# PATCH-FFT: Unmasking Dormant Hardware Trojans with Patch-Based Frequency-Domain Transformers

Hasala Senevirathne
*Computer Engineering & Computer Science Department*
*California State University Long Beach*
Long Beach, CA, USA
hasala.senevirathne01@student.csulb.edu

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

***Abstract*—Hardware Trojans embedded by malicious entities in integrated circuits can covertly leak sensitive information through power side channels, often remaining undetected in their dormant state until specific trigger conditions activate their malicious behavior. For information-leaking Trojans, detection in the dormant state is critical, as once triggered, the secret data is already exfiltrated. This paper introduces a patch-based Transformer architecture for detecting dormant hardware Trojans through frequency-domain analysis of power traces. Our approach converts time-domain power measurements into frequency-domain representations using real Fast Fourier Transform (rFFT), revealing spectral signatures hidden in conventional time-series analysis. Experimental results demonstrate that our method achieves 90.94% average detection accuracy across dormant and active Trojan scenarios, outperforming state-of-the-art approaches particularly in detecting dormant Trojans that prior time-domain methods do not address.**

***Index Terms*—Hardware Trojan Detection, Transformer, Frequency Domain, Power Side-Channel, FFT, Patch-based Classification**

## I. Introduction

Globalized semiconductor supply chains have introduced serious security risks in modern Integrated Circuits (ICs), with Hardware Trojans (HTs) among the most dangerous [1], [2]. These malicious modifications can leak cryptographic keys, steal data, or undermine system integrity while evading standard tests. Among the most critical class of HTs are information-leaking Trojans, which exfiltrate secret data through covert side channels. For these Trojans, detection after triggering is too late since the sensitive information has already been compromised. Thus, the only meaningful defense window is detecting the Trojan before it activates.

Contemporary detection approaches predominantly rely on time-domain analysis of side-channel signals, employing Machine Learning (ML) techniques such as Long Short-Term Memory (LSTM) [3], Hierarchical Temporal Memory (HTM) [4], and Convolutional Neural Network (CNN) [5] to identify anomalous power consumption patterns indicative of "active" HTs. Static-power and IDDQ (quiescent supply current)-based techniques [6]–[8] can in principle target dormant Trojans by observing the leakage current contributed by the inserted gates, but they generally require multi-pad probing, embedded on-die sensors, or population-level calibration. Time-domain dynamic-power ML detectors exhibit fundamental limitations when confronting "dormant" HTs that induce minimal perturbations in power traces [9]–[21], and only a few of these prior dynamic-power ML works evaluate detection performance in the dormant state [22].

A critical observation motivating our work is that information-leaking HTs, even when dormant, introduce physical circuit modifications (e.g., modulation logic, shift registers, additional transistor paths) that perturb the power spectrum in structured ways. Modulation-based Trojans, such as those using Code Division Multiple Access (CDMA) encoding, contain clock-driven sequential elements that produce periodic spectral artifacts at specific harmonics. Leakage-based Trojans introduce subtler broadband perturbations through additional sub-threshold current paths. These spectral signatures, while imperceptible in raw time-series data, become evident when power traces are transformed into frequency representations through real Fast Fourier Transform (rFFT) analysis. Moreover, these signatures are localized in frequency bands yet exhibit harmonic relationships across the spectrum, requiring a model that captures both local spectral features and global cross-band dependencies.

Patch-based Transformer architectures [23], [24] are naturally suited to this structure: dividing the frequency spectrum into patches captures local spectral coherence, while self-attention over the full patch sequence models the global harmonic relationships characteristic of Trojan-induced modulations. To the best of our knowledge, **this work is the first patch-based Transformer applied to frequency-domain dynamic-power traces for dormant Trojan detection, requiring neither multi-pad probing nor on-die sensors at inference time**. The key contributions of this work are:

- Proposing a patch-based Transformer for frequency-domain power traces via segmented spectral analysis, enabling local feature extraction and global context through self-attention;
- Demonstrating that rFFT-based frequency-domain representations reveal dormant Trojan signatures invisible to time-domain dynamic-power approaches, especially for information-leaking HTs that use modulation-based covert channels;
- Providing interpretable layer-wise attention maps that reveal which frequency bands drive detection decisions, showing alignment with known Trojan leakage mechanisms;
- Validating the approach through extensive experiments on eight information-leaking Trojan variants, achieving 90.94% average accuracy across dormant and active scenarios and remaining stable under synthetic process-variation augmentation.

