IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

EM Side-Channel Based Hardware Trojan Detection Using Deep SVDD for Unsupervised Anomaly Identification

Vasikarla Renuka[1], T. Swathi[2], Megavath Ravindar[3], Ramavath Sharath[4]
Assistant Professor, Dept. of ECE[1,2]
Dept. of ECE[3,4]
Brilliant Grammar School Educational Society's Group of Institutions - Integrated Campus, Hyderabad, India.
vasikarlarenuka1@gmail.com, Swathi.tulam888@gmail.com
megavathravindar2002@gmail.com, rickeysharath3@gmail.com

Abstract: *The globalization of integrated-circuit design and fabrication has increased the risk of malicious hardware modifications. This paper presents a non-invasive electromagnetic side-channel framework for hardware Trojan detection using Deep Support Vector Data Description (Deep SVDD). Electromagnetic emissions from a device under test are captured with a near-field probe, conditioned, digitized, and transformed into the frequency domain using a Hann window and Fast Fourier Transform. The magnitude-squared spectrum is computed, and dominant peaks near the operating clock and its harmonics are selected to form a compact feature vector. Only Trojan-free samples are used during offline training. Deep SVDD learns a compact latent representation of normal behavior by minimizing the distance of mapped feature vectors from a hypersphere center. During online operation, the anomaly score of a new sample is compared with a learned threshold to classify the device as normal or suspicious. MATLAB is used for dataset preparation and model training, while the online signal-processing and scoring pipeline is structured for Verilog implementation. The proposed approach does not require labeled Trojan examples, supports the detection of previously unseen modifications, and offers a practical path toward real-time trust verification of integrated circuits.*

Keywords: hardware Trojan, electromagnetic side channel, Deep SVDD, unsupervised anomaly detection, FFT, spectral features, FPGA, Verilog.

I. INTRODUCTION

Modern electronic systems increasingly depend on third-party intellectual-property cores, outsourced fabrication, and globally distributed supply chains. These practices reduce cost and development time but create opportunities for unauthorized modifications to be inserted during design, integration, or fabrication.[1]-[8]

A hardware Trojan is a deliberate modification that changes functionality, leaks information, degrades reliability, or creates hidden access under specific trigger conditions. Trojans are usually designed to remain inactive during standard production testing, which makes exhaustive functional detection impractical.[9]-[17]

Side-channel analysis detects changes in physical behavior rather than relying only on logical outputs. Power, delay, and electromagnetic signatures have been investigated for this purpose.[18],[19],[20]. Among them, electromagnetic analysis is attractive because it is non-invasive and can provide localized information about switching activity.[21]

Most learning-based Trojan detectors are supervised and require representative normal and Trojan samples. In practice, the number of possible Trojan structures is extremely large and previously unseen threats are unavoidable.[22],[23] A one-class framework that learns only normal behavior is therefore better aligned with realistic deployment conditions.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

This work combines frequency-domain EM analysis with Deep SVDD. The signal-processing stage converts noisy time-domain measurements into compact spectral features, while the learning stage models the normal feature distribution within a hypersphere. Any sufficiently large deviation from this learned region is treated as anomalous.[24]

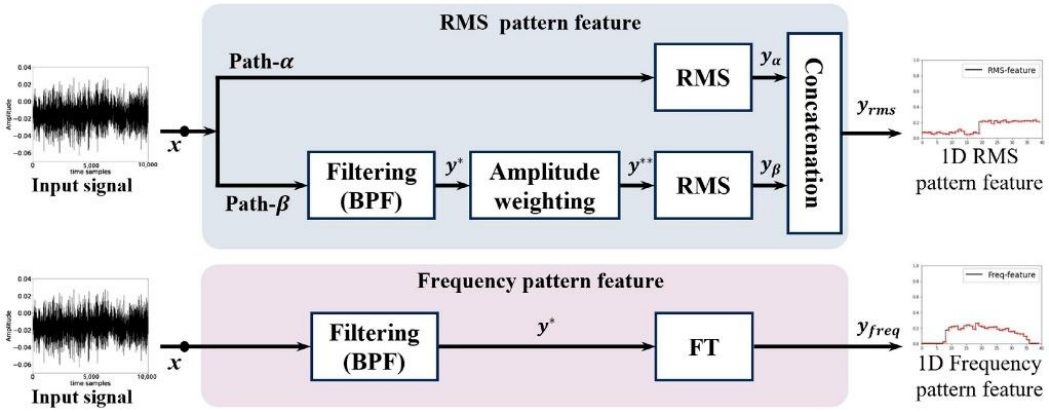

Fig. 1. EM side-channel feature extraction using RMS and frequency-domain patterns.

