Uncertainty-Aware Hardware Trojan Detection Using Multimodal Deep Learning

Rahul Vishwakarma
Computer Engineering & Computer Science Department
California State University Long Beach
Long Beach, CA, USA
rahuldeo.vishwakarma01@student.csulb.edu

Amin Rezaei
Computer Engineering & Computer Science Department
California State University Long Beach
Long Beach, CA, USA
amin.rezaei@csulb.edu

***Abstract*—The risk of hardware Trojans being inserted at various stages of chip production has increased in a zero-trust fabless era. To counter this, various machine learning solutions have been developed for the detection of hardware Trojans. While most of the focus has been on either a statistical or deep learning approach, the limited number of Trojan-infected benchmarks affects the detection accuracy and restricts the possibility of detecting zero-day Trojans. To close the gap, we first employ generative adversarial networks to amplify our data in two alternative representation modalities: a graph and a tabular, which ensure a representative distribution of the dataset. Further, we propose a multimodal deep learning approach to detect hardware Trojans and evaluate the results from both early fusion and late fusion strategies. We also estimate the uncertainty quantification metrics of each prediction for risk-aware decision-making. The results not only validate the effectiveness of our suggested hardware Trojan detection technique but also pave the way for future studies utilizing multimodality and uncertainty quantification to tackle other hardware security problems.**

***Index Terms*—Hardware Trojan; Multimodal Deep Learning; Uncertainty Quantification**

I. Introduction

Hardware Trojan (HT) insertion has become a major concern in today's fabless semiconductor manufacturing since attackers can make malicious modifications for a variety of reasons, such as information leakage, incorrect operation, or inflicting damage on the chip [1]–[4]. While comprehensive approaches are vital for countering HTs, they entail certain drawbacks. Formal methods and simulation-based testing can be resource-intensive and time-consuming. Intrusion detection systems may yield false alarms, disrupting operations. Establishing a secure supply chain can limit flexibility in supplier selection.

Recently, Machine Learning (ML) has emerged as a powerful tool for detecting HTs [5]–[9]. It leverages algorithms to identify intricate patterns indicative of Trojans, even in increasingly sophisticated attacks. By training on diverse datasets, ML models can classify circuits as Trojan-free or Trojan-infected. Real-time processing enables continuous monitoring and immediate threat response. However, challenges exist, for example, acquiring large and diverse datasets, especially for rare Trojans, which can be difficult. Moreover, models are susceptible to adversarial attacks [10], potentially undermining their decision-making. Interpretability [11] and explainability [12] are crucial for trust but can be complex in this context. Additionally, resource-intensive training and deployment may limit accessibility for smaller manufacturers. Continuous retraining is necessary to adapt to evolving Trojan techniques [13], adding complexity to maintenance.

Our goal in this paper is to address the gaps in the current ML-based approaches for identification of HTs by proposing **NOODLE**, an u**N**certainty-aware hardware Tr**O**jan detecti**O**n using multimo**D**al deep **LE**arning. The proposed method uses graph representation and tabular data and performs binary classification.

A. Related Works

The emphasis in traditional ML approaches for HT detection has primarily been on modeling techniques. This entails the development and implementation of algorithms aimed at enhancing the overall accuracy of HT detection. Many research papers have concentrated on extracting features from Register Transfer Level (RTL) or gate-level netlists and employing ML models such as Support Vector Machine (SVM) [14], Neural Network (NN) [15], eXtreme Gradient Boosting (XGB) [16], and the Random Forest (RF) classifier [17]. In [18], image classification techniques are also employed.

Multimodal Deep Learning (DL) has been a well-explored topic in the Artificial Intelligence (AI) community. Early research, exemplified by Deep Boltzmann Machines (DBM) focused on the model's capacity to understand probability distributions across inputs with multiple modes [19]. Additionally, applications of uncertainty-aware multimodal learning [20] have been successfully demonstrated in healthcare [21] and in scenarios involving multimodal task distributions [22], particularly in safety-critical environments. In our work, we target the fusion of graph [23], [24] and Euclidean data as the modalities of interest along with uncertainty estimation.

Moreover, when working in the hardware security domain, it is expected to have fewer data points that are malicious or Trojan-infected. In this context, it is necessary to work with small data [25] and this has been achieved in various domains such as material science [26] and anomaly detection [27].

