishing a *single value point prediction* supersedes the provision of a prediction set or interval in a hedged forecast. In such scenarios, opting for the label associated with the highest p-value represents a straightforward and judicious choice. This point prediction can be hedged by incorporating supplementary information elucidating the underlying uncertainty.

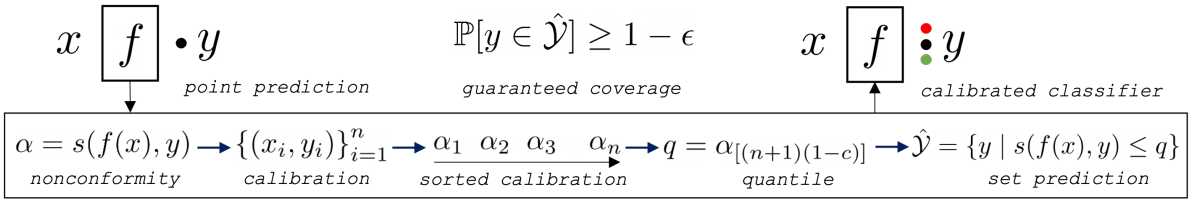


FIGURE 1. Demonstration of the conformal prediction framework.

In this thesis our work relies on Mondrian Inductive Conformal Prediction (ICP) [27] as shown in Algorithm 3. Furthermore, to decrease the rate of false negatives in alert systems, we require class-based authenticity for samples classified as "Evolving Trojan." Additionally, we must ensure that the samples labeled as "Evolving Trojan" are indeed genuine to attain this goal.

To ensure the integrity of the non-conformity scores computation, we exclusively account for the scores pertaining to instances sharing the identical class as the tested object x_{n+1}. This approach is delineated as

$p_{n+1}^{C_k} = \frac{\left|\left\{i \in 1,\ldots,q : y_i = C_k, \alpha_{n+1}^{C_k} \leq \alpha_i\right\}\right|}{|\{i \in 1,\ldots,q : y_i = C_k\}|}$ Here, $p_{n+1}^{C_k}$ represents the non-conformity score for the class C_k, computed by considering the proportion of instances with the class label C_k that possess a non-conformity score lower than or equal to the non-conformity score of the object x_{n+1}. This

Algorithm 1: Mondrian ICP

Input : Training data D, test instance x, significance level α, number of trees T, and maximum tree depth d.

Output Prediction set $C(x)$ for x.

:

Divide D into T disjoint subsets $D_1, \ldots, D_T$;

for $t \leftarrow 1$ **to** T **do**

  Sample D'_t from D_t by recursively partitioning D_t along randomly chosen hyperplanes until each partition contains at most $2d$ points.

  Train a classification model M_t on D'_t.

Compute the conformity scores $s_t(x)$ of x with respect to each model M_t.

Sort the conformity scores $s_t(x)$ in decreasing order.

Compute the p-values p_t of the T conformity scores $s_t(x)$ using the formula $p_t = \frac{T-t+1}{T}$.

Compute the threshold h such that $h = s_t(x)$ if $p_t > \alpha$, otherwise $h = \infty$.

Construct the prediction set $C(x)$ as the set of all labels y such that $s_t(y) \geq h$ for all models M_t.

return $C(x)$

computation ensures a focused assessment of conformity within the context of the specific class under consideration.

Guaranteed Coverage of Prediction

In the domain of HT detection, it is not only important to have a high level of confidence in the predictions made by a model but also a guarantee of the coverage of each prediction. The property of guaranteed coverage is an inherent property of conformal prediction, which provides statistical guarantees of the correctness of the model's predictions [28]. The theoretical guarantee of coverage is based on the significance level, which is the probability of the model making a mistake. For example, if we set the significance level to 0.05, it means that we allow the model to make mistakes 5% of the time.

The theoretical guarantee of coverage is valid for any input x, that the true output label y will be contained in the prediction set $C(x)$ with a probability of at least $1 - \alpha$, where α is the significance level. Mathematically, this can be expressed as:

$$P(y \in C(x)) \geq 1 - \alpha$$

In other words, the probability of making a mistake is bounded by α, and as α decreases, the size of the prediction set decreases, leading to higher confidence in the model's predictions. Similarly, if the value of α increases, consequently the size of the prediction set also increases; however, this also reduces the significance level (confidence) of the predictions.

For example, if we set $\alpha = 0.05$, it means that we are 95% confident that the true output label y is contained in the prediction set $C(x)$ for any input x. The use of conformal prediction provides a strong theoretical guarantee of the correctness of the model's predictions in the context of HT detection, and the corresponding proof is given in Theorem 2.

Theorem 1. *Let $\mathcal{D}$ be a probability distribution over a set $\mathcal{X} \times \{0, 1\}$, where $\mathcal{X}$ is a set of input features and $\{0, 1\}$ is the set of labels. Let $f : \mathcal{X} \to \{0, 1\}$ be a binary classifier, and let $\epsilon \in (0, 1)$ be a confidence level. Then, the conformal prediction algorithm outputs a set of predictions $C(x) \subseteq \{0, 1\}$ for each input $x \in \mathcal{X}$ such that:*

$$\mathbb{P}[(x, y) \sim \mathcal{D}, y \in C(x)] \geq 1 - \epsilon$$

where $(x, y) \sim \mathcal{D}$ denotes sampling a pair (x, y) from the distribution $\mathcal{D}$.

Proof: The proof follows from the construction of the conformal prediction algorithm. Given an input x, the algorithm outputs a set of predictions $C(x)$ based on the observed labels of the training examples with similar input features to x. The algorithm guarantees that each prediction in $C(x)$ has a p-value less than or equal to ϵ for any new input with the same feature vector as x. Since the algorithm outputs a set of predictions, the probability that at least one of the predictions is correct is at least $1 - \epsilon$.

Corollary 1. *Let $\mathcal{D}$, f, and ϵ be as in Theorem 2. For any sample size n, the conformal prediction*

algorithm outputs a set of predictions $C(x_1), \ldots, C(x_n)$ *for each input* $x_1, \ldots, x_n \in \mathcal{X}$ *such that:*

$$\mathbb{P}[\forall i \in \{1, \ldots, n\}, (x_i, y_i) \sim \mathcal{D}, y_i \in C(x_i)] \geq 1 - \epsilon$$

where $(x_i, y_i) \sim \mathcal{D}$ *denotes sampling a pair* (x_i, y_i) *from the distribution* $\mathcal{D}$ *for each* i*.*

Proof: The proof follows from a union bound over the n samples:

$$\begin{aligned}
&\mathbb{P}[\forall i \in \{1, \ldots, n\}, (x_i, y_i) \sim \mathcal{D}, y_i \in C(x_i)] \\
&\geq 1 - \sum_{i=1}^{n} \mathbb{P}[(x_i, y_i) \sim \mathcal{D}, y_i \notin C(x_i)] \\
&\geq 1 - n\epsilon
\end{aligned}$$

where the second inequality follows from Theorem 2.

Ensuring Guaranteed Prediction Coverage

In the domain of HT detection, the confidence level of model predictions is very important, as is the assurance of encompassing all potential outcomes for the decision making. The concept of guaranteed coverage in conformal prediction offers statistical guarantee for the accuracy of model predictions [28]. The theoretical underpinning of coverage guarantee hinges on the significance level, denoting the permissible probability of model errors. For instance, setting the significance level to 0.05 implies tolerance for errors 5% of the time with 95% guarantee.

The theoretical coverage guarantee stipulates that for any given input x, the true output label y will be encompassed within the prediction set $C(x)$ with a probability of at least $1 - \alpha$, where α represents the significance level. This is mathematically expressed as:

$$P(y \in C(x)) \geq 1 - \alpha$$

In other words, the probability of wrong prediction is confined by α. As α diminishes, the size of the prediction set contracts, increasing confidence in model predictions. Conversely, elevating α

expands the prediction set, albeit at the cost of reduced prediction confidence.

For example, setting $\alpha = 0.05$ signifies 95% confidence that the true output label y lies within the prediction set $C(x)$ for any given input x. Leveraging conformal prediction furnishes a robust theoretical framework ensuring the accuracy of model predictions in HT detection, substantiated by the proof presented in Theorem 2.

Theorem 2. *Let $\mathcal{D}$ denote a probability distribution over a set $\mathcal{X} \times \{0, 1\}$, where $\mathcal{X}$ represents input features and $\{0, 1\}$ denotes labels. Consider a binary classifier $f : \mathcal{X} \to \{0, 1\}$, and let $\epsilon \in (0, 1)$ denote a confidence level. Then, the conformal prediction algorithm furnishes a set of predictions $C(x) \subseteq \{0, 1\}$ for each input $x \in \mathcal{X}$ such that:*

$\mathbb{P}[(x, y) \sim \mathcal{D}, y \in C(x)] \geq 1 - \epsilon$ *where $(x, y) \sim \mathcal{D}$ signifies sampling a pair (x, y) from the distribution $\mathcal{D}$.*

Proof: The proof stems from the structure of the conformal prediction algorithm. Given an input x, the algorithm generates a set of predictions $C(x)$ based on observed labels of training examples sharing similar input features with x. The algorithm ensures that each prediction in $C(x)$ possesses a p-value no greater than ϵ for any new input bearing the same feature vector as x. Since the algorithm yields a set of predictions, the probability of at least one correct prediction is at least $1 - \epsilon$.

Corollary 2. *Consider $\mathcal{D}$, f, and ϵ as in Theorem 2. For any sample size n, the conformal prediction algorithm provides a set of predictions $C(x_1), \ldots, C(x_n)$ for each input $x_1, \ldots, x_n \in \mathcal{X}$ such that:*

$\mathbb{P}[\forall i \in \{1, \ldots, n\}, (x_i, y_i) \sim \mathcal{D}, y_i \in C(x_i)] \geq 1 - \epsilon$ *where $(x_i, y_i) \sim \mathcal{D}$ denotes sampling a pair (x_i, y_i) from the distribution $\mathcal{D}$ for each i.*

Proof: The proof follows from a union bound over the n samples:

$$\mathbb{P}[\forall i \in \{1, \ldots, n\}, (x_i, y_i) \sim \mathcal{D}, y_i \in C(x_i)]$$
$$\geq 1 - \sum_{i=1}^{n} \mathbb{P}[(x_i, y_i) \sim \mathcal{D}, y_i \notin C(x_i)]$$
$$\geq 1 - n\epsilon$$

where the second inequality follows from Theorem 2.

Multimodal Learning

Integrated learning across multiple modalities [29] offers a sophisticated approach to tackle intricate challenges by consolidating insights from diverse data sources, encompassing text, images, and audio, among others. In our specific context, we leverage graphical data alongside tabular representations of source circuits. This holistic strategy facilitates the capture of intricate relationships often overlooked when scrutinizing individual modalities in isolation, thereby enhancing the model's predictive capabilities.

From a mathematical standpoint, Let $X_1, X_2, ..., X_M$ denote M distinct modalities of data, each characterized by its feature space $\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_M$.

The primary objective is to elucidate a mapping f that delineates intermodal relationships. This can be mathematically articulated as:

$$f : \mathcal{F}_1 \times \mathcal{F}_2 \times ... \times \mathcal{F}_M \rightarrow \mathcal{Y} \tag{1}$$

where $\mathcal{Y}$ represents the target space, embodying the desired prediction.

The crux of the challenge lies in adeptly consolidating insights from disparate modalities, a task approached through diverse methodologies such as late fusion or early fusion.

In late fusion [30], features are independently extracted from each modality and amalgamated at a later stage. This paradigm treats modalities as distinct entities until decision-making, characterized by:

$$f(x_1, x_2, ..., x_M) = g(h_1(x_1), h_2(x_2), ..., h_M(x_M)) \tag{2}$$

where h_i denotes feature extraction for modality i, and g amalgamates the extracted features.

In early fusion [31], information from diverse modalities converges at the input level, yielding a cohesive joint feature representation, expressed as:

$$f(x_1, x_2, ..., x_M) = h(x_1, x_2, ..., x_M) \tag{3}$$

where h amalgamates the raw input data from all modalities.

Related Works

The application of traditional machine learning (ML) techniques in hardware Trojan (HT) detection has primarily concentrated on modeling methodologies. This involves the development and implementation of algorithms aimed at enhancing the overall accuracy of HT detection systems. The input data for these models typically includes features extracted from the Register Transfer Level (RTL) code, represented both in tabular and graphical formats to illustrate the circuit's structure. Various surveys on ML approaches for HT attack detection have been conducted, as referenced in [17, 18, 21, 32]. Additionally, in some studies such as [33, 34], image classification techniques have been employed, while in others, multimodal image processing has been utilized [35]. The predominant focus of these investigations has been on feature extraction from gate-level netlists and the utilization of ML models such as Support Vector Machine (SVM) [36], Neural Network (NN) [37], eXtreme Gradient Boosting (XGB) [38], and Random Forest (RF) classifiers [39].