## II. Related Work

**Side-channel HT detection.** Side-channel analysis has proven effective for HT detection. Path delay fingerprinting [25] identifies timing anomalies but requires extensive characterization. Power consumption analysis [26] compares measurements with golden models but struggles with process variations. In particular, a Siamese deep learning framework is proposed that processes power side-channel traces, reporting up to 86.78% detection accuracy on the Trojan Power & EM Side-Channel dataset [3]. A brain-inspired model known as HTM has also been proposed for HT detection, reporting 92.2% on triggered Trojans only [4]. Gate-level netlist approaches such as MLNNs (MLNNs) [27] and recurrent netlist detectors [28] form a separate family that does not operate on power traces. None of the above dynamic-power ML works report dormant-state accuracy.

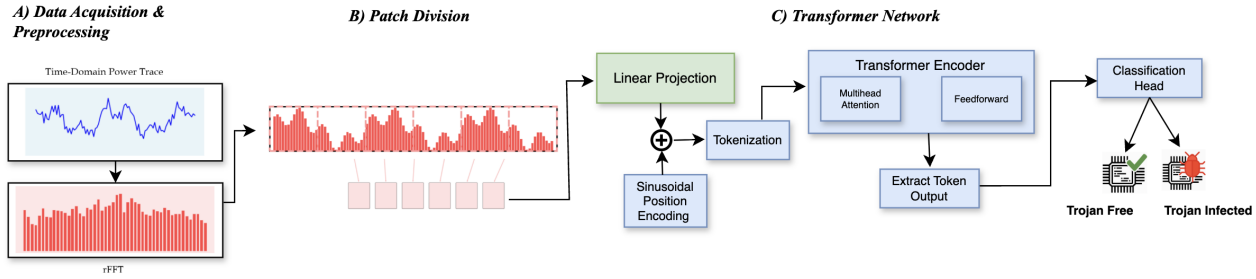


Fig. 1. Overview of **PATCH-FFT**: rFFT log-magnitude → 1D patch embedding → Transformer encoder → binary head.

**Static-power and IDDQ-based detection of dormant Trojans.** A separate line of work targets dormant Trojans by exploiting the leakage current contributed by the inserted gates. Gate-level leakage characterization is formulated as a linear program against the design's gate model, achieving high simulated accuracy but relying on segmentation that becomes ill-conditioned under realistic process variation [6]. IDDQ is measured simultaneously from multiple supply pads, and spatial anomalies are exploited to localize Trojans, but this approach requires multi-pad probing and population-level calibration [7]. An on-chip $V_{DD}$ sensor array is embedded, and Trojan-induced IR drops are statistically fingerprinted across a lot of dies [8]. While these techniques demonstrate that the dormant-detection problem is not categorically intractable, they require either simulation-only flows, multi-pad/embedded-sensor instrumentation, or golden population statistics. Our work is complementary: we use a single, externally observed dynamic-power trace and a learned spectral model, addressing the same dormant HT detection goal under significantly weaker measurement assumptions.

**Time series spectral analysis.** FFT-based side-channel methods [29], [30] have demonstrated superior noise resilience and feature extraction capabilities compared to time-domain approaches. The rFFT variant leverages the Hermitian symmetry inherent in real-valued signals, offering computational efficiency while retaining all relevant spectral information. The patch-based paradigm, pioneered by Vision Transformers (ViT) [24], treats input data as sequences of local regions while maintaining global context through attention. PatchTST [31] demonstrated that patch-based processing of time-series data improves both computational efficiency and modeling capacity. We extend frequency-domain power analysis, where patches map to spectral bands carrying HT-induced modulations.