B. Contributions

In this paper, we investigate the feasibility of applying a multimodal ML approach for HT identification by deriving two data representations of circuits. The first is graphical representation [28] of circuits, and the second is euclidean data

[29] derived by processing the Abstract Syntax Tree (AST) of the RTL files (Verilog). Although the use of multimodal approaches for improved model accuracy has been used in other domains, we do not see any application in Trojan identification. For uncertainty-aware multimodal learning, we believe the logic should be implemented at the information fusion level of the modalities, and thus, we leverage the p-values aggregation with conformal prediction. Our main contributions are as follows:

- Proposing a multimodal learning approach using graph and euclidean data of the hardware circuits. To the best of our knowledge, this study is the first to investigate and implement a multimodal approach for HT detection.
- Suggesting a model fusion approach using p-values with an uncertainty quantifier. By employing p-values as statistical measures, we can systematically assess each modality's contribution to the overall prediction. This not only enhances the interpretability of the fusion process but also enables more robust decision-making.
- Addressing the challenges of missing modalities and solving the issue of handling an imbalanced and small dataset by leveraging generative adversarial networks.

II. Preliminaries

A. Multimodal Learning

Multimodal learning [30] addresses complex problems by integrating information from multiple modalities, such as text, images, and audio, to obtain a comprehensive understanding of a given phenomenon. In our case, we use graphical data and tabular representations of the source circuits. This approach enables models to capture nuanced relationships that may be overlooked when considering each modality in isolation, and thus empowers the model to make more robust predictions.

From a mathematical perspective, multimodal learning involves the integration of data representations into a unified framework. Let $X_1, X_2, ..., X_M$ represent M different modalities of data, each with their respective feature spaces $\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_M$. The task is to learn a mapping f that captures the relationships between these modalities. Mathematically, this can be formulated as:

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

where $\mathcal{Y}$ is the target space, representing the desired prediction.

The challenge lies in effectively combining information from diverse modalities, which can be approached through various techniques such as late fusion or early fusion.

In late fusion [31], features are extracted independently from each modality and then combined at a later stage. This approach treats modalities as separate entities until a decision needs to be made and can be represented as:

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

where h_i represents feature extraction for modality i, and g combines the extracted features.

In early fusion [32], information from different modalities is combined at the input level, resulting in a joint feature representation which can be expressed as:

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

where h combines the raw input data from all modalities.

B. Calibrated Prediction

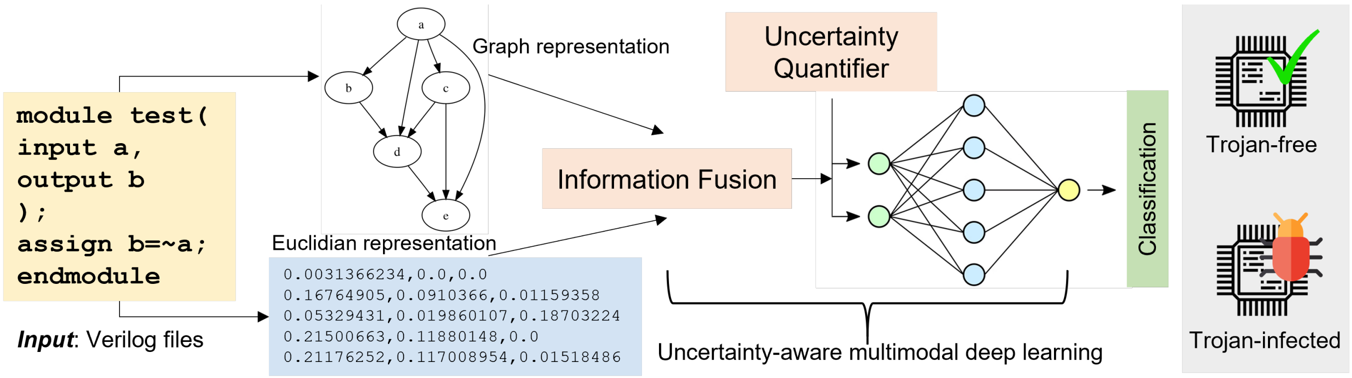
Calibration involves ensuring that a model's confidence score accurately reflects the true probability of the prediction's correctness [33]. Let X be the input data, and Y be the output label. Given a training dataset $D = (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, the goal is to learn a function f that can predict the correct output label y for a given input x. The output of the model for an input x can be denoted as $f(x)$, and the true probability of the prediction's correctness can be denoted as $P(y = 1|x)$. A calibrated model produces a confidence score $g(x)$ that reflects the true probability of correctness of the prediction. The goal of calibration is to ensure that the confidence score $g(x)$ is well-calibrated, i.e., $P(y = 1|g(x) = p) = p$ for all p in the range $[0, 1]$.