With the emergence of Reinforcement Learning (RL) as a successful method in other domains, several endeavors have explored its application in the realm of hardware security. Notable examples include RL-based static detection [40] and RL integrated with adaptive sampling for on-chip detection [41]. In these methodologies, a prevalent strategy entails the initial training of a classifier model followed by the fine-tuning of hyperparameters. This iterative process is designed to mitigate the false negative rate, thereby enhancing the overall accuracy of the model. While Graph Neural Network (GNN) [42, 43] and Abstract Syntax Tree (AST) [44] are generated for the Register

Transfer Level (RTL) code, it remains unclear how these graphical representations can effectively capture both the structural and behavioral attributes of the circuit.

Additionally, addressing the phenomenon of concept drift becomes imperative post-deployment of a model, given the potential dissimilarity between newly acquired data and the original training dataset. An illustrative instance in the domain of security applications is detailed in [45], which entails the transformation of data samples into a lower-dimensional space and the autonomous derivation of a distance metric capable of assessing their disparities. Notably, while concept drift has garnered attention in various domains, its exploration within the context of Hardware Trojans (HTs) remains scant, despite the inherent evolution of HTs over time.

Moreover, the interpretability aspect of machine learning (ML) finds its niche within the realm of hardware security. For instance, SHapley Additive exPlanations (SHAP) have been applied in studies such as [46], [47], and [48], showcasing promising outcomes on benchmark datasets. However, the utility of SHAP is marred by inherent limitations, including the disregard for causality and susceptibility to human biases. It predominantly evaluates feature contributions within a given dataset, neglecting to elucidate their real-world behaviors, which may diverge from the dataset context. The focal point of traditional machine learning approaches in Hardware Trojan (HT) detection primarily revolves around modeling methodologies. This entails the crafting and execution of algorithms aimed at bolstering the overarching accuracy of HT detection systems. The model receives inputs derived from features extracted from Register Transfer Level (RTL) code, which are depicted in both tabular and graphical formats, encapsulating the circuit's architecture. Numerous surveys have been undertaken to explore the applicability of ML in detecting HT attacks.

Many scholarly works has worked into feature extraction from Register Transfer Level (RTL) or gate-level netlists, harnessing ML models such as Support Vector Machine (SVM) [36], Neural Network (NN) [37], eXtreme Gradient Boosting (XGB) [38], and the Random Forest (RF) classifier [39]. Notably, [34] also explores the utilization of image classification techniques in this domain.

Multimodal deep learning (DL) has garnered significant attention within the Artificial Intelligence (AI) community. Early investigations, typified by Deep Boltzmann Machines (DBM), were dedicated to enhancing the model's ability to comprehend probability distributions across diverse input modalities [49]. Furthermore, the application of uncertainty-aware multimodal learning techniques has yielded successful outcomes in healthcare [50] and scenarios featuring multimodal task distributions, particularly in safety-critical environments [51]. In this thesis, I aim to amalgamate graph [52, 53] and Euclidean data modalities, complemented by uncertainty estimation methodologies.

Moreover, within the domain of hardware security, the scarcity of data points representing malicious or Trojan-infected instances is anticipated. Consequently, the utilization of small data becomes imperative [54]. Such practices have been successfully applied across various domains, including material science [55] and anomaly detection [56].

Contributions

In this thesis, the research is centered around the application of multimodal deep learning for hardware trojan identification, with a focus on mitigating associated challenges such as missing modalities and imbalanced datasets. Addressing the first challenge involves effectively managing missing modalities and implementing uncertainty-aware multimodal fusion strategies. To tackle this, we utilize graphical representations of circuits [57] and Euclidean data extracted from processing the Abstract Syntax Tree (AST) of RTL files (Verilog) [58]. While multimodal approaches have been beneficial in improving model accuracy across various domains, their adoption in trojan identification has been limited. For uncertainty-aware multimodal learning, we propose integrating logic at the information fusion level of modalities, utilizing p-values aggregation within a conformal prediction framework.

The second challenge we address is quantifying uncertainty associated with hardware trojan prediction outcomes and ensuring the validity of predicted labels, particularly when dealing with a small number of highly imbalanced data points. Specifically, we aim for a machine learning

classifier capable of predicting the true label of a new data point with a 95% provable guaranteed coverage, crucial for risk-sensitive domains. Developing such a system holds significant potential for decision-makers evaluating detected labels as "Trojan-Infected." Additionally, we explore methods for ranking detected "Trojan-Infected" circuits to facilitate more informed decision-making.

Our primary contributions can be summarized as follows:

- Introduction of a *multimodal learning approach using both graph and Euclidean data* extracted from hardware circuits.
- Proposal of a *model fusion approach leveraging p-values as statistical measures*, systematically evaluating each modality's contribution to overall prediction.
- Addressing the *challenges associated with missing modalities* and resolving issues related to imbalanced and small datasets by leveraging GAN. Introducing the concept of hardware trojan evolution and presenting a method for generating evolving trojans with high precision using a conformalized generative adversarial network.
- Introduction of a novel concept of *guaranteed coverage of the prediction set*, proposing a tunable significance level through conformal prediction for hardware trojan detection. Additionally, defining an algorithm-agnostic and explainability-aware reject prediction made by the machine learning model. When uncertain about identifying evolving trojans, the model rejects the prediction, prompting human intervention for manual investigation.
- Proposal of a *ranking mechanism for evolved trojans* by assigning confidence scores to predictions.

CHAPTER 2

DETECTING EVOLVING HARDWARE TROJANS

> The scientist has a lot of experience with ignorance and doubt and uncertainty, and this experience is of very great importance, I think. When a scientist doesn't know the answer to a problem, he is ignorant. When he has a hunch as to what the result is, he is uncertain. And when he is pretty damn sure of what the result is going to be, he is still in some doubt. Scientific knowledge is a body of statements of varying degrees of certainty – some most unsure, some nearly sure, but none absolutely certain.
>
> — Richard Feynman

Introduction

HT insertion refers to the malicious alteration of a hardware component's design, which can lead to device malfunctions, leakage of sensitive data, or physical damage [1]. With the semiconductor industry increasingly adopting a fabless model, the risk of HTs being inserted during various stages of manufacturing has grown, posing a significant security threat to hardware systems. Traditional HT detection methods, such as signature-based approaches [2], which analyze Integrated Circuit (IC) functionality, layout, and timing, often struggle against sophisticated HT insertion attacks, particularly as Trojans can evolve over time. As a result, there has been a growing trend towards employing machine learning based solutions for more effective HT detection. However, despite efforts to adhere to best practices in ML evaluation, there are concerns about its efficacy in the hardware security context [3]. Many existing ML-based solutions lack sufficient information about dataset distributions and struggle to evaluate attacks in the face of concept drift or evolving datasets. To address these issues, recent studies, such as [4], have examined the effectiveness of ML in addressing hardware security challenges, recognizing that ML may not be a universal solution for all such problems [5].

When employing machine learning methods, a significant concern arises regarding the reliability of models, particularly in detecting HTs. Despite claims of minimal false positive rates, models may not generalize well to unseen data due to concept drift, a phenomenon driven by adversaries' evolving HT insertion techniques. The ramifications of overlooking a false positive extend beyond financial losses to potentially life-threatening scenarios, especially in critical domains such as implantable medical devices. Hence, there is a pressing need for additional evaluation metrics that can complement existing methodologies, ensuring robust decision-making and comprehensive prediction coverage.

This thesis introduces PALETTE, an ex**P**lainable fr**A**mework for evo**L**ving hardwar**E** **T**rojan de**TE**ction in risk-sensitive domains. Drawing on the algorithm-agnostic statistical inference technique of conformal prediction [6], our framework provides risk-aware theoretical guaranteed coverage of predictions, addressing the challenge of concept drift [7]. Notably, PALETTE serves as a non-invasive overlay atop existing ML models, offering a nuanced approach to prediction. Instead of a binary outcome, it furnishes a set prediction of detected labels, ensuring 95% average inclusion of the correct class, thereby enhancing prediction reliability and decision-making confidence.

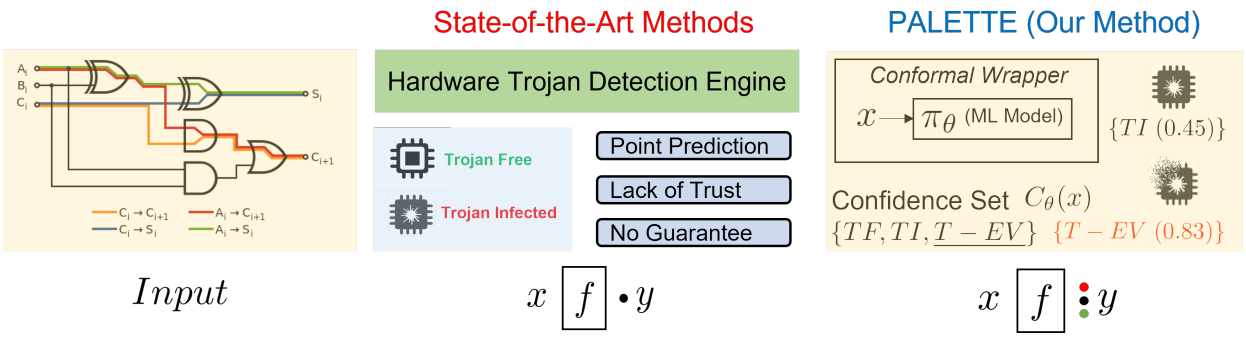


FIGURE 2. Comparison of traditional ML hardware Trojan detector with conformal inference.

In this chapter, we lay the groundwork for hardware security researchers to utilize machine learning in addressing their challenges and promote understanding of risk-managed predictions with assured coverage.

Notion of Evolution & Hardware Trojans

Darwin, in his seminal work *On the Origin of Species*, coined the term "descent with modification" as an alternative to "evolution." [59] Expanding on this concept, Futuyma provided a comprehensive definition of biological evolution, encompassing changes in organism properties over generations. Focusing on hardware Trojans (HTs), we refine the concept of evolution under the assumptions that HT characteristics evolve over time due to deliberate modifications by attackers.

Structural changes in HTs can be represented mathematically, denoted as

$E_{HT} \rightarrow HT \blacksquare HT_{structural\_changes}$ where $\blacksquare$ denotes the operation for structural alterations creating an evolved Trojan E_{HT}.

Behavioral changes are akin to natural selection, where attackers design HTs to adapt to the integrated circuit (IC) environment, making their malicious impact harder to detect. Leveraging these assumptions, we incorporate the notion of evolution to generate an evolved dataset for machine learning-based HT detection engines. While anomaly detection methods can identify evolved HTs, our focus lies on predicting HT evolution, enabling preemptive measures against potential threats. Notably, existing literature lacks consideration of evolutionary aspects in HT detection methodologies.

We utilize the aforementioned assumption to incorporate the concept of evolution and generate an evolved dataset for machine learning-based HT detection systems. In the realm of HTs, we have the option to either detect or forecast their evolution well in advance. While anomaly detection methods can identify evolved HTs, our focus is on predicting their evolution. Anticipating the evolutionary changes in the dataset enables us to implement targeted measures to mitigate the impact of HT insertion. To date, we have not encountered any literature that addresses the evolutionary aspect in the design of HT detection methodologies.

The method proposed in [60] for evolutionary dataset optimization aims to optimize a real-valued function within a subset of all possible datasets. However, due to the non-independent and non-identically distributed (Non-IID) nature of our real-time data, adapting this approach to our

Algorithm 2: Conformalized GAN

Input : Training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^p$ and $y_i \in \{0, 1\}$ are the feature vector and label for the i-th example, respectively; significance level α; number of conformal predictors M; GAN generator G; discriminator model D

Output Conformalized discriminator model D_{CP}

:

for $m = 1$ **to** M **do**
  $\mathcal{D}_m \leftarrow$ bootstrap sample of $\mathcal{D}$;
  Train GAN generator G_m on $\mathcal{D}_m$;
  Generate synthetic dataset $\mathcal{D}^m_{\text{synth}} = \{G_m(z_i)\}_{i=1}^{n}$, where $z_i \in \mathbb{R}^k$ are random noise vectors;
  Train discriminator model D_m on $\mathcal{D}_m \cup \mathcal{D}^m_{\text{synth}}$;
for $i = 1$ **to** n **do**
  $X_i \leftarrow \{x_i\} \cup \{G_m(z_i)\}_{m=1}^{M}$, where $z_i \in \mathbb{R}^k$ are random noise vectors;
  $\text{CP}_i \leftarrow$ conformal predictor trained on (X_i, y_i) with significance level α;
  $p_i \leftarrow \text{CP}_i(D(x_i))$;
Train conformalized discriminator model D_{CP} on $\{(x_i, y_i, p_i)\}_{i=1}^{n}$;
For each sample x_i in the test set D_{test}, make a prediction based on whether $D(x_i)$ is within the prediction interval I_i:

$$y_i = \begin{cases} 1 & \text{if } D(x_i) \notin I_i \\ 0 & \text{if } D(x_i) \in I_i \end{cases}$$

return D_{CP}

specific case is not feasible. An alternative strategy could involve employing evolutionary algorithms, as illustrated in Box2d [61], where the objective is to evolve the structure of a toy car by translating the car's geometry into chromosomes. Nonetheless, a key challenge with this approach is the lack of prior knowledge regarding the structure of the evolved Trojan, which complicates its applicability to our scenario.