## III. Architecture

In this section, we propose **PATCH-FFT**, a **P**atch-based **A**pproach for **T**rojan **C**ountermeasures in **H**ardware based on **F**requency-domain **F**eatures and **T**ransformers shown in Fig. 1. **PATCH-FFT** identifies three scenarios: (1) **No Trojan**: The circuit is clean and behaves according to specifications; (2) **Dormant Trojan**: A Trojan exists but has not yet been triggered; and (3) **Active Trojan**: A Trojan is triggered and executes its payload. The architecture leverages rFFT to reveal spectral signatures invisible in time-domain representations, followed by patch-based Transformer processing that captures both local frequency features and global patterns of Trojan activity.

### A. Frequency-Domain Transformation

Instead of operating directly on raw time-domain samples, we represent each power trace as a compact frequency-domain signature. Given a discrete-time signal $x[n]$, $n = 0, \ldots, T-1$, we compute the rFFT and retain the $\lfloor T/2 \rfloor + 1$ non-redundant positive-frequency coefficients exploiting Hermitian symmetry. To obtain a real-valued spectral representation, we map complex coefficients to their magnitudes and apply log-magnitude scaling with per-trace $z$-score normalization:

$$\tilde{X}_{\log}(k) = \frac{\log(|X_{\text{rFFT}}(k)| + \varepsilon) - \mu_x}{\sigma_x + \varepsilon}, \tag{1}$$

where $\mu_x$ and $\sigma_x$ are the mean and standard deviation of $\log(|X_{\text{rFFT}}(\cdot)| + \varepsilon)$ across the $\lfloor T/2 \rfloor + 1$ frequency bins of trace $x$, and $\varepsilon$ is a small constant for numerical stability. This normalization removes global scale and offset, encouraging the downstream model to focus on relative spectral shapes rather than absolute power levels.

TABLE I
PATCH-FFT ARCHITECTURE AND TRAINING HYPERPARAMETERS.

| Hyperparameter | Value |
|---|---|
| Input length $F = \lfloor T/2 \rfloor + 1$ (with $T = 2500$) | 1251 |
| Patch size $p$ | 16 |
| Model dimension $d_{\text{model}}$ | 128 |
| Attention heads $h$ ($d_k = d_v = 16$) | 8 |
| Encoder layers $L$ | 4 |
| FFN dimension $d_{\text{ff}}$ | 256 |
| Dropout | 0.1 |
| Label-smoothing $\alpha$ | 0.02 |
| Optimizer / weight decay | AdamW / $10^{-4}$ |
| Learning rate (cosine, 2-epoch warm-up) | $10^{-3}$ |
| Batch size | 128 |

### B. Patch Embedding, Transformer, and Classification Head

The spectrum $\tilde{X}_{\log} \in \mathbb{R}^F$ (with $F$=1251 for $T$=2500) is divided into $N = \lceil F/p \rceil = 79$ non-overlapping patches of length $p$=16. The final patch is zero-padded to equal length. Each patch is linearly projected to a $d_{\text{model}}$=128 token. We add sinusoidal positional encodings [23] and prepend a learnable CLS token, producing an 80-token input sequence.

The sequence passes through $L$=4 Transformer encoder blocks. Each block applies $h$=8-head self-attention:

$$\text{Attention}(Q, K, V) = \text{Softmax}(QK^\top / \sqrt{d_k})V, \tag{2}$$

followed by a $d_{\text{ff}}$=256 feed-forward network with residual connections, layer normalization, and dropout 0.1. The normalized CLS representation is mapped through a linear classification head to two output logits corresponding to *TrojanDisabled* and *TrojanEnabled*. During training, we apply a cross-entropy loss with label smoothing. Algorithm 1 summarizes the inference pipeline.