Calibration is a crucial aspect in HT detection since it aids in determining the likelihood of the existence of a Trojan in a circuit, which can have a significant impact on decision-making. In situations where a model's confidence score is high, but the likelihood of a Trojan's presence is low, it is reasonable to assume that the circuit does not contain a Trojan. Conversely, if the confidence score is low but the likelihood of a Trojan's presence is high, further investigation of the circuit is necessary.

```
Algorithm 1: 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 $i$th
        data point belonging to the $j$th data source and $y_i$ is
        the class label of the $i$th 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 $i$th data source and
  $j = 1, \ldots, M$ corresponds to the $j$th 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 $r_E$.
```

C. Conformal Prediction

Conformal Prediction (CP) is a ML framework that assesses prediction uncertainty by generating sets of possible predictions [34]. This approach strengthens the inference of conventional



```
module test(
input a,
output b
);
assign b=~a;
endmodule
```

```
0.0031366234,0.0,0.0
0.16764905,0.0910366,0.01159358
0.05329431,0.019860107,0.18703224
0.21500663,0.11880148,0.0
0.21176252,0.117008954,0.01518486
```

Fig. 1: NOODLE framework: The input consists of an RTL file (Verilog), which undergoes conversion into both graph and Euclidean representations, and then input into a multimodal deep learning classifier. This classifier yields a decision indicating whether the circuit is Trojan-infected or Trojan-free.

models, ensuring their reliability and enabling confidence estimation for individual predictions. It is worth noting that minority classes often bear a disproportionate burden of errors when label-conditional validity is lacking [35]. Nevertheless, this challenge can be mitigated by ensuring label-conditional validity, which guarantees that the error rate, even for the minority class, will eventually converge to the selected significance level in the long run.

In certain instances, CP may yield uncertain predictions, signifying that prediction sets contain more than one possible value. This happens when none of the labels can be rejected at the specified significance level. Moreover, when employing CP, the confusion matrix differs from the conventional one due to the distinctive nature of prediction sets, which encompass multiple values rather than a single one. In cases where providing a single-point prediction may be more appropriate than a prediction set or interval in a hedged forecast, opting for the label with the highest p-value is a straightforward and reasonable choice.

There has been limited exploration of the application of CP in modal fusion [36]. This method entails treating individual data sources as separate hypothesis tests within the CP framework. Subsequently, it utilizes p-value combination techniques as a test statistic for the combined hypothesis after the fusion process. Our approach relies on the Mondrian Inductive Conformal Prediction (ICP) methodology [37] outlined in Algorithm 1 for the uncertainty-aware fusion of various modalities during classification. This algorithm can be effectively extended for both early and late fusion of the modalities.

III. Multimodal Hardware Trojan Detection

While state-of-the-art works on HT detection have focused mainly on choosing the right algorithm and choosing different representations of the dataset for improved accuracy, incorporating different modalities of the same data and feeding it to the ML system has not been investigated. By performing information fusion derived from different modalities, a more refined data representation can be achieved. Furthermore, in a practical scenario, we encounter missing values while collecting data, and this may lead to missing modalities when dealing with a multimodal ML approach. So, we also need a method that handles missing modalities for any given dataset. Lastly, in the domain of hardware security, it is difficult to get enough data for training, especially the labels marked as Trojan-infected because of the rarity of the event. In such a situation, we need to work with limited data.

Our proposed *NOODLE* framework is shown in Fig. 1 emphasizing the design and implementation, and a pseudocode is also provided in Algorithm 2. We choose to use two modalities, i.e., graph and tabular data representations. Methods like multimodal autoencoder [38] have been used for handing missing modalities; however, we use Generative Adversarial Networks (GANs) [39] to increase the dataset size to 500 data points as it aims to generate samples that are consistent with the joint distribution of the observed modalities and facilitate more effective multimodal fusion. The data points labeled as Trojan-Free (TF) will be segregated, and only these will be used to generate more data points using GAN so that they represent the same label, and we will do the same for data labeled as Trojan-Infected (TI). Before performing multimodal learning, we first explain the working of uncertainty-aware model fusion.