II. RELATED WORK

2.1 Conventional Hardware Trojan Detection

Traditional methods include functional testing, scan-based analysis, formal verification, reverse engineering, and destructive inspection. Functional and scan tests are limited by rare Trojan triggers, while destructive analysis is costly and unsuitable for large-scale screening.

2.2 Side-Channel-Based Detection

Power analysis measures changes in aggregate current consumption, and timing analysis evaluates delay deviations. Both can be affected by process, voltage, temperature, and measurement noise. EM analysis provides improved spatial sensitivity and can reveal localized switching changes introduced by small Trojan circuits.

2.3 Machine Learning for Trojan Detection

Support vector machines, neural networks, and ensemble classifiers have been used with side-channel features. These methods achieve strong performance for known Trojan classes but depend on labeled attack data and may generalize poorly to unseen modifications.

2.4 One-Class and Unsupervised Anomaly Detection

One-Class SVM and autoencoders model normal behavior without explicit anomaly labels. One-Class SVM can be sensitive to kernel and hyperparameter selection, while autoencoders may reconstruct some anomalies well. Deep SVDD directly optimizes a compact latent representation and provides a simple distance-based anomaly score.

2.5 Research Gap

Existing work frequently relies on golden chips, labeled Trojan datasets, high-dimensional raw traces, or computationally expensive classifiers. A compact EM spectral feature pipeline combined with one-class deep learning and a hardware-friendly inference mechanism remains an important practical need.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

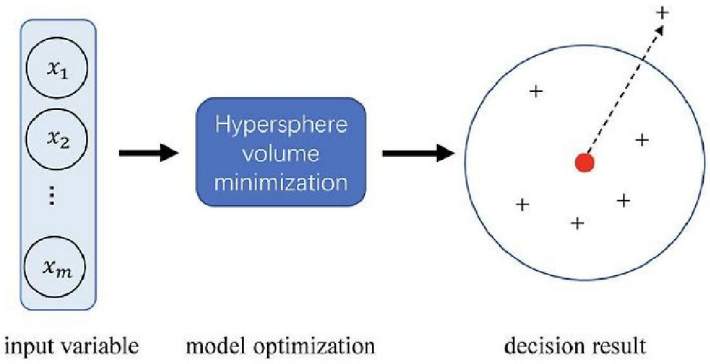

Fig. 2. Deep SVDD concept: normal samples are enclosed in a compact hypersphere.