Genetic Algorithm

Genetic algorithms (GAs) [62] have been applied to optimize the architecture of neural networks (NNs) for the analysis of logic locking security [63]. The process of designing an effective fitness function poses a significant challenge when employing GA. In our investigation, one potential approach involves assessing the collective efficacy of detection methodologies within an

ensemble framework and subsequently comparing the resemblance of the evolved Trojan with known Hardware Trojans (HTs) stored in a reference dictionary. However, a noteworthy limitation of this fitness function arises from its inability to accurately evaluate Trojans that deviate from the established patterns.

Generative Adversarial Network

Drawing upon principles from game theory and optimization, the primary objective of generative modeling [64] is to scrutinize a set of training instances and glean insights into the probability distribution that generated them. Generative Adversarial Networks (GANs) have exhibited prowess in tasks such as detecting counterfeit images [65] and synthesizing images from textual descriptions [66]. Notably, there has been a discernible shift towards harnessing GANs for processing tabular data, as evidenced by the advent of conditional GANs, as exemplified by Xu et al. [67], which demonstrate efficacy even with datasets exhibiting significant class imbalance. Notable open-source libraries facilitating such endeavors include those developed by Ashrapov [68], Lederrey et al. [69], and Zhao et al. [70].

The motivation for employing GANs in synthesizing datasets for HT detection is threefold.

Addressing Highly Imbalanced Data

In practical scenarios, datasets containing labeled instances of Trojan-Infected circuits are scarce and challenging to discern, resulting in highly imbalanced datasets. Leveraging insights from existing literature, GANs present a viable solution for generating synthetic datasets that exhibit a more realistic distribution, thereby complementing the model training process.

Handling Non-IID Scenarios for Law of Large Numbers

Given the propensity for evolved Trojans to deviate from their originating distribution, it becomes imperative to account for non-IID (non-independent and identically distributed) random variables. Illustratively, scenarios akin to those elucidated in [71] necessitate a departure from traditional statistical assumptions. As dataset sizes burgeon, the statistical reliability and consistency of the data augment. However, the idiosyncrasies of the non-IID case mandate meticulous consideration, given that evolved HTs may

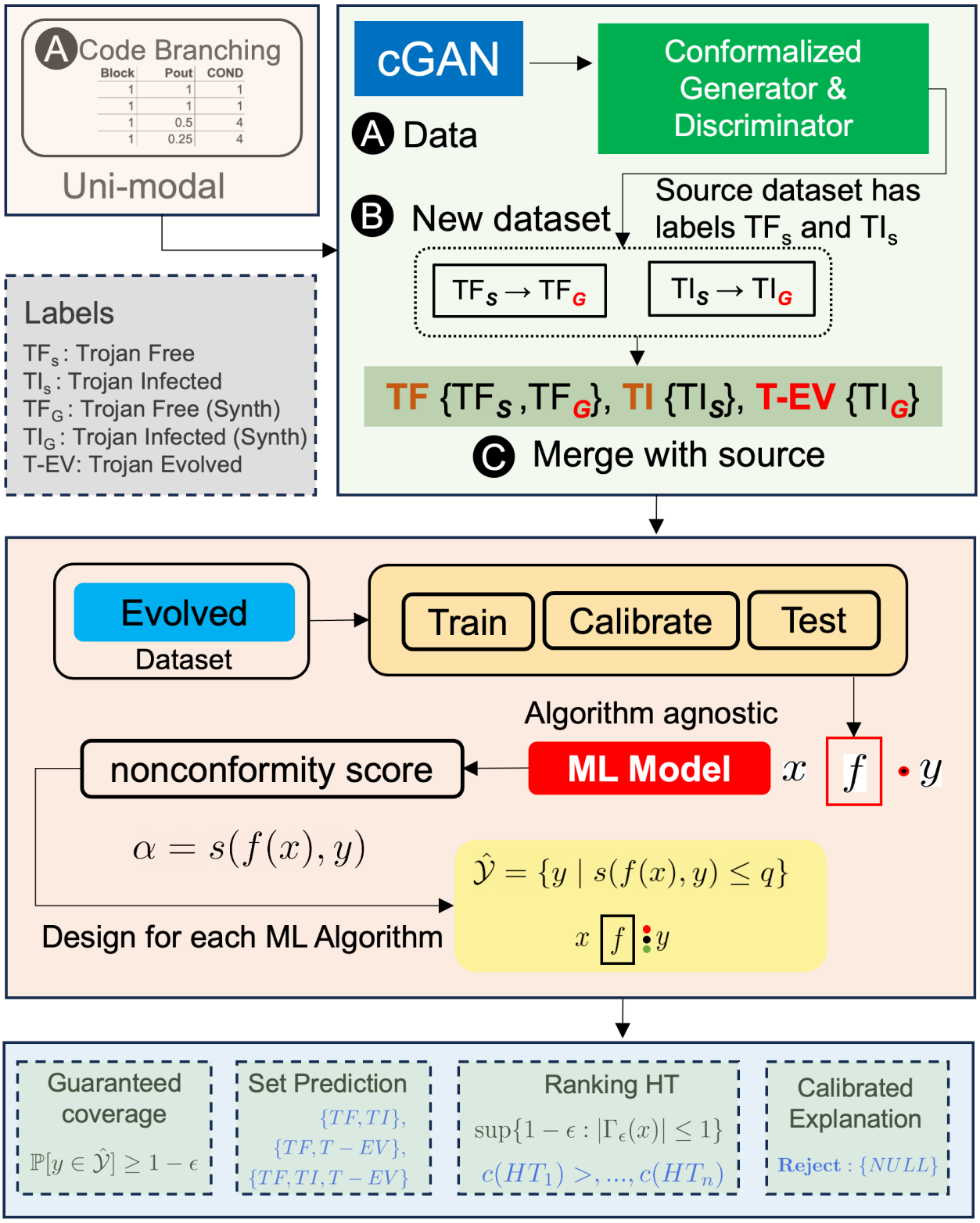


FIGURE 3. PALETTE is a proposed methodology aimed at designing evolving hardware Trojans.

not adhere to the distribution characteristics observed in the original dataset.

Catering to Risk-Sensitive Applications

Amidst the backdrop of potential financial repercussions, the tolerance for false positives diminishes significantly. In light of this, a proactive approach involves crafting synthetic datasets that closely mirror real-world scenarios. Leveraging GANs, particularly by conformalizing both the discriminator and generator components, offers a means to design datasets that encapsulate the nuances of real-world HT insertion scenarios.

Designing & Predicting Evolving Hardware Trojans

The proposed methodology for evolving HT detection, aptly dubbed PALETTE, is delineated in Fig. 3. Comprising four integral components, this framework embodies a concerted effort to anticipate and preemptively address the evolving threat landscape.

(1) In any machine learning based solutions, the initial step invariably involves the extraction of a dataset tailored to the problem at hand. In the context of HT, this dataset may encompass various forms such as images, tables, or graphs, each serving as a representation of the underlying hardware components. For instance, in prior research endeavors [72, 73], scanning electron microscope images have been employed as features extracted from integrated circuits for HT classification. Notably, a trend in HT detection entails the utilization of graph neural networks [42], which involves the conversion of register transfer level code to abstract syntax trees followed by graph-based classification or transformation of the graph into a vector for subsequent classification tasks.

In this thesis, we leverage features derived from code branching, sourced from the comprehensive trustHub chip-level trojan dataset [58], and the synthetic netlist dataset generated via the GAINESIS platform [74]. These datasets constitute a diverse array of representative information, facilitating robust ML model training for HT detection.

(2) The proposed solution introduces the conformalized generative adversarial network (CGAN) algorithm, outlined in Algorithm 2. Inspired by the seminal work of [75], which harnesses principled uncertainty intervals to produce high-fidelity images from corrupted inputs, our algorithm endeavors to generate evolved representations of HTs with a guaranteed containment of

true semantic factors. Drawing from this inspiration, we aim to produce high-quality evolved HT representations by employing conformal prediction techniques on existing datasets such as TrustHub [58] and GAINESIS [74].

The algorithm integrates conformal prediction methods to generate evolving HT instances while simultaneously ascertaining associated confidence levels via prediction intervals. Fig. 4 juxtaposes the TrustHub source dataset with the synthetically generated evolved dataset, illustrating the transition from RTL circuit representations to tabular structures comprising numeric arrays. Unlike conventional GAN approaches, our proposed methodology offers a heightened level of reliability in generating evolving HTs, thus contributing to the advancement of HT detection methodologies.

It's important to highlight that the original dataset is characterized by just two labels: Trojan-Free (TF) and Trojan-Infected (TI). Consequently, the data generated via conformalized GAN also reflects this binary classification. Here, all instances labeled as Trojan-Infected (TI) are categorized as Evolved Trojans, indicated by the label T-EV, due to their generation via GAN methods. This process results in the creation of a comprehensive dataset housing evolved Hardware Trojans, featuring three distinct labels: TF, TI, T-EV.

(3) Following dataset processing, it is input into the conformal inference engine, which, rather than providing singular point predictions, generates set predictions based on predefined significance levels. Notably, this method remains algorithm-agnostic, allowing for the utilization of various machine learning classifiers, whether statistical or deep learning-based, as depicted in Fig. 3. Subsequently, a non-conformity score is computed for each prediction, with the p-value indicating the probability of the prediction's accuracy and facilitating the determination of guaranteed coverage. A crucial aspect of this solution lies in its interpretation within risk-sensitive domains, where even a single erroneous decision cannot be tolerated.

(4) We explore four distinct inferential scenarios rooted in conformal inference. The aim is to quantify the uncertainty linked with each prediction and minimize the False Discovery Rate (FDR)

for Trojan-Free (TF), Trojan-Infected (TI), or Evolving Trojan (T-EV) cases. The initial scenario guarantees coverage, asserting that, based on a user-defined significance level, the predicted label will fall within that class. This involves assigning a significance level considering the risk associated with the prediction, and applying it to the p-values of each label for the circuit's data point. The second scenario leverages conformal prediction's inherent characteristic, producing a set prediction that may encompass all labels TF, TI, T-EV, a blend of labels TF, T-EV or TI, T-EV, or a single label TF, TI, or T-EV. Thirdly, the predicted HTs are ranked by calculating the confidence of each prediction, used to rank the severity of Trojan infection (TI, T-EV). This prioritization aids in determining which instances require immediate mitigation efforts. Lastly, the fourth scenario provides calibrated explanations for predictions where the model is uncertain and rejects the prediction, thus addressing the limitations of local explanations by SHAP. This approach offers a calibrated method to rationalize why a specific prediction must be rejected, signified by a *NULL* set,

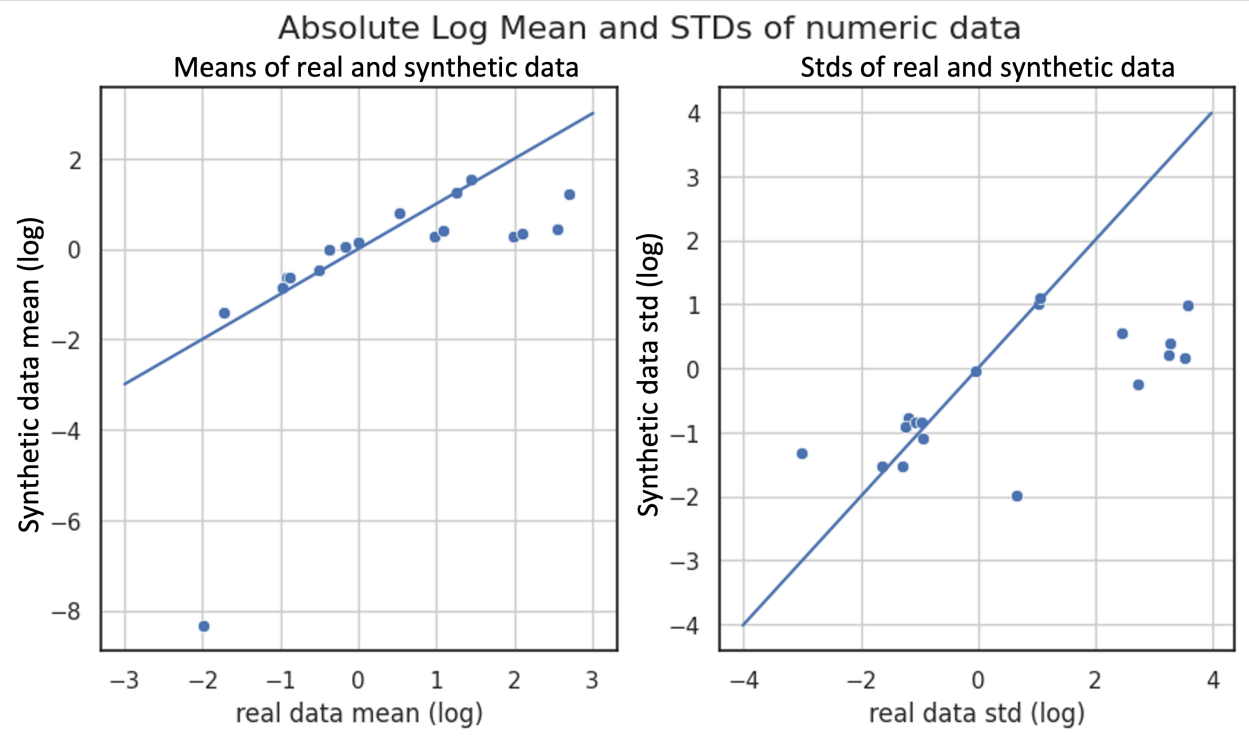


FIGURE 4. Contrasting the authentic trust-hub chip-level trojan dataset.

indicating the model's inability to output a prediction for a specific significance level $(1-\alpha)$. These four customized prediction scenarios, tailored to risk awareness, are detailed alongside experimental findings in Section 9.