To perform an uncertainty-aware multimodal fusion, we leverage conformal prediction p-values for the model fusion as

Algorithm 2: 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 $\mathbf{G}$ and Euclidean data $\mathbf{T}$.
    if $\exists$ missing modalities then
        perform GAN to impute the missing modality.
Feed the modalities to CNN-based classifier.
for each modalities $M$ do
    Use Algorithm 1 for uncertainty-aware information fusion.
    Perform early fusion.
    Perform late fusion.
Choosing the winning fusion method.
return $D$.
```

described in Algorithm 1. First, we use a Convolutional Neural Network (CNN)-based classifier for graph and tabular data sources with a designed non-conformity score that provides p-values for each label and for each data modal. The below non-conformity score can be used in the CP framework to get 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 conformal prediction region at a specified confidence level, r_E, is then presented as a set containing all the class labels with a p-value greater than $1 - E$. The mentioned steps helps in realization of uncertainty-aware multimodal fusion.

After obtaining a sufficient number of data points for the experiment, we implement multimodal ML using the graph and tabular data. Specifically, we have employed a CNN for binary classification. It is worth mentioning that any ML model can be optimized through hyperparameter tuning to enhance accuracy. However, our primary emphasis is on assessing the effectiveness of uncertainty-aware multimodality by accessing early and late fusions. Finally, the model will be used to make further informed decisions for the detection of HTs.

IV. Experimental Results

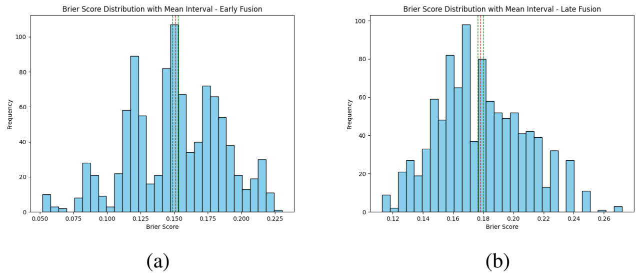
We used Python (3.9) and implemented *NOODLE* on macOS (13.3.1) with 8GB RAM. The experimental results with source code and the dataset are hosted on GitHub[1].

A. Dataset

For our experiment, we have used the features extracted from the TrustHub RTL-level (Verilog) Trojan dataset based on code branching features [29] and the graph dataset in [28] which includes RTL source code files (Verilog) for each IP core design containing both malicious and non-malicious functions.

TABLE I: Brier score comparison for different 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



(a)

(b)

Fig. 2: NOODLE's Brier score (a) Early fusion (b) Late fusion

B. Brier Score

For any of the classification problem statements, the most common performance metric is model accuracy, followed by various other complementing metrics such as precision recall and F1-score. However, these metrics can be misleading in situations where the class distribution is imbalanced, as in our case. For this reason, we have used the Brier score as an evaluation metric for assessing the quality of probabilistic predictions in the classification of HTs. The Brier score, which offers insights into accuracy and calibration, is defined as follows:

$$BS = \frac{1}{N} \sum_{i=1}^{N} (p_i - o_i)^2 \tag{5}$$

where N is the number of instances, p_i is predicted probability for instance i, and o_i is the observed outcome for instance i. The Brier score ranges from 0 to 1. A score of 0 indicates perfect accuracy, meaning the predicted probabilities perfectly match the actual outcomes. A score of 1 signifies complete inaccuracy, where the predicted probabilities are entirely different from the actual outcomes.

We begin the evaluation process by independently assessing each modality. This involves conducting binary classification on both the graph dataset and the tabular data. The resulting comparative Brier scores for these classification tasks are presented in Table I. The experimental outcome demonstrates that, when employing the same CNN-based deep learning model with identical hyperparameters, the graph dataset yields a superior Brier score (0.1798) compared to the tabular data (0.1913). It is worth noting that while we established a baseline model using CNN, any other alternative classification algorithms can also be employed in this context.

Then, we tested *NOODLE* with two different information fusion approaches, i.e., early fusion (feature) and late fusion (decision). As shown in Table I, the early fusion approach, which combines the graph and tabular data before processing, yields a Brier score of 0.1685. On the other hand, the late fusion strategy, which integrates the graph and table data after individual processing, demonstrated the best performance with a Brier score of 0.1589.

It is worth noting that neither of these data fusion methods can be deterministically labeled as superior [40] as each one of them will demonstrate their potential to produce favorable