**Algorithm 1** PATCH-FFT inference for one power trace. The same procedure is invoked with the dormant-detection checkpoint (positive class = *TrojanEnabled*) or the active-detection checkpoint (positive class = *TrojanTriggered*).

**function** DETECTTROJAN($x$, model, positive_class)
  $X \leftarrow \mathrm{rFFT}(x)$
  $X_{\log} \leftarrow \log(|X| + \varepsilon)$
  $\tilde{X}_{\log} \leftarrow z\text{-score}(X_{\log})$ ▷ $z$-score over frequency bins
  $\mathbf{h}_{\mathrm{cls}} \leftarrow \mathrm{TransformerEncoder}(\mathrm{Patchify}(\tilde{X}_{\log}))$
  $\hat{y} \leftarrow \arg\max\big(\mathrm{Softmax}(W_{\mathrm{cls}}\mathbf{h}_{\mathrm{cls}} + \mathbf{b}_{\mathrm{cls}})\big)$
  **if** $\hat{y} = 1$ **then return** positive_class
  **else return** *TrojanDisabled*
  **end if**
**end function**

## IV. Experimental Setup

### A. Threat Model and Assumptions

This work targets key-leakage HTs inserted during the design or fabrication process. The evaluation uses the public IEEE/TrustHub dataset, which reproduces this threat on a commercial FPGA platform; claims are scoped to this FPGA-based evaluation, and ASIC silicon validation is left as future work.

**PATCH-FFT** requires golden chips only during training, where labeled Trojan-free and Trojan-induced traces are used to learn the classifier. At deployment, the model takes a single externally measured power trace from the device under test and outputs a binary verdict. The measurement requires only an external shunt resistor and oscilloscope, with no multi-pad probing, on-die sensors, IDDQ instrumentation, or invasive optical access, matching golden-free inference [3], [4] and avoiding extra instrumentation [7], [8].

### B. Dataset and Preprocessing

We evaluate **PATCH-FFT** using the publicly available IEEE HT Power & EM Side-Channel Dataset [32], which includes power traces from AES-128 implementations carrying TrustHub Trojan variants. The target design is synthesized onto the Xilinx Spartan-6 (45 nm) cryptographic FPGA of a SAKURA-G side-channel evaluation board [33]; the supply current is acquired through the on-board shunt resistor on the $V_{cc\text{-}int}$ rail and digitized by an external oscilloscope. All traces are captured at $25^{\circ}\mathrm{C}$ from a single FPGA instance per benchmark, so the public dataset does not span natural die-to-die process variation. For each Trojan configuration (Trojan-free baseline, Trojan-disabled/dormant, Trojan-triggered/active) the dataset provides on the order of 10,000 traces, where each trace is a univariate time series of length $T = 2500$ samples captured during one AES encryption. For this study, we focus on eight Trojan variants that employ key-leakage mechanisms. Although AES-T400/T1600/T1700 use "RF/EM radiation" as the listed leakage channel, the modulator and counter logic that drives those off-die emissions also draws clock-correlated current from the FPGA core rail; the supply-rail shunt trace carries a weaker replica of the same modulation, and we retain it as the sole input for all eight benchmarks.

### C. Training Setup

We use the power side-channel traces and reorganize them into two binary classes, *TrojanDisabled* and *TrojanEnabled* or *TrojanDisabled* and *TrojanTriggered*. For each AES-Tx benchmark we (i) pool all available traces belonging to the two relevant classes, (ii) shuffle them with a fixed seed, and (iii) partition them into a $70\%/10\%/20\%$ training / validation / test split, stratified by class label, yielding approximately 14,000 / 2,000 / 4,000 traces per benchmark. The three subsets are disjoint at the trace level; no trace appears in more than one subset, and a separate model checkpoint is trained per benchmark. The dormant detection accuracy is computed on the held-out $20\%$ test split, and the active accuracy is obtained analogously by replacing the positive class with *TrojanTriggered*. This is the same intra-benchmark evaluation protocol adopted by [3] and [4]; cross-benchmark transfer is left to future work.