Experimental Results

In this section, we present the experimental outcomes based on two distinct datasets. The first dataset originates from GAINESIS [74], a synthetic dataset featuring binary labels. The second dataset is sourced from the Trust-Hub chip-level Trojan dataset [58], which comprises VHDL or Verilog source code files for individual IP core designs, encompassing both malicious and non-malicious functionalities. Typically, the malicious functions are nested within conditional statements that are rarely executed. As a result, the ML features are extracted from these conditional statements. Our solution was implemented using Python (version 3.9) on macOS (version 13.3.1), operating with 8 GB RAM and a built-in GPU. The experimental results, along with the source code and datasets, are publicly available on GitHub [1].

Source Dataset

For our experiment, we utilized features extracted from the TrustHub chip-level Trojan dataset [58], focusing on RTL design using *Code branching features*. This dataset comprises VHDL or Verilog source code files for each IP core design, encompassing both malicious and benign functions. The malicious functions are typically hidden within conditional statements that are rarely executed, thereby prompting the extraction of machine learning features from these conditional statements.

Additionally, we incorporated a synthetic dataset from GAINESIS, characterized by two labels, {TF,TI}. The outcomes of our experimentation will be detailed in the subsequent sections.

Evolved Dataset

Initially, we generated 10,000 data points utilizing the proposed conformalized GAN based on the provided source dataset, selectively choosing only 20% of the evolved dataset. The generated dataset is labeled as TF_G and TI_G, whereas the source dataset comprises labels TF_S and TI_S. In our evolved dataset, we established three distinct labels as follows:

[1] https://github.com/cars-lab-repo/PALETTE/

Trojan-Free (TF), encompassing TF_S and TF_G; Trojan-Infected (TI), solely considering the label TI_S; and finally, Evolved Trojan (T-EV), consisting of the label TI_G.

$$Label = \{TF, TI, T - EV\}$$

Subsequently, the dataset was partitioned into a training set, calibration set, and test set with a ratio of 2:1:1. The training dataset comprised 1436 instances of TF, 114 instances of TI, and 308 instances of T-EV. For calibration, there were 470 TF instances, 33 TI instances, and 117 T-EV instances. Notably, the calibration set included 18% of T-EV instances, while both the train and test sets contained 16% each.

The dataset division into the training, calibration, and test sets is detailed in Table 1.

TABLE 1. Distribution of Dataset for Model Input

	Train	*Calibration*	*Test*
TF	1436	470	471
TI	114	33	44
T-EV	308	117	105
Total count	1858	620	620
T-EV	**16.50%**	**18.87%**	**16.93%**

Baseline Model

We have the flexibility to select any classification algorithm as a baseline model because the PALETTE framework, outlined in Section 1, is agnostic to specific algorithms. In this context, logistic regression serves as our chosen classifier for identifying evolving HTs, and we assess model accuracy as a performance metric. When logistic regression is employed independently for HT detection, the overall accuracy achieves 0.85. However, by integrating conformal inference as a wrapper over logistic regression, the accuracy improves notably to 0.88 for $\alpha = 0.05$ and 0.90 for $\alpha = 0.1$. This enhancement underscores the performance gains achievable by incorporating conformal inference into any classification model. A comprehensive breakdown of results is available on our GitHub repository.

TABLE 2. Confusion Matrix for Base Model and Conformal Inference

	Logistic Regression			*Conformal Inference*		
	TF	**TI**	**T-EV**	**TF**	**TI**	**T-EV**
TF	462	8	8	525	24	44
TI	11	26	2	0	10	7
T-EV	52	0	51	0	0	10

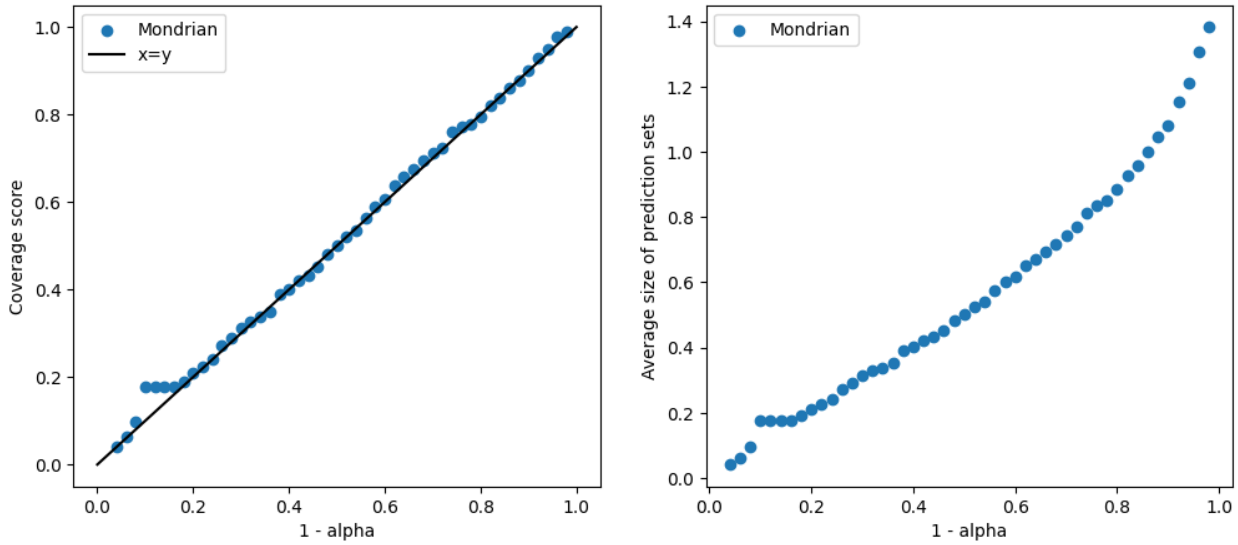


FIGURE 5. Efficient coverage and the average size of prediction sets.

Conformal Inference

It's important to underscore that the conformal inference framework can be coupled with any classification algorithm. In our study, we've opted for logistic regression paired with Mondrian conformal predictors. The *p*-value serves as a gauge of confidence in the predictions made by the ML model. It operates akin to a rating system, indicating the model's performance in predicting new data. To compute the *p*-value, we contrast the model's forecast for new data with its predictions for the data it was trained on, employing hypothesis testing. A low *p*-value for new data suggests significant divergence from the model's prior encounters, potentially indicating less reliable predictions. Hence, it's imperative to exercise caution in interpreting predictions from the model if the *p*-value is exceedingly low.

The outcomes acquired subsequent to implementing conformal inference for identifying evolving HTs are delineated in Table 3. Each row represents an individual circuit, with the truth

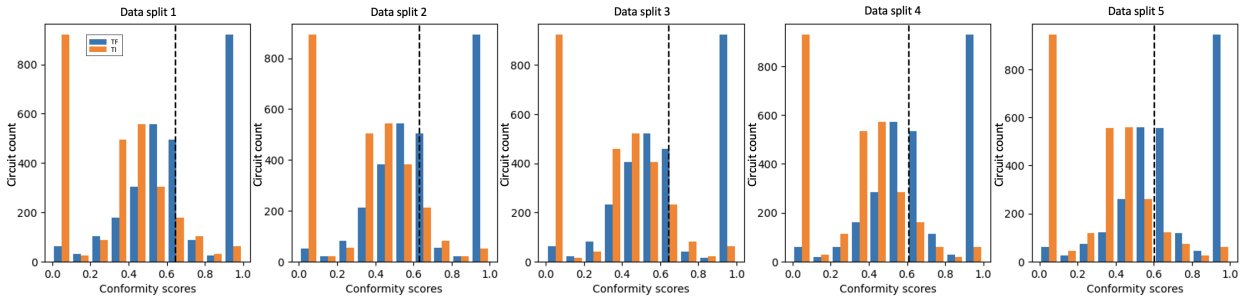


FIGURE 6. The Mondrian conformal predictor's score distribution.

TABLE 3. Conformal Inference and Corresponding p-Values for the Trust-Hub Dataset

	TF	TI	T-EV	pTF	pTI	pT-EV	y_pred	Conf
1	T	F	F	0.319	0	0.003	TF	0.997
2	T	F	F	0.243	0.002	0.006	TF	0.994
3	T	T	F	0.161	0.078	0.016	TF	0.992
4	T	T	T	0.114	0.053	0.119	T-EV	0.886
5	T	F	F	0.645	0.001	0.004	TF	0.996
6	T	F	T	0.653	0	0.971	T-EV	0.365
7	T	F	F	0.3	0	0.002	TF	0.998

values for Trojan-Free (TF), Trojan-Infected (TI), and Evolving Trojan (T-EV) listed in the respective columns. Additionally, the associated p-values for each label are presented in the columns denoted as p_{TF}, p_{TI}, and $p_{\text{T-EV}}$. Furthermore, the predicted Trojan status for each circuit with a significance level of $\alpha = 0.05$ is recorded in the column labeled as y_{pred}. The column designated as Conf signifies the confidence score corresponding to each detected label for every circuit, computed as $1 - 2^{\text{nd}}p_{\max}$.

An application of conformal inference lies in enhancing the quality of detection for evolving HTs. For instance, in Table 3, circuit 2 is identified as Trojan-Free because the p-values for TI and T-EV fall below the threshold of $\alpha = 0.05$. Conversely, circuits 4 and 6 are recognized as infected with an evolved Trojan. In the case of circuit 4, it is observed that the p-values for TF, TI, and T-EV exceed the value of α, leading to all labels being set as True (T), with the maximum p-value specified for the detected label.

Utilizing conformal inference allows us to assert with 95% detection assurance (as determined

TABLE 4. Conformal Inference Applied to the GAINESIS Dataset

circuit	TI	TF	y-pred	Conf
1	FALSE	TRUE	TF	0.891
2	FALSE	TRUE	TF	0.796
3	FALSE	TRUE	TF	0.996
4	FALSE	TRUE	TF	0.997
...	...	...	...	...
4596	FALSE	TRUE	TF	1
4597	FALSE	TRUE	TF	0.991
4598	TRUE	FALSE	TI	0.995
4599	FALSE	TRUE	TF	0.989
4600	FALSE	TRUE	TF	0.992

by $\alpha = 0.05$ selected by the user) that circuit 4 is identified as an evolving Trojan with a confidence score of 0.886. This capability facilitates granular-level reasoning for ensuring trustworthy and robust decision-making processes.

Another property of conformal prediction is "prediction set", the prediction set refers to a range of possible labels assigned to a specific instance, capturing the uncertainty associated with the model's prediction. Instead of providing a single deterministic prediction, conformal prediction offers a set of potential labels along with a measure of confidence or significance level. In current scenario, the set can include all the three labels, any of the two labels, single label, or no labels are all (empty or NULL set).

For example, in Table 3, let's pick instance number 3. Here, we have considered the value of α as 0.05. So, to create a prediction set based on the obtained p-values (derived from the non-conformity measure), we will skip all the labels whose p-value is *less* than 0.05 (α). In this case we will skip T-EV as its p-value is 0.016. This will give us a prediction set TF, TI. The predicted label will be TF because pTF (0.161) is greater than pT-EV (0.078), and the confidence of the prediction is calculated by 1 - 2nd p_max (1 0.078), i.e., 0.922.

Conformal prediction is algorithm-agnostic and can be used as a wrapper over any existing machine learning algorithm, provided that a nonconformity score is designed for each algorithm.

TABLE 5. Analyzing the Effectiveness of Conformal Predictors

alpha	mondrian	raps	naïve	top_k
0.05	10	37	35	0
0.5	45	57	57	61
0.9	45	61	61	61

Let: $\mathcal{A}$ be the set of all machine learning algorithms; $\mathcal{N}$ be the set of nonconformity scores designed for each algorithm in $\mathcal{A}$, and $\mathrm{CP}(\mathcal{N}, \cdot)$ represent the conformal prediction framework using nonconformity scores.