III. METHODOLOGY

3.1 Proposed System Overview

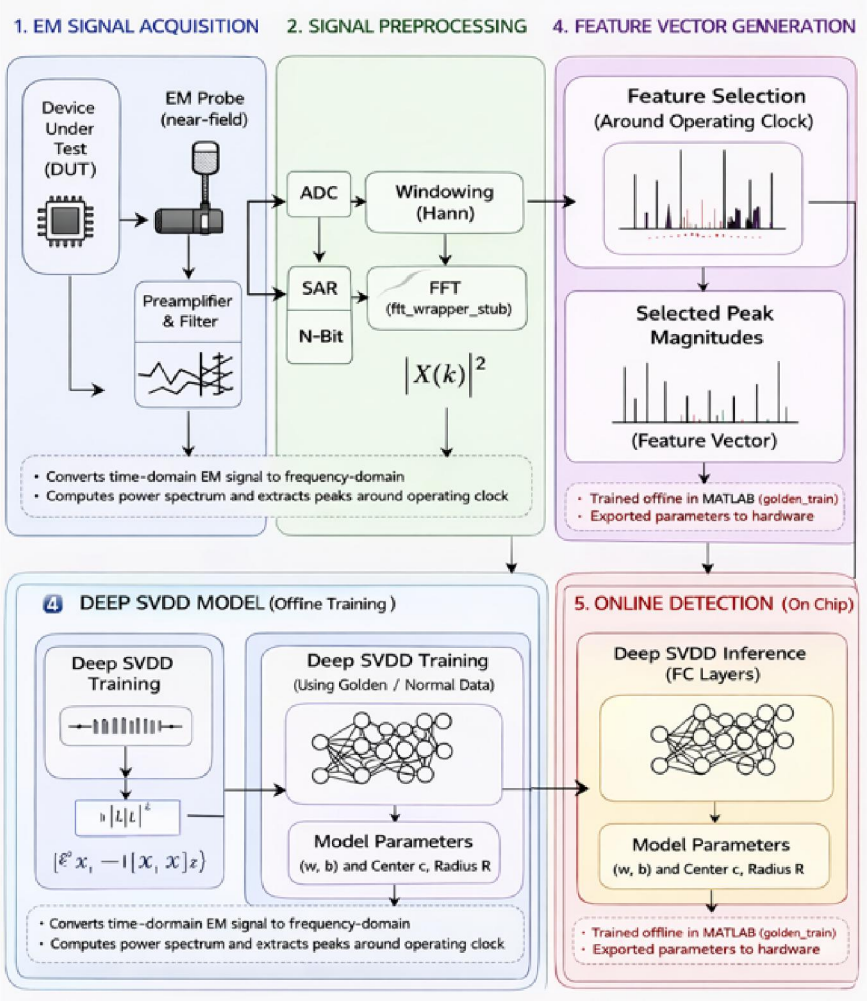

Fig. 3. Proposed EM-based hardware Trojan detection framework.

DOI: 10.48175/IJARSCT-38083

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

The proposed framework is divided into five stages: EM signal acquisition, preprocessing, feature-vector generation, offline Deep SVDD training, and online anomaly detection. The offline stage is executed in MATLAB using only golden samples. The trained network parameters, center vector, and anomaly threshold are exported to the online hardware-oriented pipeline.

3.2 EM Signal Acquisition

The device under test is operated with a repeatable workload and stable clock. A near-field magnetic or electric probe is positioned over the selected region of the device. The weak probe signal is amplified and filtered before digitization. Consistent probe placement, shielding, gain, sampling rate, and operating conditions are essential for repeatable measurements.

$$x[n] = s[n] + v[n]$$

Here, s[n] is the device-dependent EM emission and v[n] represents measurement noise and environmental interference.

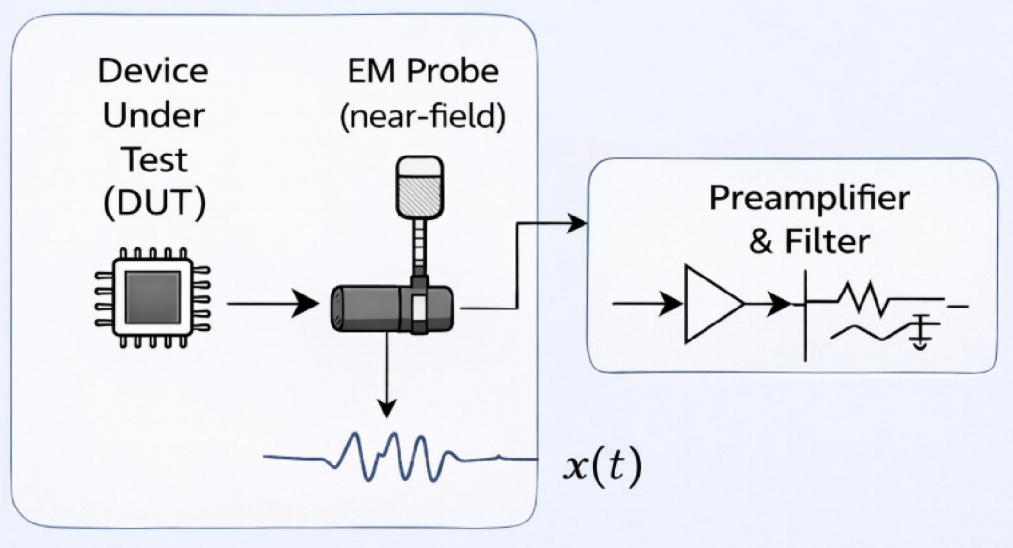

Fig. 4. EM signal acquisition using a near-field probe and preamplifier/filter.

3.3 Windowing and Spectral Transformation

A finite data frame is multiplied by a Hann window to reduce spectral leakage before applying the N-point FFT.

$$w[n] = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right), 0 \le n \le N-1$$

$$X[k] = \sum_{n=0}^{N-1} x[n]\, w[n]\, e^{-j\frac{2\pi kn}{N}}, 0 \le k \le N-1$$

The power spectrum is obtained from the squared magnitude of the complex FFT output.