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

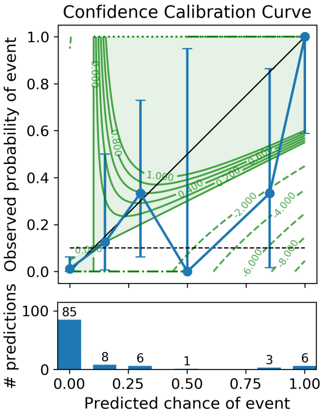


Fig. 3: NOODLE's confidence calibration curve

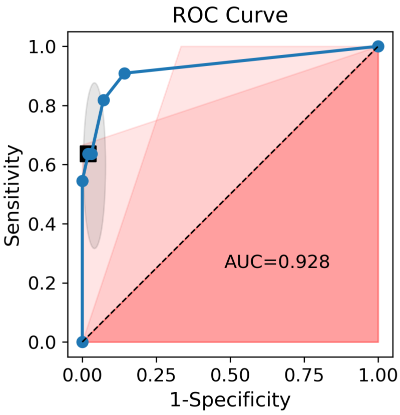


Fig. 4: NOODLE's ROC-AUC curve under late fusion

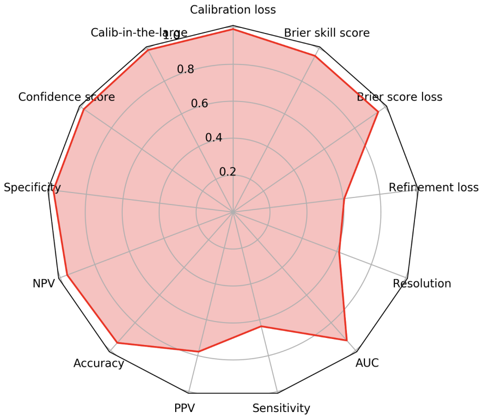


Fig. 5: NOODLE's radar plot for consolidated metrics

outcomes when the data distribution changes. For this reason, we implemented both of the fusion approaches and chose the approach that provides a better Brier score (i.e., closer to 0), as mentioned in Step 8 of Algorithm 2. The corresponding Brier score distribution with mean interval is also shown in Fig. 2a and Fig. 2b for early and late fusion, respectively. This provides a comprehensive view of predictive accuracy across multiple scenarios and is also useful for comparing models and understanding the variability in performance.

C. Confidence Calibration Curve

The confidence calibration curve plots observed probabilities of occurrence as a function of the predicted probabilities for the classification model, as shown in Fig. 3. For the model to be perfectly calibrated, it will have all data points along the diagonal; however, in our case, the model is not well calibrated because of the highly imbalanced dataset. These are the cases on which any decision-maker should focus while making a risk-aware decision and not completely relying on accuracy alone. It helps evaluate the alignment between a model's predicted probabilities and the actual likelihood of events.

A histogram at the bottom of Fig. 3 shows the predicted chance for 109 test data. It describes the distribution of the forecasts and helps with visualization of the sharpness, i.e., tendency of the predictions to lie at the extremes of the 0-1 distribution, and is equal to the variance of the predictions.

D. ROC-AUC Curve

The Receiver Operating Characteristic (ROC) curve illustrates the balance between sensitivity and specificity in a model. It provides a visual representation of how these two metrics change as the threshold for classifying a condition varies. The Area Under the Curve (AUC), on the other hand, quantifies the likelihood that a randomly chosen pair of circuits, one with the Trojan and one without, will be accurately classified by the model. The *NOODLE*'s ROC-AUC curve is given in Fig. 4.

The white area represents the optimal zone for model performance, and the lightly shaded red areas represent the zones of acceptable efficacy. The values for ROC-AUC range from 0 to 1, where values near '1' suggest that it can effectively discriminate between TF and TI cases with a high degree of confidence, and if the value is near '0', the model's performance is worse than random guessing. In our case, the value is 0.928, which suggests that the model is performing well.

E. Radar Plot

The radar plot provides a visual means of presenting complex, multi-dimensional data, as shown in Fig. 5. When appraising the effectiveness of a predictor, there is a tendency to focus narrowly on a limited set of metrics. However, the radar plot provides a method for gaining a comprehensive understanding of performance across diverse dimensions. In a radar chart, each variable is represented along its corresponding axes (some variables have been normalized to conform to the 0-1 range of the radial axis). It is also important to organize the variables in a way that clusters connected ideas or principles. This aids in conducting a thorough evaluation of various facets of performance.

In the given radar plot, we have metrics related to discrimination, which include AUC, resolution, and refinement loss. Following these are combined metrics assessing both calibration and discrimination, namely the Brier score and Brier skill score. As shown in the figure, the model is less sensitive and has high accuracy. This implies that while the model is generally accurate in its predictions, it may not be as effective in identifying all the actual TI cases. This could be due to a higher number of false negatives, which means the model is missing some of the positive cases.

V. Conclusion