The conformal prediction framework can be applied to any machine learning algorithm $A \in \mathcal{A}$ by using the corresponding nonconformity score $N \in \mathcal{N}$. In mathematical terms:

$$\mathrm{CP}(N, A)$$

This signifies that conformal prediction (CP) is applied to a specific machine learning algorithm (A) using its associated nonconformity score (N). The algorithm-agnostic nature of conformal prediction allows it to serve as a wrapper, accommodating different algorithms by leveraging their respective nonconformity scores.

We present the outcomes for binary labels (TF, TI) concerning the GAINESIS dataset in Table 4. The validation of the method was conducted on a total of 4600 synthetic circuits, encompassing instances both with and without Trojans. The associated confidence scores are detailed in the column labeled *Conf*.

Furthermore, we investigated various adaptations of conformal predictors as outlined in [76]. Table 5 demonstrates that the Mondrian conformal predictor adopts a rigorous approach in detecting evolving hardware Trojans compared to the risk-adaptive prediction set methods such as *raps*, *naive*, and *top_k*, each with different significance levels. The *naive* and *top_k* methods initially retrieve the model output of the true class and then derive the estimated set prediction by extracting quantiles from the score distribution. In contrast, the *raps* method arranges the model output in descending

order to accumulate the output of the true class and subsequently employs it to obtain quantiles from cumulative score distributions. Notably, at a significantly high coverage of 95% (α = 0.05), both *raps* and *naive* methods detect nearly three times more Trojans compared to Mondrian, while the detection coverage becomes nearly equivalent as the coverage level is heightened.

Performance Metrics

In contrast to traditional classification tasks, which typically yield receiver operating characteristic (ROC) curves and area under curve (AUC) scores, conformal inference offers alternative performance metrics: *effective coverage* and *efficiency*, represented by the average prediction set size. Unlike ROC and AUC, which can be influenced by imbalanced datasets, effective coverage and efficiency provide more robust measures of per formance. As depicted in Figure 5, Mondrian conformal predictors exhibit varying performance across these metrics. Effective coverage indicates the proportion of instances where the true label falls within the predicted region, with higher coverage values suggesting a more conservative prediction approach. Conversely, efficiency, reflecting the size of the label sets, serves as a direct measure of the predictor's ability to reject class labels, with smaller sets indicating higher efficiency.

When evaluating conformal prediction methods, various performance metrics are considered, as detailed in Table 6, spanning significance levels from 0.05 to 0 .9. For instance, the avg_c metric signifies the average number of class labels in the prediction sets, providing insight into the predictor's accuracy in discarding class labels. The significance level acts as a threshold governing the frequency of incorrect predictions by the machine learning model. Adjusting this level allows for balancing between prediction accuracy and precision, ensuring optimal model performance.

Additionally, performance metrics for the GAINESIS dataset are illustrated in Figure 6, showcasing the conforming score distribution across the five calibration folds for the Mondrian conformal predictor. Notably, consistent conforming scores are observed across each calibration split, indicating stability in performance across different calibration sets.

TABLE 6. Performance Metrics of Conformal Inference

sig	mean_err	avg_c	n_correct	mean_T-EV
0.05	0.049	1.040	589	0.012
0.1	0.102	0.941	556	0.045
0.2	0.204	0.812	493	0.133
0.3	0.303	0.701	431	0.220
0.4	0.406	0.596	367	0.319
0.5	0.504	0.497	307	0.423
0.6	0.604	0.397	245	0.536
0.7	0.702	0.298	184	0.650
0.8	0.798	0.202	125	0.764
0.9	0.900	0.100	61	0.884

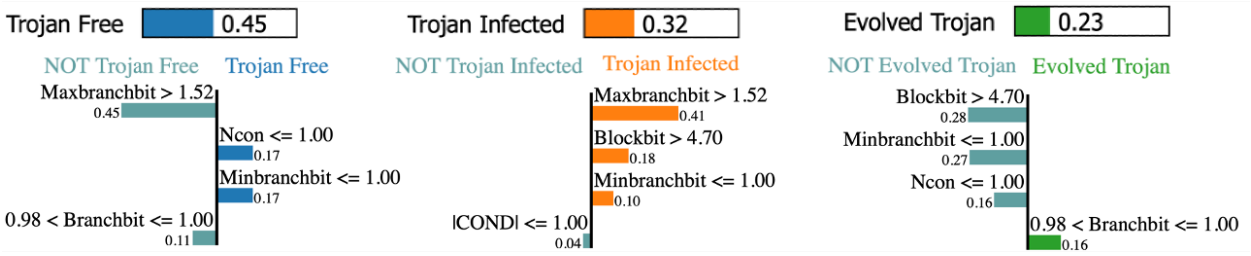


FIGURE 7. Calibrated explanation for decision rejection.

Risk-Aware Ranking

We utilize the confidence scores derived from conformal inference to establish a hierarchical ranking for evolved Hardware Trojans (HTs). For circuits 12, 13, and 14 sourced from the Trust-Hub dataset, their confidence scores (C) were computed using a significance level of $\alpha = 0.05$:

$$\alpha_{0.05}(\text{circuit } 12) = \{T - EV\}_{C=0.88}$$

$$\alpha_{0.05}(\text{circuit } 13) = \{T - EV\}_{C=0.81}$$

$$\alpha_{0.05}(\text{circuit } 14) = \{T - EV\}_{C=0.61}$$

The confidence in the model's predictions is assessed through its p-value, indicating the likelihood of achieving a similar outcome under the null hypothesis. Greater confidence levels correspond to heightened accuracy. The confidence metric is defined as:

TABLE 7. Utilization of Confidence for Risk-Conscious Prioritization

	confidence	credibility	y_pred
1	0.997	0.319	TF
2	0.994	0.242	TF
3	0.922	0.162	TF
4	0.886	0.119	T-EV
5	1	0.645	TF
6	0.999	0.97	T-EV
7	0.998	0.301	TF

$$\text{Confidence}(x) = \sup\{1 - \epsilon : |\Gamma_\epsilon(x)| \leq 1\}$$

Ranking predictions via conformal inference furnishes a nuanced approach to evaluating their reliability. This hierarchical ranking empowers decision-makers to set thresholds or confidence levels for accepting or rejecting predictions based on their position in the hierarchy.

This framework provides a flexible tool for balancing accuracy and reliability across diverse applications. Table 7 illustrates the confidence and credibility of the identified labels, with credibility assessed by considering the maximum p-value within the specified set prediction. Credibility serves as a measure of the quality of new data points.

Calibrated Explanations for Reject

In scenarios where the model fails to detect evolving HTs, it responds with a declaration of uncertainty, represented by an empty set, denoting "I don't know." In contexts sensitive to risk, an absence of output from the model surpasses a decision lacking confidence. Our framework not only offers explanations for decision rejections but ensures their calibration, as illustrated in Figure 7, diverging from conventional explainability methods. For instance, when applying a significance level of 0.5 to a given circuit, none of the p-values for Trojan-Free (TF) (0.45), Trojan-Infected (TI) (0.32), and Evolving Trojan (T-EV) (0.23) surpass the significance threshold, leading to the rejection of the decision. The rationale for this rejection is elucidated through Local Interpretable Model-agnostic Explanations (LIME) [77]. However, unlike SHAP, which overlooks causality and

is susceptible to human biases, our method ensures the calibration of explanations prior to their provision. This process involves generating modified instances of the original data, termed perturbed instances, by introducing minor random alterations. Subsequently, conformal prediction is employed to delineate prediction regions, estimating the reliability or confidence level of the explanations. LIME is then reapplied to these perturbed instances to produce explanations for each. The prediction regions established via conformal prediction serve as a calibration mechanism, guaranteeing that the explanations accurately reflect their degree of reliability.

CHAPTER 3

MULTIMODAL LEARNING FOR HARDWARE TROJAN DETECTION

On what is fear: non-acceptance of uncertainty. If we accept that uncertainty it becomes an adventure! — Jalāl al-Dīn Muḥammad Rūmī

Introduction

The infiltration of HTs has emerged as a pressing issue in today's fabless semiconductor manufacturing landscape. Attackers exploit opportunities to introduce malicious alterations, posing significant security risks such as data breaches, operational malfunctions, and chip damage [8–11]. The intricate stages of manufacturing offer numerous entry points for HT insertion, thus threatening the integrity of hardware systems. Vulnerabilities extend from the initial design phase, encompassing RTL code development and integration of third-party IP, to potential intrusions during electronic design automation (EDA) processes like synthesis and place-and-route. Additionally, vulnerabilities arise during mask preparation [78] and lithography in wafer fabrication, as well as throughout packaging, testing, and post-production phases, including third-party manufacturing and distribution. To address these risks effectively, robust security measures are imperative, spanning hardware design practices [13], supply chain management [14, 15], and comprehensive post-manufacturing testing [16] within the semiconductor industry.

To confront the challenges posed by HTs, a multifaceted approach is essential in today's technological landscape. This approach encompasses rigorous design integrity verification through formal verification [79] and simulation-based testing, complemented by advanced intrusion detection systems (IDS) for continuous monitoring and prompt detection of suspicious activities. Moreover, hardware security measures like obfuscation [80], encryption [81], and secure boot [16] are critical for fortification against HT insertion and mitigation of their impact when detected. Additionally, secure boot processes [16] and real-time monitoring further bolster chip integrity,

while adherence to recognized security certification standards ensures compliance with industry best practices.

Despite the necessity of comprehensive approaches for countering HTs, they are not without challenges. Formal methods and simulation-based testing can be resource-intensive and time-consuming, while intrusion detection systems may generate false alarms, disrupting operations. Establishing a secure supply chain may limit flexibility in supplier selection, and post-manufacturing testing incurs both time and cost. Therefore, achieving a delicate balance between these considerations and the imperative for robust Trojan defenses is paramount for semiconductor manufacturers.

Machine learning has recently emerged as an effective method for detecting HTs [17–21]. ML algorithms can discern patterns indicative of trojans, facilitating the classification of circuits as either trojan-free or trojan-infected. This capability enables continuous monitoring and rapid response to potential threats. However, several challenges accompany this approach. Acquiring large and diverse datasets, particularly those containing rare trojans, poses difficulties. Additionally, ML models are susceptible to adversarial attacks [22], which may undermine their decision-making processes. Ensuring interpretability [23] and explainability [24] of ML-based trojan detection methods is crucial for building trust. Furthermore, the resource-intensive nature of training and deploying ML models may limit accessibility for smaller manufacturers. Continuous retraining is necessary to adapt to evolving Trojan techniques [82], adding complexity to maintenance efforts.

NOODLE, an acronym for Uncertainty-aware Hardware Trojan Detection using Multimodal Deep Learning (NOODLE), is introduced in this chapter to bridge the existing gaps in ML-based methods for identifying HTs. This novel approach combines graph representation and tabular data to perform binary classification, aiming to enhance the accuracy and reliability of HT detection.

Multimodal Hardware Trojan Detection

While recent advancements in HT detection have primarily concentrated on selecting appropriate algorithms and refining dataset representations to enhance accuracy, there has been

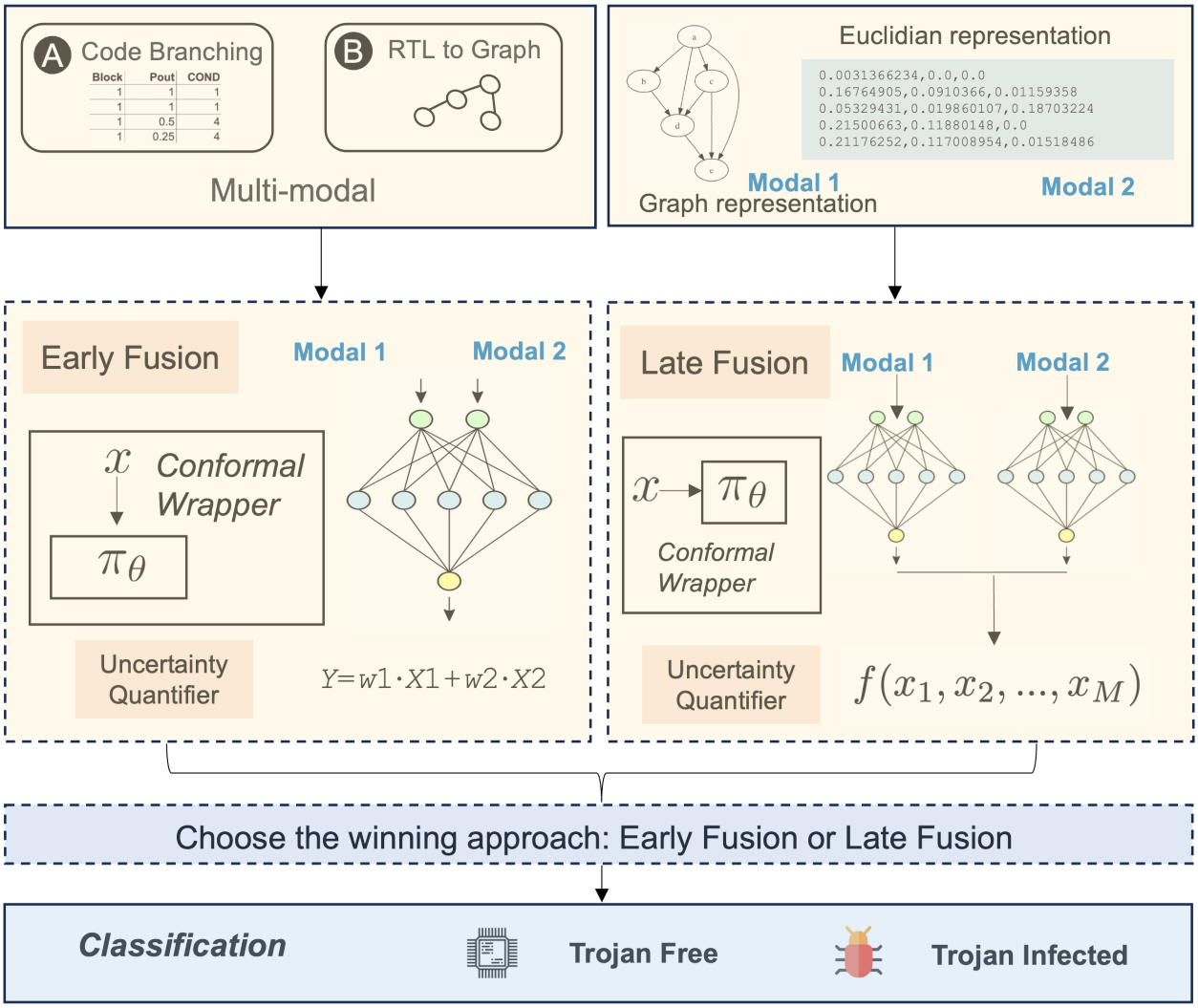


FIGURE 8. NOODLE framework, an RTL file (Verilog) serves as the input.