$$P[k] = |X[k]|^2 = (Re\{X[k]\})^2 + (Im\{X[k]\})^2$$

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

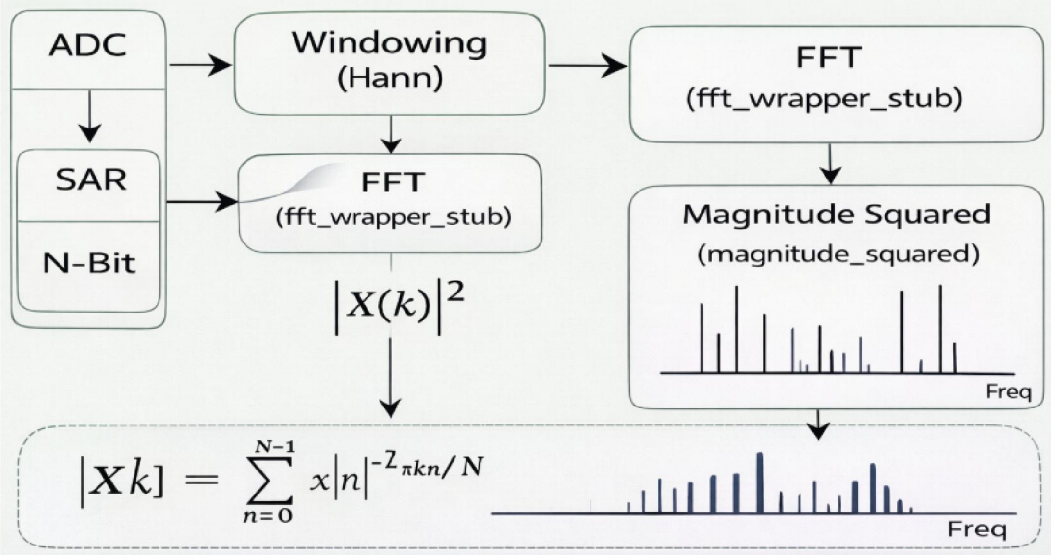

Fig. 5. Signal preprocessing with ADC, Hann window, FFT, and magnitude-squared computation.

3.4 Spectral Feature Extraction

The complete spectrum is high-dimensional and contains many noise-dominated bins. The proposed system selects the strongest components in a frequency band around the operating clock and selected harmonics. The ordered peak magnitudes form the feature vector.

$$\boldsymbol{f} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix} = [p_1, p_2, \ldots, p_m]^T$$

Features may be normalized using statistics obtained from the training set to reduce sensitivity to absolute probe gain.

$$\tilde{f}_i = \frac{f_i - \mu_i}{\sigma_i + \varepsilon}$$

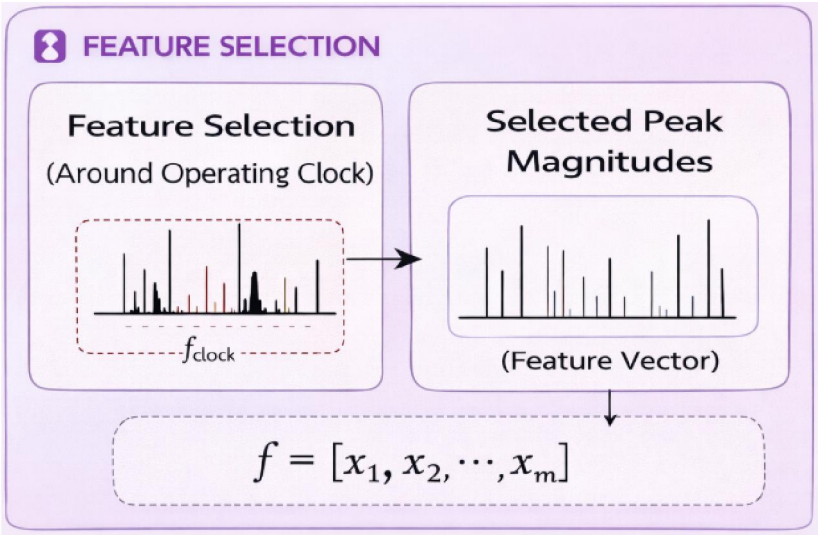

Fig. 6. Selection of dominant spectral peaks for compact feature-vector generation.

DOI: 10.48175/IJARSCT-38083

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

3.5 Deep SVDD Training

A feed-forward neural network φ (·; W) maps the normalized spectral feature vector into a lower-dimensional latent space. Only golden samples are used for training. The center c is initialized from the mean of the initial latent representations, and the network parameters are optimized to minimize the average squared distance to this center.

$$\mathcal{L}(W) = \frac{1}{N}\sum_{i=1}^{N} \| \phi ,(\tilde{\mathbf{f}}_i; W) - \mathbf{c} \|_2^2 + \lambda \| W \|_2^2$$

This objective encourages normal samples to occupy a compact region. The decision radius or threshold is selected from the upper quantile of training or validation anomaly scores.