In this paper, we have addressed the growing concern of maliciously inserted hardware Trojans into chips at various stages of production in an era where fabless manufacturing is hard to trust. Specifically, we adopted an innovative approach by utilizing generative adversarial networks to expand our dataset with two distinct representation modalities: graph and tabular. Additionally, we introduced an uncertainty-aware multimodal deep learning framework called *NOODLE* for detecting hardware Trojans. We assessed our findings using both early and late fusion strategies, offering a comprehensive evaluation of our approach's efficacy. Moreover, we integrated metrics for uncertainty quantification for each prediction, enabling us to make decisions that are mindful of potential risks. The utilization of multimodality and uncertainty quantification shows great potential for addressing other critical challenges in hardware security such as logic locking [41]–[44]. These contributions collectively represent a significant step forward in enhancing the security and reliability of hardware systems in the face of emerging threats.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Award No. 2245247.

References

[1] 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. Springer, 2017, pp. 247–276.

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

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

[4] 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.

[5] K. I. Gubbi, B. Saber 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*, vol. 22, no. 3, pp. 1–26, 2023.

[6] 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. 10 796–10 826, 2020.

[7] 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)*, 2019, pp. 1–6.

[8] 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.

[9] 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*, vol. 19, no. 2, article 18, 2023.

[10] 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.

[11] 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.

[12] 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*, 2020, pp. 3511–3512.

[13] 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)*, 2023, pp. 1–9.

[14] C. Bao, D. Forte, and A. Srivastava, "On application of one-class svm to reverse engineering-based hardware trojan detection," in *5th International Symposium on Quality Electronic Design (ISQED)*, 2014, pp. 47–54.

[15] 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.

[16] 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, 2019.

[17] 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 *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4.

[18] 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.

[19] 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.

[20] H. Wang, J. Zhang, Y. Chen, C. Ma, J. Avery, L. Hull, and G. Carneiro, "Uncertainty-aware multi-modal learning via cross-modal random network prediction," in *European Conference on Computer Vision (ECCV)*, 2022, pp. 200–217.

[21] U. Sarawgi, "Uncertainty-aware ensembling in multi-modal ai and its applications in digital health for neurodegenerative disorders," Ph.D. dissertation, Massachusetts Institute of Technology, 2021.

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

[23] 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.

[24] 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, no. 4, 2023, pp. 5141–5150.

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

[26] 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.

[27] 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.

[28] 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. [Online]. Available: https://dx.doi.org/10.21227/j1vv-hw18

[29] H. Salmani, M. Tehranipoor, S. Sutikno, and F. Wijitrisnanto, "Trust-hub trojan benchmark for hardware trojan detection model creation using machine learning," 2022. [Online]. Available: https://dx.doi.org/10.21227/px6s-sm21

[30] 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)*, 2011, pp. 689–696.

[31] 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.

[32] 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.

[33] 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," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 14 003–14 014.

[34] G. Shafer and V. Vovk, "A tutorial on conformal prediction." *Journal of Machine Learning Research*, vol. 9, no. 3, pp. 371–421, 2008.

[35] 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.

[36] V. N. Balasubramanian, S. Chakraborty, and S. Panchanathan, "Conformal predictions for information fusion: A comparative study of p-value combination methods," *Annals of Mathematics and Artificial Intelligence*, vol. 74, no. 1-2, pp. 45–65, 2015.

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

[38] 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)*, 2017, pp. 202–208.

[39] 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.

[40] 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, 2017, pp. 36–41.

[41] 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.

[42] 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)*, 2022, pp. 145–148.

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

[44] 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)*, 2023, pp. 339–344.