minimal exploration into integrating diverse modalities of the same data and integrating them into ML systems. Incorporating information fusion from various modalities can yield a more nuanced data representation. Additionally, real-world datasets often contain missing values, resulting in absent modalities when employing a multimodal ML approach. Consequently, a method capable of handling missing modalities in any dataset is essential. Furthermore, within the hardware security domain, acquiring sufficient training data, particularly for labels denoting Trojan-infected instances, is challenging due to the rarity of such occurrences. Therefore, it becomes imperative to develop methodologies that effectively leverage limited data resources.

Benefits of Mulitmodal Learning Approach

The use of multimodal approach had proved better than unimodal approach based on the previous research works.

Let, X_{tabular} be the feature set derived from the tabular (AST) representation, X_{graph} be the feature set derived from the graph (graph2vec) representation, Y be the binary trojan classification (1 for Trojan Induced (TI), 0 for Trojan Free (TF)), $f_{\text{tabular}}(X_{\text{tabular}})$ be the mapping function for the tabular representation.

The model can be represented as a combination of both modalities:

$Y = g(f_{\text{tabular}}(X_{\text{tabular}}, \theta_{\text{tabular}}), f_{\text{graph}}(X_{\text{graph}}, \theta_{\text{graph}}))$

In this section, the benefits are emphasized with the following points.

- Comprehensive information representation: $X_{\text{combined}} = [X_{\text{tabular}}, X_{\text{graph}}]$
- Rich feature set: $X_{\text{combined}} = [X_{\text{tabular}}, X_{\text{graph}}]$
- Enhanced model robustness: $Y = g(f_{\text{tabular}}(X_{\text{tabular}}, \theta_{\text{tabular}}) \cdot f_{\text{graph}}(X_{\text{graph}}, \theta_{\text{graph}}))$
- Improved generalization: $Y = g(f_{\text{tabular}}(X_{\text{tabular}}, \theta_{\text{tabular}}) + f_{\text{graph}}(X_{\text{graph}}, \theta_{\text{graph}}))$
- Handling missing or noisy data:
  $Y = g(\text{impute}(f_{\text{tabular}}(X_{\text{tabular}}, \theta_{\text{tabular}})), f_{\text{graph}}(X_{\text{graph}}, \theta_{\text{graph}}))$

Our proposed framework, NOODLE, as illustrated in Figure 8, underscores the emphasis on design and implementation. Additionally, a pseudocode outlining the framework's operational steps is provided in Algorithm 4. In our approach, we opt for utilizing two modalities: graph and tabular data representations. While previous methodologies have employed techniques like multimodal autoencoders [83] to handle missing modalities, we adopt generative adversarial networks (GANs) [84] to augment the dataset size to 500 data points. GANs aim to generate samples consistent with the joint distribution of the observed modalities, facilitating more effective

multimodal fusion. Specifically, we segregate data points labeled as TF and employ GANs to generate additional TF-labeled data points. We apply the same process to data labeled as TI.

Before using multimodal learning, we elucidate the process of uncertainty-aware model fusion. To achieve uncertainty-aware multimodal fusion, we leverage conformal prediction p-values for

Algorithm 3: Uncertainty-aware information fusion

Input : Number of data sources N;
Training sets for each data source
$T_1 = \{(x_1^{(1)}, y_1), \ldots, (x_n^{(1)}, y_n)\}, \ldots, T_N = \{(x_1^{(N)}, y_1), \ldots, (x_n^{(N)}, y_n)\}$, where $x_i^{(j)}$ is the ith data point belonging to the jth data source and y_i is the class label of the ith data point;
Number of classes M;
Class labels $y^{(i)} \in Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(M)}\}$;
Classifiers $S_1, \ldots, S_N$ for each data source;
Confidence level E.

Output Conformal prediction regions $r_E = \{y^{(j)} : \hat{p}_j > 1 - E, y^{(j)} \in Y\}$.

:

Get the new unlabeled example w.r.t each data source $x_{n+1}^{(1)}, \ldots, x_{n+1}^{(N)}$.
Evaluate conformal predictors and classifiers $S_1, \ldots, S_N$ corresponding to each data source, compute p-values $p_j^{(i)}$, where $i = 1, \ldots, N$ corresponds to the ith data source and $j = 1, \ldots, M$ corresponds to the jth class label.
for *each class label* $y^{(j)}$, $j = 1, \ldots, M$ **do**
    Compute p-value, $\hat{p}_j$, of combined hypothesis from N modalities
return r_E.

Algorithm 4: Multimodal deep learning

Input : RTL-level files (Verliog) of circuits

Output Decision (D) = Trojan-free or Trojan-infected

:

for *each circuit C* **do**
    Convert C to Graph data **G** and Euclidean data **T**.
    if ∃ *missing modalities* **then**
        perform GAN to impute the missing modality.
Feed the modalities to *CNN*-based classifier.
for *each modalities M* **do**
    Use Algorithm 3 for uncertainty-aware information fusion.
    Perform early fusion.
    Perform late fusion.
Choosing the winning fusion method.
return D.

model fusion, as delineated in Algorithm 3. Initially, we employ a convolutional neural network (CNN)-based classifier for graph and tabular data sources, incorporating a specifically designed non-conformity score. This non-conformity score furnishes p-values for each label and each data modality. Subsequently, these p-values are integrated into the conformal prediction framework to obtain calibrated conformal predictions.

$$NS = \sum_{t=1}^{T} B_t(x, y) \tag{4}$$

where $B_t(x, y)$ is the non-conformity score of (x, y) computed from a classifier, h_t. Thus, for every class label $y(j)$, $j \in \{1, ..., M\}$, we have an individual null hypothesis for each data source, $H0_1, H0_2, ..., H0_N$, where M is the number of class labels, which in our case is either TF or TI, and N is the number of data sources. Thus, for every class label $y(j)$, we obtain N p-values, $p(i)$, $i = 1, ..., N$ (one for each modality). These p-values are then combined into a new test statistic $C(p(1), ..., p(N))$, which is used to test the combined null hypothesis $H0$ for class label $y(j)$.

The delineation of the conformal prediction region, as defined by r_E, manifests as a set encompassing all class labels characterized by a p-value exceeding $1 - E$. These procedural steps substantiate the realization of uncertainty-aware multimodal fusion.

Following the acquisition of a substantial corpus of data points for experimentation, the instantiation of multimodal machine learning ensues, leveraging both graph and tabular data. Specifically, the adoption of a convolutional neural network facilitates binary classification. While it remains pertinent to acknowledge the potential for optimization of any machine learning model through hyperparameter tuning to bolster accuracy, our primary directive revolves around the evaluation of the efficacy of uncertainty-aware multimodality, interrogating both early and late fusions. Consequently, the model is poised to engender more informed decisions in the domain of HT detection.

Experimental Results

The implementation of NOODLE was conducted utilizing Python version 3.9, executed on a macOS platform with an 8GB RAM configuration. The experimental outcomes, alongside the source code and dataset, have been made publicly accessible on GitHub[1], facilitating transparency and reproducibility in our research endeavors.

Dataset

In our experimental setup, we meticulously selected datasets that offer comprehensive coverage and intricate insights into the detection of HTs. Specifically, we harnessed the features extracted from the TrustHub RTL-level (Verilog) Trojan dataset, derived from code branching features [58]. This dataset comprises RTL source code files (Verilog) encompassing diverse IP core designs, wherein both malicious and non-malicious functions are embedded. Additionally, we integrated the graph dataset delineated in [57], which augments our analysis by providing an alternative perspective on HT detection. This dataset encapsulates RTL source code files (Verilog) for various IP core designs.

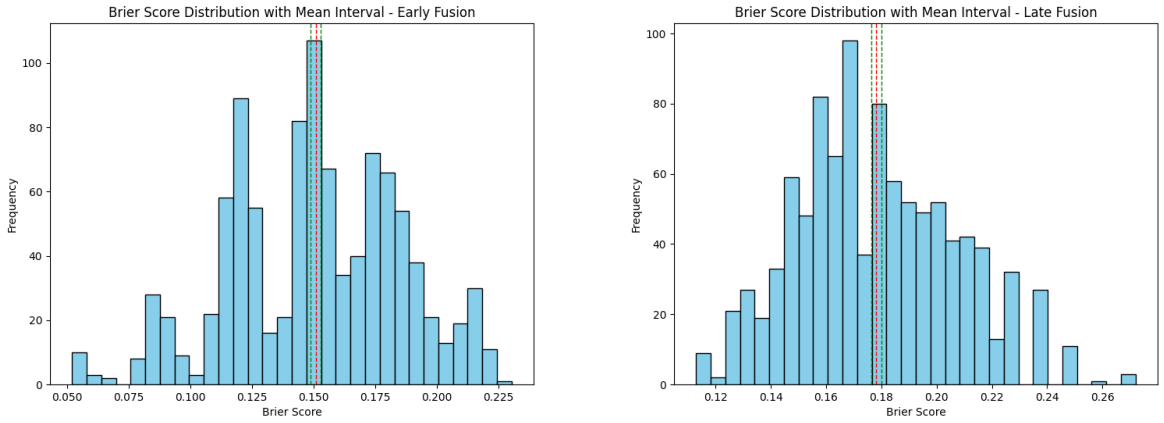
TABLE 8. Comparison of Brier Scores Across Various Modalities

Dataset	Brier Score
Graph-based Data	0.1798
Tabular-based Data	0.1913
NOODLE - Early Fusion (Graph + Tabular)	0.1685
NOODLE - Late Fusion (Graph + Tabular)	0.1589

Brier Score

In many classification scenarios, accuracy serves as the primary metric for evaluating model performance, often supplemented by additional measures such as precision, recall, and F1-score. However, when dealing with imbalanced class distributions, these metrics may not provide a complete picture of model effectiveness. To address this issue in the context of detecting Hardware Trojans (HTs), we employ the Brier score as an alternative evaluation metric. The Brier

[1]https://github.com/cars-lab-repo/NOODLE



(a) **(b)**

FIGURE 9. The Brier scores for NOODLE are presented in fusion approaches.

score offers insights into both accuracy and calibration of probabilistic predictions. Mathematically, it is expressed as:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(p_i - o_i)^2$$

Here, N represents the total number of instances, p_i denotes the predicted probability for instance i, and o_i represents the observed outcome for the same instance. The Brier score ranges between 0 and 1, with 0 indicating perfect accuracy where predicted probabilities precisely align with actual outcomes, and 1 indicating complete inaccuracy, signifying a complete mismatch between predicted probabilities and actual outcomes.

We initiate the evaluation process by conducting separate assessments of each modality. This involves performing binary classification tasks on both the graph dataset and the tabular data. The comparative Brier scores resulting from these classification tasks are summarized in Table 8. The experimental findings reveal that, when employing an identical CNN-based deep learning model with consistent hyperparameters, the graph dataset yields a superior Brier score of 0.1798 compared to the tabular data, which yields a score of 0.1913. It is important to note that while we utilized a

CNN as the baseline model, alternative classification algorithms can also be considered within this framework.

Subsequently, we evaluate the NOODLE framework utilizing two distinct information fusion methodologies: early fusion (feature-level fusion) and late fusion (decision-level fusion). As indicated in Table 8, the early fusion approach, which integrates the graph and tabular data before processing, results in a Brier score of 0.1685. Conversely, the late fusion strategy, which integrates the graph and tabular data after individual processing, exhibits superior performance with a Brier score of 0.1589.