The model is trained end-to-end using cross-entropy loss with label smoothing ($\alpha = 0.02$), AdamW optimizer (lr=$10^{-3}$) with cosine annealing schedule and linear warmup for the first 2 epochs, batch size 128, and early stopping with patience of 5 epochs based on validation accuracy.

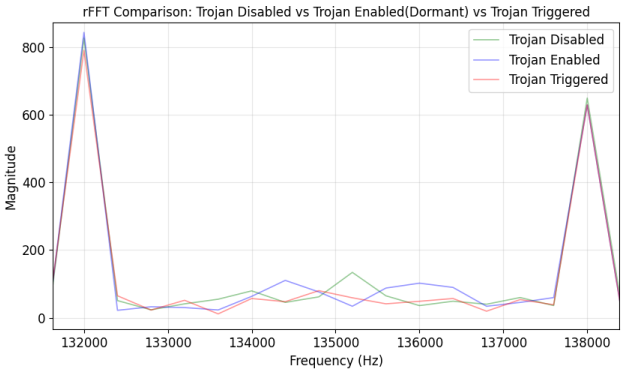


Fig. 2. rFFT magnitude spectrum of AES-T400 in baseline, dormant, and active modes, zoomed to the band where the three modes differ most (dataset-native frequency units; see Sec. IV).

## V. Experimental Results

### A. Frequency-Domain Signatures of HTs

Fig. 2 compares rFFT magnitudes for a representative AES-T400 trace in baseline, dormant, and active modes. Individual bin differences are too subtle to isolate by inspection. However, they become discriminative when aggregated over adjacent non overlapping frequency patches with global self attention, a property **PATCH-FFT** exploits.

### B. Performance Analysis

Table II reports detection accuracy across eight key-leakage HTs. **PATCH-FFT** achieves $89.56\%$ dormant and $92.31\%$ active accuracy ($90.94\%$ average). CDMA-modulation Trojans (T700/T800/T1000) include free-running counter and PN-sequence logic on the AES clock. This logic switches even when the payload trigger has not fired, producing sharp spectral peaks that the model detects with near-perfect accuracy. RF-based Trojans (T400/T1600/T1700) produce moderate spectral artifacts. Leakage-current Trojans (T600/T2000) create only broadband sub-threshold perturbations, making them the hardest variants to separate from PV noise.

### C. Process-Variation Robustness

Because the dataset contains a single FPGA instance, we synthesize PV replicas by perturbing each trace as

$$x'[n] = (1 + \alpha)\, x[n] + \eta_{\mathrm{LP}}[n], \quad \alpha \sim \mathcal{N}(0, \sigma^2), \tag{3}$$

with $\eta_{\mathrm{LP}}$ zero-mean Gaussian noise low-pass filtered to the FPGA core bandwidth and RMS-scaled by $\sigma \cdot \mathrm{RMS}(x)$. The multiplicative

TABLE II
PATCH-FFT ACCURACY ACROSS DIFFERENT AES HARDWARE TROJAN VARIANTS.

| Benchmark | Key-Leakage Mechanism | Native Channel | Dormant (%) | Active (%) |
|---|---|---|---|---|
| AES-T600 | Leakage current modulated by key bits | Leakage current | 77.3 | 83.6 |
| AES-T2000 | Increased leakage current for key bits equal to 0 | Leakage current | 88.4 | 92.1 |
| AES-T700 | CDMA-based modulation of dynamic power after trigger seq. | Dynamic power | 96.7 | 99.2 |
| AES-T800 | CDMA-based modulation of dynamic power after trigger seq. | Dynamic power | 100.0 | 100.0 |
| AES-T1000 | Increased dynamic power after trigger sequence | Dynamic power | 95.5 | 97.8 |
| AES-T400 | Key bits transmitted via modulated RF signal on unused pin | RF / EM emission | 89.2 | 92.4 |
| AES-T1600 | Key bits transmitted via RF signal after trigger sequence | RF / EM emission | 82.8 | 84.3 |
| AES-T1700 | Key bits transmitted via RF signal driven by counter | RF / EM emission | 86.6 | 89.1 |