It is essential to acknowledge that neither of these data fusion methods can be unequivocally labeled as superior, as each method may demonstrate its efficacy under varying data distributions [85]. Hence, we implemented both fusion approaches and selected the one that yields a lower Brier score (i.e., closer to 0), as delineated in Step 8 of Algorithm 4. The corresponding Brier score distributions with mean intervals are illustrated in Fig. 9a and Fig. 9b for early and late fusion, respectively. This comprehensive depiction of predictive accuracy across multiple scenarios facilitates model comparison and provides insights into performance variability.

Confidence Calibration Curve

The confidence calibration curve depicted in Figure 10 illustrates the correspondence between observed probabilities and predicted probabilities generated by the classification model. Ideally, a perfectly calibrated model would exhibit all data points aligning along the diagonal. However, in our scenario, the model's calibration is hindered due to the imbalanced nature of the dataset. It is imperative for decision-makers to consider these instances when making risk-aware decisions, emphasizing the limitations of relying solely on accuracy metrics. This evaluation aids in assessing the coherence between the model's predicted probabilities and the actual likelihood of events.

Additionally, the histogram featured at the bottom of Figure 10 showcases the distribution of predicted probabilities for 109 test data points. This visualization offers insights into the sharpness of the predictions, reflecting the propensity of forecasts to cluster towards the extremities of the 0-1

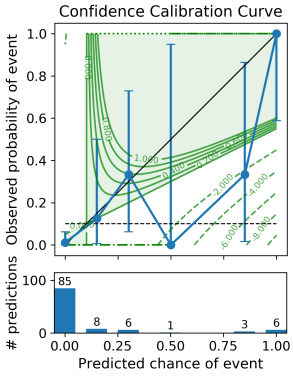


FIGURE 10. Confidence calibration curve of NOODLE

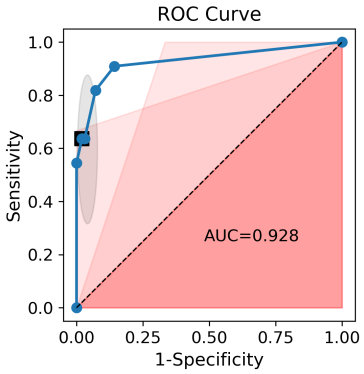


FIGURE 11. ROC-AUC curve of NOODLE with late fusion

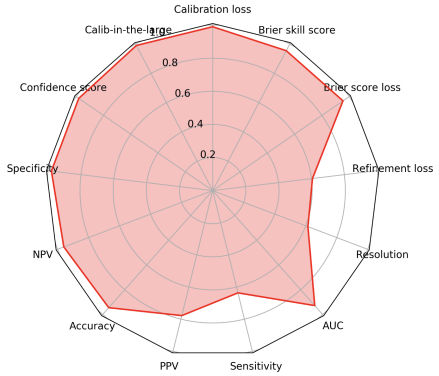


FIGURE 12. Radar plot for aggregated metrics in NOODLE

distribution, which corresponds to the variance of the predictions.

ROC-AUC Curve

The receiver operating characteristic (ROC) curve delineates the equilibrium between sensitivity and specificity within a model, offering a graphical depiction of their fluctuations across different classification th resholds. Conversely, the area under the curve (AUC) serves as a numerical measure denoting the probability that a randomly selected pair of circuits, one with a Trojan and one without, will be correctly classified by the m odel. F ig. 11 depicts the ROC-AUC curve for the NOODLE model.

The white region delineates the optimal performance zone of the model, while the faintly shaded red regions depict acceptable efficacy zo nes. ROC-AUC values vary between 0 and 1, with those closer to 1 indicating a strong ability to differentiate between TF and TI instances with considerable certainty. Conversely, values approaching 0 denote performance inferior to random chance. In this instance, the ROC-AUC value of 0.928 signifies the model's commendable performance.

Radar Plot

The radar plot, depicted in Fig. 12, serves as a valuable tool for visualizing complex, multi-dimensional data. While evaluating the performance of a predictor, there is often a tendency

to focus on a limited set of metrics. However, the radar plot offers a comprehensive perspective by portraying performance across diverse dimensions. In this chart, each variable is represented along its corresponding axis, with some variables normalized to fit within the 0-1 range of the radial axis. Organizing the variables in a manner that clusters related concepts or principles facilitates a thorough evaluation of various aspects of performance.

Within the radar plot, metrics related to discrimination are highlighted, including the AUC, resolution, and refinement loss. Additionally, combined metrics that assess both calibration and discrimination, such as the Brier score and Brier skill score, are presented. Analysis of the radar plot indicates that the model demonstrates lower sensitivity but high accuracy. This suggests that while the model generally provides accurate predictions, it may not effectively identify all instances of Trojan infection. This discrepancy could be attributed to a higher incidence of false negatives, indicating that the model fails to detect some positive cases.

CHAPTER 4

CONCLUSIONS AND FUTURE WORKS

> The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards.
>
> — Gene Spafford

In this thesis, novel methods were designed to fix the evident gaps in contemporary hardware security research. A deep learning approach, complemented by uncertainty awareness, was employed to discern and detect evolving hardware trojans. In this study we systematically addressed the case of missing modalities in the dataset, enhancing the overall quality of the proposed framework. Additionally, the thesis contributed to the field by addressing a previously overlooked evaluation metric, aiming to quantify predictions generated by machine learning methods in the intricate task of hardware Trojan detection.

Chapter 2 presented a methodology for generating a high-quality evolving dataset utilizing a conformalized generative adversarial network. Subsequently, we introduced an algorithm-agnostic framework named PALETTE designed for the detection of evolving hardware Trojans with guaranteed coverage. Additionally, we introduced a novel approach for rejecting decisions by providing calibrated explanations. PALETTE demonstrates efficiency in detecting hardware Trojans while also providing uncertainty quantification for each detection. Our findings underscore potential avenues for researchers in related hardware security domains, such as logic locking [86–89], to reconsider the application of machine learning-based solutions and redefine metrics for evaluating their methodologies. While we acknowledge that there is no foolproof solution against zero-day attacks, a robust method to minimize the likelihood of an attack and a proactive defense approach can significantly mitigate potential threats.

Chapter 3 explored the escalating issue of hardware Trojans clandestinely inserted into chips

during various stages of production, particularly within the context of the increasingly distrustful landscape of fabless manufacturing. We innovatively employed generative adversarial networks to augment our dataset, encompassing two distinct modalities: graph and tabular representations. Furthermore, we introduced an uncertainty-aware multimodal deep learning framework named NOODLE to detect hardware Trojans. Our evaluation encompassed both early and late fusion strategies, offering a comprehensive assessment of our approach's effectiveness. Additionally, we integrated uncertainty quantification metrics for each prediction, facilitating informed decision-making while considering potential risks. The incorporation of multimodality and uncertainty quantification holds promise for tackling other critical challenges in hardware security, such as logic locking [86–89]. These contributions collectively mark a significant advancement in bolstering the security and dependability of hardware systems amid evolving threats.

Future Works

Building upon the foundational work presented in this study, future research endeavors should focus on enhancing the technical aspects of hardware Trojan detection. This entails delving into uncertainty quantification within multimodal deep learning frameworks, achieved through the development of alternative non-conformity measures tailored to the implemented deep learning algorithms.

Exploration of alternative generative models: One avenue for future investigation involves exploring alternative generative models beyond conformalized generative adversarial networks (cGANs). Assessing the efficacy of cutting-edge generative models such as Wasserstein GANs or Progressive GANs could shed light on enhancing the quality of the evolving dataset. This, in turn, could bolster the robustness of hardware Trojan detection mechanisms.

Enhancement of uncertainty quantification methodologies: While the NOODLE framework demonstrates innovation, there is potential for further refinement in uncertainty quantification methodologies. Integrating Bayesian deep learning techniques or ensemble methods might yield more precise and calibrated uncertainty estimates. Such enhancements could enhance the reliability

of the detection system, particularly in dynamic environments.

Additionally, there are avenues for expanding the multimodal capabilities of the NOODLE framework. The incorporation of additional modalities, such as temporal or spectral representations, holds promise for augmenting the framework's ability to discern subtle variations indicative of hardware Trojans. Investigating optimal fusion strategies for these modalities and evaluating their impact on both detection performance and uncertainty quantification represents a fertile area for future exploration. Moreover, there is potential for integrating the proposed frameworks into hardware security testing environments, including Hardware Security Modules (HSMs) or Field Programmable Gate Arrays (FPGAs), for real-world validation. Addressing practical challenges related to resource constraints, latency, and scalability will be paramount to ensuring seamless integration into existing hardware security infrastructures.

REFERENCES

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

[2] A. Gbade-Alabi, D. Keezer, V. Mooney, A. Y. Poschmann, M. Stöttinger, and K. Divekar, "A signature based architecture for trojan detection," in *Proceedings of the 9th Workshop on Embedded Systems Security*, pp. 1–10, 2014.

[3] F. Ceschin, "Spotting the differences: Quirks of machine learning (in) security," (Santa Clara, CA), USENIX Association, Jan. 2023.

[4] E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in *31st USENIX Security Symposium (USENIX Security 22), USENIX Association, Boston, MA*, 2022.

[5] W. Liu, C.-H. Chang, X. Wang, C. Liu, J. M. Fung, M. Ebrahimabadi, N. Karimi, X. Meng, and K. Basu, "Two sides of the same coin: Boons and banes of machine learning in hardware security," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, no. 2, pp. 228–251, 2021.

[6] G. Shafer and V. Vovk, "A tutorial on conformal prediction.," *Journal of Machine Learning Research*, vol. 9, no. 3, 2008.

[7] R. J. Tibshirani, R. Foygel Barber, E. Candes, and A. Ramdas, "Conformal prediction under covariate shift," *Advances in neural information processing systems*, vol. 32, 2019.

[8] H. Salmani, "Hardware trojan attacks and countermeasures," in *Fundamentals of IP and SoC Security: Design, Verification, and Debug* (S. Bhunia, S. Ray, and S. Sur-Kolay, eds.), pp. 247–276, Springer, 2017.

[9] S. Bhasin and F. Regazzoni, "A survey on hardware trojan detection techniques," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2021–2024, 2015.

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

[11] H. Salmani, "Cotd: Reference-free hardware trojan detection and recovery based on controllability and observability in gate-level netlist," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 338–350, 2017.

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

[13] N. Muralidhar, A. Zubair, N. Weidler, R. Gerdes, and N. Ramakrishnan, "Contrastive graph convolutional networks for hardware trojan detection in third party ip cores," in *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 181–191, IEEE, 2021.

[14] E. C. Chang, *Supplier Development Framework in Supply Chain Cybersecurity Evaluation of Small and Medium-sized Enterprises*. PhD thesis, Massachusetts Institute of Technology, 2023.

[15] J. Panduro-Ramirez, D. Buddhi, V. Vekariya, B. G. Pillai, N. Tida, *et al.*, "The effective role of cyber security in supply chain to enhance supply chain performance and collaboration," in *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 838–842, IEEE, 2023.

[16] M. Monjur, J. Calzadillas, and Q. Yu, "Hardware security risks and threat analyses in advanced manufacturing industry," *ACM Transactions on Design Automation of Electronic Systems*, 2023.

[17] K. I. Gubbi, B. S. Latibari, A. Srikanth, T. Sheaves, S. A. Beheshti-Shirazi, S. M. Pd, S. Rafatirad, A. Sasan, H. Homayoun, and S. Salehi, "Hardware trojan detection using machine learning: A tutorial," *ACM Transactions on Embedded Computing Systems*, 2023.

[18] Z. Huang, Q. Wang, Y. Chen, and X. Jiang, "A survey on machine learning against hardware trojan attacks: Recent advances and challenges," *IEEE Access*, vol. 8, pp. 10796–10826, 2020.

[19] K. G. Liakos, G. K. Georgakilas, S. Moustakidis, P. Karlsson, and F. C. Plessas, "Machine learning for hardware trojan detection: A review," in *Panhellenic Conference on Electronics & Telecommunications (PACET)*, pp. 1–6, 2019.

[20] D. Koblah, R. Acharya, D. Capecci, O. Dizon-Paradis, S. Tajik, F. Ganji, D. Woodard, and D. Forte, "A survey and perspective on artificial intelligence for security-aware electronic design automation," *ACM Transactions on Design Automation of Electronic Systems*, vol. 28, no. 2, pp. 1–57, 2023.

[21] T. Ç. Köylü, C. R. W. Reinbrecht, A. Gebregiorgis, S. Hamdioui, and M. Taouil, "A survey on machine learning in hardware security," *ACM Journal on Emerging Technologies in Computing Systems*, 2023.

[22] M. T. West, S.-L. Tsang, J. S. Low, C. D. Hill, C. Leckie, L. C. Hollenberg, S. M. Erfani, and M. Usman, "Towards quantum enhanced adversarial robustness in machine learning," *Nature Machine Intelligence*, pp. 1–9, 2023.