TABLE III
PATCH-FFT DORMANT-STATE ACCURACY UNDER SYNTHETIC PROCESS VARIATION (MEAN OVER EIGHT BENCHMARKS, TEN SEEDS).

| $\sigma$ (% RMS) | Dormant Acc. (%) | $\Delta$ vs. clean |
|---|---|---|
| 0 | 89.56 | – |
| 5 | 88.71 | $-0.85$ |
| 10 | 86.94 | $-2.62$ |
| 15 | 84.10 | $-5.46$ |
| 20 | 80.42 | $-9.14$ |

TABLE IV
PATCH-FFT PERFORMANCE AGAINST OTHER MODELS.

| Method | Accuracy (%) |
|---|---|
| HTM [4] | 92.2 (triggered only) |
| Siamese LSTM [3] | 86.78 (triggered only) |
| **PATCH-FFT (dormant)** | **89.56** |
| **PATCH-FFT (triggered)** | **92.31** |
| **PATCH-FFT (average)** | **90.94** |

term captures die-to-die $V_{th}/I_{dsat}$ shifts. The band-limited additive term emulates correlated within-die variability [34]–[36]. Training data is clean; perturbation is applied only to the held-out test split (10 seeds per $\sigma$).

Table III shows that dormant accuracy degrades by $< 1$ pp at $\sigma$=5% and retains $>84\%$ at $\sigma$=15%. An ablation isolating each term reveals that $z$-score normalization absorbs the multiplicative gain. The additive band-limited noise drives nearly all of the degradation. This confirms that **PATCH-FFT**'s spectral-locality property tolerates die-mean shifts.

### D. Comparison with State-of-the-Art Works

Table IV compares **PATCH-FFT** to prior side-channel based HT detectors. Crucially, neither HTM [4] nor Siamese LSTM [3] evaluate or report dormant-state detection accuracy. **PATCH-FFT** achieves 89.56% on the harder dormant case and 92.31% on triggered Trojans, showing that frequency-domain patch-based analysis enables reliable dormant HT detection, a capability not addressed by prior time-domain methods.

### E. Layer-wise Frequency Sensitivity

Fig. 3 shows encoder layer–frequency heatmaps for AES-T600 and AES-T800. For AES-T600, early layers respond broadly while deeper layers concentrate on narrow bands where dormant traces deviate from the baseline. For AES-T800, attention quickly collapses onto harmonics of the CDMA spreading sequence, confirming that the model learns physically meaningful spectral features rather than spurious correlations.

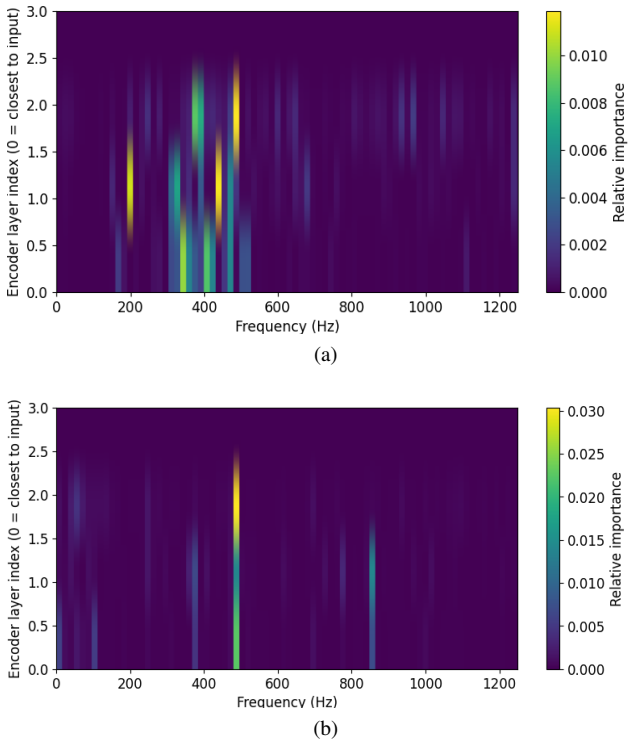


Fig. 3. Attention heatmaps for different Trojan variants: (a) AES-T600 (b) AES-T800.

## VI. CONCLUSION

In this paper, we presented **PATCH-FFT**, a patch-based frequency-domain Transformer for HT detection from power side-channel traces. By operating on rFFT spectra and attending to frequency patches, the model achieves strong accuracy not only in detecting active Trojans but also in identifying dormant Trojans, a capability that prior dynamic-power ML methods do not address. Modulation-based HTs that produce sharp spectral signatures are detected with near-perfect accuracy, while subtler leakage-current variants remain more challenging.

**PATCH-FFT** achieves 89.56% dormant and 92.31% active accuracy (90.94% average) and remains stable under synthetic process-variation augmentation. Future work will focus on improving detection of leakage-current Trojans through time- and frequency-domain feature fusion, evaluating cross-benchmark transfer, validating the approach on EM traces, and extending evaluation to fabricated ASIC silicon.

## REFERENCES


[1] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," In *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 10–25, 2010.

[2] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware Trojans," In *IEEE Computer*, vol. 43, no. 10, pp. 39–46, 2010.

[3] A. Nasr, K. Mohamed, A. Elshenawy, and M. Zaki, "A Siamese deep learning framework for efficient hardware Trojan detection using power side-channel data," In *Scientific Reports*, vol. 14, no. 13013, pp. 1–13, 2024.

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

[5] A. Pungati, M. Jain, and S. Rawat, "Comparative analysis of multiresolution features through power side-channel signals for hardware Trojan detection," In *SSRN Electronic Journal*, 2023.

[6] S. Wei and M. Potkonjak, "Scalable hardware Trojan diagnosis," In *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 6, pp. 1049–1057, 2012.

[7] J. Aarestad, D. Acharyya, R. Rad, and J. Plusquellic, "Detecting Trojans through leakage current analysis using multiple supply pad $I_{DDQ}$s," In *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 893–904, 2010.

[8] M. Lecomte, J. Fournier, and P. Maurine, "An on-chip technique to detect hardware Trojans and assist counterfeit identification," In *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 12, pp. 3317–3330, 2017.

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

[10] T. Perez and S. Pagliarini, "A side-channel hardware Trojan in 65nm CMOS with 2$\mu$W precision and multi-bit leakage capability," In *Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 9–10, 2022.

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

[12] Z. Wang, Y. Xue, and H. Wang, "Secure run-time hardware Trojan detection using lightweight analytical models," In *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 1–9, 2023.

[13] S. Kaji, D. Fujimoto, and Y. Hayashi, "Simulation-based approach to generating golden data for PCB-level hardware Trojan detection using capacitive sensor," In *IEEE Physical Assurance and Inspection of Electronics (PAINE)*, pp. 1–7, 2023.

[14] R. Vishwakarma and A. Rezaei, "Uncertainty-aware hardware Trojan detection using multimodal deep learning," In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, 2024.

[15] R. Vishwakarma and A. Rezaei, "Uncertainty-aware unimodal and multimodal learning for evolving hardware Trojan detection," In *Journal of Hardware and Systems Security*, Vol. 9, pp. 1–23, 2025.

[16] T. Ishikawa, K. Yokooji, Y. Midoh, N. Miura, M. Shintani, and J. Shiomi, "Hardware Trojan detection by fine-grained power domain partitioning," In *Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 1257–1263, 2025.

[17] J. Maynard and A. Rezaei, "Reconfigurable run-time hardware Trojan mitigation for logic-locked circuits," In *IEEE 17th Dallas Circuits and Systems Conference (DCAS)*, pp. 1–6, 2024.