[23] X. Li, H. Xiong, X. Li, X. Wu, X. Zhang, J. Liu, J. Bian, and D. Dou, "Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond," *Knowledge and Information Systems*, vol. 64, no. 12, pp. 3197–3234, 2022.

[24] R. Caruana, S. Lundberg, M. T. Ribeiro, H. Nori, and S. Jenkins, "Intelligible and explainable machine learning: Best practices and practical challenges," in *Proceedings of the 26th ACM*

SIGKDD international conference on knowledge discovery & data mining, pp. 3511–3512, 2020.

[25] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," *Advances in neural information processing systems*, vol. 32, 2019.

[26] T. Löfström, H. Boström, H. Linusson, and U. Johansson, "Bias reduction through conditional conformal prediction," *Intelligent Data Analysis*, vol. 19, no. 6, pp. 1355–1375, 2015.

[27] H. Boström, U. Johansson, and T. Löfström, "Mondrian conformal predictive distributions," in *Conformal and Probabilistic Prediction and Applications*, pp. 24–38, PMLR, 2021.

[28] A. N. Angelopoulos and S. Bates, "Conformal prediction:: A gentle introduction," *Foundations and Trends® in Machine Learning*, vol. 16, no. 4, pp. 494–591, 2023.

[29] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings International Conference on Machine Learning (ICML)*, pp. 689–696, 2011.

[30] V. H. Trong, Y. Gwang-hyun, D. T. Vu, and K. Jin-young, "Late fusion of multimodal deep neural networks for weeds classification," *Computers and Electronics in Agriculture*, vol. 175, p. 105506, 2020.

[31] T. M. Nguyen, T. Nguyen, T. M. Le, and T. Tran, "Gefa: early fusion approach in drug-target affinity prediction," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 19, no. 2, pp. 718–728, 2021.

[32] S. Kundu, X. Meng, and K. Basu, "Application of machine learning in hardware trojan detection," in *2021 22nd International Symposium on Quality Electronic Design (ISQED)*, pp. 414–419, IEEE, 2021.

[33] U. J. Botero, R. Wilson, H. Lu, M. T. Rahman, M. A. Mallaiyan, F. Ganji, N. Asadizanjani, M. M. Tehranipoor, D. L. Woodard, and D. Forte, "Hardware trust and assurance through reverse engineering: A tutorial and outlook from image analysis and machine learning perspectives," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 17, no. 4, pp. 1–53, 2021.

[34] M. Ashok, M. J. Turner, R. L. Walsworth, E. V. Levine, and A. P. Chandrakasan, "Hardware trojan detection using unsupervised deep learning on quantum diamond microscope magnetic field images," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 18, no. 4, pp. 1–25, 2022.

[35] D. C. Bowman and J. M. Emmert, "Hardware trojan detection through multimodal image processing and analysis," in *2022 IEEE International Symposium on Smart Electronic Systems (iSES)*, pp. 712–717, 2022.

[36] C. Bao, D. Forte, and A. Srivastava, "On application of one-class svm to reverse engineering-based hardware trojan detection," in *Fifteenth International Symposium on Quality Electronic Design*, pp. 47–54, IEEE, 2014.

[37] K. Hasegawa, M. Yanagisawa, and N. Togawa, "A hardware-trojan classification method using machine learning at gate-level netlists based on trojan features," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 100, no. 7, pp. 1427–1438, 2017.

[38] C. Dong, J. Chen, W. Guo, and J. Zou, "A machine-learning-based hardware-trojan detection approach for chips in the internet of things," *International Journal of Distributed Sensor Networks*, vol. 15, no. 12, p. 1550147719888098, 2019.

[39] K. Hasegawa, M. Yanagisawa, and N. Togawa, "Trojan-feature extraction at gate-level netlists and its application to hardware-trojan detection using random forest classifier," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, IEEE, 2017.

[40] V. Gohil, H. Guo, S. Patnaik, and J. Rajendran, "Attrition: Attacking static hardware trojan detection techniques using reinforcement learning," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1275–1289, 2022.

[41] H. Chen, X. Zhang, K. Huang, and F. Koushanfar, "Adatest: Reinforcement learning and adaptive sampling for on-chip hardware trojan detection," *ACM Transactions on Embedded Computing Systems*, vol. 22, no. 2, pp. 1–23, 2023.

[42] L. Alrahis, S. Patnaik, M. Shafique, and O. Sinanoglu, "Embracing graph neural networks for hardware security," in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–9, 2022.

[43] A. Hepp, J. Baehr, and G. Sigl, "Golden model-free hardware trojan detection by classification of netlist module graphs," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1317–1322, IEEE, 2022.

[44] T. Han, Y. Wang, and P. Liu, "Hardware trojans detection at register transfer level based on machine learning," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, 2019.

[45] L. Yang, W. Guo, Q. Hao, A. Ciptadi, A. Ahmadzadeh, X. Xing, and G. Wang, "Cade: Detecting and explaining concept drift samples for security applications.," in *USENIX security symposium*, pp. 2327–2344, 2021.

[46] Z. Pan and P. Mishra, "Hardware trojan detection using shapley ensemble boosting," in *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, pp. 496–503, 2023.

[47] E. Downing, Y. Mirsky, K. Park, and W. Lee, "Deepreflect: Discovering malicious functionality through binary reconstruction.," in *USENIX Security Symposium*, pp. 3469–3486, 2021.

[48] G. Severi, J. Meyer, S. E. Coull, and A. Oprea, "Explanation-guided backdoor poisoning attacks against malware classifiers.," in *USENIX Security Symposium*, pp. 1487–1504, 2021.

[49] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," *Journal of Machine Learning Research*, vol. 15, no. 84, pp. 2949–2980, 2014.

[50] U. Sarawgi, *Uncertainty-Aware Ensembling in Multi-Modal AI and its Applications in Digital Health for Neurodegenerative Disorders*. PhD thesis, Massachusetts Institute of Technology, 2021.

[51] C. Almecija, A. Sharma, and N. Azizan, "Uncertainty-aware meta-learning for multimodal task distributions," *ArXiv preprint ArXiv:2210.01881*, 2022.

[52] Y. Ektefaie, G. Dasoulas, A. Noori, M. Farhat, and M. Zitnik, "Multimodal learning with graphs," *Nature Machine Intelligence*, vol. 5, no. 4, pp. 340–350, 2023.

[53] S. Kim, N. Lee, J. Lee, D. Hyun, and C. Park, "Heterogeneous graph learning for multi-modal medical data analysis," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 5141–5150, 2023.

[54] T. Nyiri and A. Kiss, "What can we learn from small data," *Infocommunications Journal, Special Issue on Applied Informatics*, pp. 27–34, 2023.

[55] P. Xu, X. Ji, M. Li, and W. Lu, "Small data machine learning in materials science," *npj Computational Materials*, vol. 9, no. 1, p. 42, 2023.

[56] A. Ghamisi, T. Charter, L. Ji, M. Rivard, G. Lund, and H. Najjaran, "Anomaly detection in automated fibre placement: Learning with data limitations," *ArXiv preprint ArXiv:2307.07893*, 2023.

[57] S.-Y. Yu, R. Yasaei, Q. Zhou, T. Nguyen, and M. A. A. Faruque, "Hw2vec: A graph learning tool for automating hardware security," *arXiv preprint arXiv:2107.12328*, 2021.

[58] H. Salmani, M. Tehranipoor, S. Sutikno, and F. Wijitrisnanto, "Trust-hub trojan benchmark for hardware trojan detection model creation using machine learning," 2022.

[59] C. Darwin, "On the origin of species, 1859," 2016.

[60] H. Wilde, V. Knight, and J. Gillard, "Evolutionary dataset optimisation: learning algorithm quality through evolution," *Applied Intelligence*, vol. 50, pp. 1172–1191, 2020.

[61] E. Catto, "Box2d," *Available fro m: http://www. box2d. org*, 2010.

[62] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[63] D. Sisejkovic, F. Merchant, L. M. Reimann, H. Srivastava, A. Hallawa, and R. Leupers, "Challenging the security of logic locking schemes in the era of deep learning: A neuroevolutionary approach," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 17, no. 3, pp. 1–26, 2021.

[64] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[65] U. Ojha, Y. Li, and Y. J. Lee, "Towards universal fake image detectors that generalize across generative models," *arXiv preprint arXiv:2302.10174*, 2023.

[66] M. Kang, J.-Y. Zhu, R. Zhang, J. Park, E. Shechtman, S. Paris, and T. Park, "Scaling up gans for text-to-image synthesis," *arXiv preprint arXiv:2303.05511*, 2023.

[67] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[68] I. Ashrapov, "Tabular gans for uneven distribution," *arXiv preprint arXiv:2010.00638*, 2020.

[69] G. Lederrey, T. Hillel, and M. Bierlaire, "Datgan: Integrating expert knowledge into deep learning for synthetic tabular data," *arXiv preprint arXiv:2203.03489*, 2022.

[70] Z. Zhao, H. Wu, A. Van Moorsel, and L. Y. Chen, "Gtv: Generating tabular data via vertical federated learning," *arXiv preprint arXiv:2302.01706*, 2023.

[71] R. Yonetani, T. Takahashi, A. Hashimoto, and Y. Ushiku, "Decentralized learning of generative adversarial networks from non-iid data," *arXiv preprint arXiv:1905.09684*, 2019.

[72] N. Vashistha, H. Lu, Q. Shi, M. T. Rahman, H. Shen, D. L. Woodard, N. Asadizanjani, and M. Tehranipoor, "Trojan scanner: Detecting hardware trojans with rapid sem imaging combined with image processing and machine learning," in *ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis*, p. 256, ASM International, 2018.

[73] Q. Shi, N. Vashistha, H. Lu, H. Shen, B. Tehranipoor, D. L. Woodard, and N. Asadizanjani, "Golden gates: A new hybrid approach for rapid hardware trojan detection using testing and imaging," in *2019 IEEE international symposium on hardware oriented security and trust (HOST)*, pp. 61–71, IEEE, 2019.

[74] K. G. Liakos, G. K. Georgakilas, F. C. Plessas, and P. Kitsos, "Gainesis: Generative artificial intelligence netlists synthesis," *Electronics*, vol. 11, no. 2, p. 245, 2022.

[75] S. Sankaranarayanan, A. N. Angelopoulos, S. Bates, Y. Romano, and P. Isola, "Semantic uncertainty intervals for disentangled latent spaces,"

[76] S. Bates, A. Angelopoulos, L. Lei, J. Malik, and M. Jordan, "Distribution-free, risk-controlling prediction sets," *Journal of the ACM (JACM)*, vol. 68, no. 6, pp. 1–34, 2021.

[77] J. Dieber and S. Kirrane, "Why model why? assessing the strengths and limitations of lime," *arXiv preprint arXiv:2012.00093*, 2020.

[78] A. Belous, V. Saladukha, A. Belous, and V. Saladukha, "Methods of detecting hardware trojans in microcircuits," *Viruses, Hardware and Software Trojans: Attacks and Countermeasures*, pp. 453–502, 2020.

[79] A. Nahiyan, M. Sadi, R. Vittal, G. Contreras, D. Forte, and M. Tehranipoor, "Hardware trojan detection through information flow security verification," in *2017 IEEE International Test Conference (ITC)*, pp. 1–10, IEEE, 2017.

[80] N. S. Nishita, A. J. Shamly, N. Sivamangai, R. Naveenkumar, J. C. Marghi, and M. S. C. Bose, "Hardware trojan prevention using logic obfuscation," in *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, pp. 1466–1470, IEEE, 2023.

[81] S. S. Chandra, R. J. Kannan, B. S. Balaji, S. Veeramachaneni, and S. Noor Mahammad, "Efficient design and analysis of secure cmos logic through logic encryption," *Scientific Reports*, vol. 13, no. 1, p. 1145, 2023.

[82] R. Vishwakarma and A. Rezaei, "Risk-aware and explainable framework for ensuring guaranteed coverage in evolving hardware trojan detection," in *EEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 1–9, 2023.

[83] N. Jaques, S. Taylor, A. Sano, and R. Picard, "Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction," in *IEEE*

International Conference on Affective Computing and Intelligent Interaction (ACII), pp. 202–208, 2017.

[84] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[85] I. Gallo, A. Calefati, and S. Nawaz, "Multimodal classification fusion in real-world scenarios," in *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 5, pp. 36–41, 2017.

[86] A. Rezaei, R. Afsharmazayejani, and J. Maynard, "Evaluating the security of efpga-based redaction algorithms," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2022.

[87] A. Rezaei, A. Hedayatipour, H. Sayadi, M. Aliasgari, and H. Zhou, "Global attack and remedy on ic-specific logic encryption," in *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 145–148, 2022.

[88] J. Maynard and A. Rezaei, "Dk lock: Dual key logic locking against oracle-guided attacks," in *International Symposium on Quality Electronic Design (ISQED)*, pp. 1–7, 2023.

[89] Y. Aghamohammadi and A. Rezaei, "Cola: Convolutional neural network model for secure low overhead logic locking assignment," in *Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI)*, p. 339–344, 2023.