[18] L. Chen, Y. Gamal, Y. Li, S. -Y. Yu, I. Alouani, and M. A. A. Faruque, "DART: Distribution-aware hardware Trojan detection," In *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 9600–9609, 2025.

[19] R. Kumar Kundu, K. Khalil, E. Garcia, E. Grassia, P. Calyam, and K. A. Hoque, "PEARL: An adaptive and explainable hardware Trojan detection using open source and enterprise large language models," In *IEEE Access*, vol. 13, pp. 133755–133772, 2025.

[20] B. Li, C. Dong, D. Qiu, M. Chen, and Y. Yang, "Defense in the reverse fragment: RL-based partial netlist hardware Trojan detection," In *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1–7, 2025.

[21] W. Hu, B. Li, L. Wu, Y. Li, X. Li, and L. Hong, "Design for assurance: Employing functional verification tools for thwarting hardware Trojan threat in 3PIPs," In *IEEE Transactions on Dependable and Secure Computing*, vol. 23, no. 1, pp. 132–148, 2026.

[22] H. Senevirathne, R. Vishwakarma, and A. Rezaei, "Early detection of hardware trojans using neural controlled differential equations and analysis of power traces," In *IEEE 19th Dallas Circuits and Systems Conference (DCAS*, pp. 1–6, 2026.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, "Attention is all you need," In *International Conference on Neural Information Processing Systems (NIPS)*, pp. 6000–6010, 2017.

[24] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," In *arXiv:2010.11929*, 2020.

[25] Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," In *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*, pp. 51–57, 2008.

[26] R. Rad, J. Plusquellic, and M. Tehranipoor, "A sensitivity analysis of power signal methods for detecting hardware Trojans under real process and environmental conditions," In *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 12, pp. 1735–1744, 2009.

[27] K. Hasegawa, M. Oya, M. Yanagisawa, and N. Togawa, "Hardware Trojans classification for gate-level netlists using multi-layer neural networks," In *IEEE International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pp. 227–232, 2017.

[28] R. Lu, H. Shen, Y. Su, H. Li, and X. Li, "GramsDet: Hardware Trojan detection based on recurrent neural network," In *IEEE Asian Test Symposium (ATS)*, pp. 111–116, 2019.

[29] E. Mateos and C. H. Gebotys, "A new correlation frequency analysis of the side channel," In *Proceedings of theWorkshop on Embedded Systems Security (WESS)*, article 4, pp. 1–5, 2010.

[30] C. He, B. Hou, L. Wang, Y. En, and S. Xie, "A novel hardware Trojan detection method based on side-channel analysis and PCA algorithm," In *International Conference on Reliability, Maintainability and Safety (ICRMS)*, pp. 1043–1046, 2014.

[31] Y. Nie, N. H. Nguyen, P. Sinhamahapatra, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," In *International Conference on Learning Representations (ICLR)*, 2023.

[32] R. Yasaei, S. Faezi, and M. A. Al Faruque, "Hardware Trojan Power & EM Side-Channel dataset," In *IEEE Dataport*, 2021. [Online]. Available: https://dx.doi.org/10.21227/9fwb-8978

[33] H. Guntur, J. Ishii, and A. Satoh, "Side-channel attack user reference architecture board SAKURA-G," In *IEEE Global Conference on Consumer Electronics (GCCE)*, pp. 271–274, 2014.

[34] S. Sun, H. Zhang, X. Cui, L. Dong, and X. Fang, "Electromagnetic side-channel hardware Trojan detection based on transfer learning," In *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1742–1746, 2022.

[35] S. Yuan, K. Wang, M. Chen, B. Yu, and Y. Liu, "A process variation-resistant, golden-free hardware Trojan detection method based on side-channel analysis," In *Security and Communication Networks*, vol. 2021, Article 8839222, 2021.

[36] K. Bernstein et al., "High-performance CMOS variability in the 65-nm regime and beyond," In *IBM Journal of Research and Development*, vol. 50, no. 4/5, pp. 433–449, 2006.