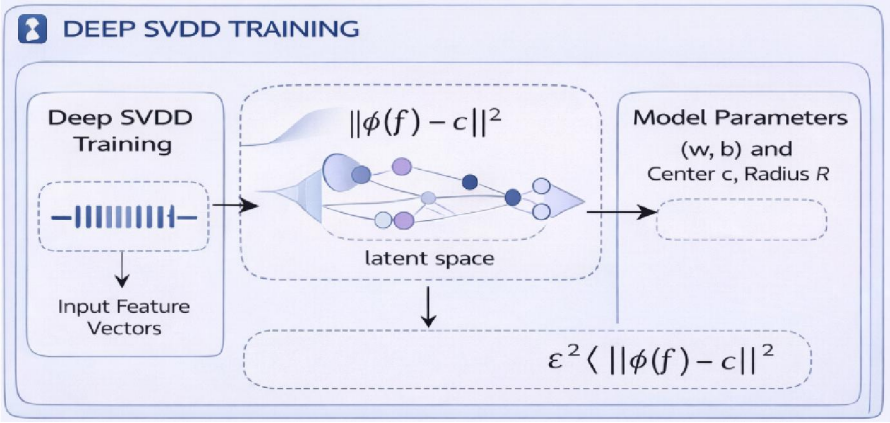

Fig. 7. Offline Deep SVDD training using only golden feature vectors.

3.6 Online Inference and Decision

A new measurement is processed using the same acquisition, normalization, FFT, and peak-selection pipeline. The trained network produces a latent representation, and the squared distance to the center is used as the anomaly score.

$$A(\tilde{\boldsymbol{f}}) = \| \phi \square(\tilde{\boldsymbol{f}}; W^*) - \boldsymbol{c} \|_2^2$$

$$Decision = \begin{cases} Normal, & A(\tilde{\boldsymbol{f}}) \le \tau, \\ Trojan/Suspicious, & A(\tilde{\boldsymbol{f}}) > \tau, \end{cases}$$

The inference path is hardware-friendly because it requires fixed-point fully connected layers, activation functions, subtraction from the center, squaring, accumulation, and threshold comparison.

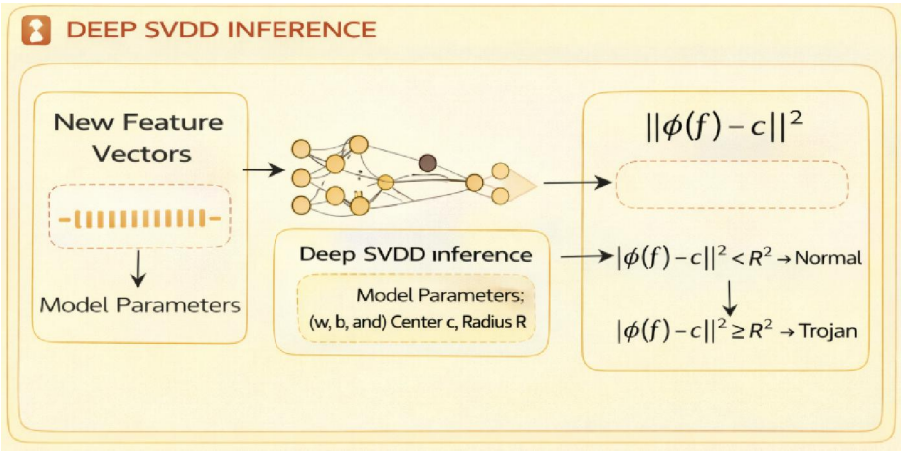

Fig. 8. Deep SVDD inference and threshold-based anomaly decision.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

3.7 MATLAB-to-Verilog Implementation Flow

Collect and organize golden_train, golden_test, and trojan_test datasets.
Generate windowed power spectra and reduced feature vectors in MATLAB.
Train the Deep SVDD model using only golden_train samples.
Determine normalization parameters, latent center c, and threshold τ.
Quantize network weights, biases, center values, and threshold for fixed-point hardware.
Implement FFT interface, magnitude-squared block, peak extractor, inference layers, anomaly scorer, and top-level detector in Verilog.
Verify the complete pipeline using a testbench and compare fixed-point outputs with MATLAB reference values.

3.8 Evaluation Metrics

Performance should be reported using accuracy, precision, recall, F1-score, true-positive rate, false-positive rate, area under the ROC curve, inference latency, throughput, resource utilization, and power. Because the model is trained only on normal data, threshold sensitivity and false-alarm behavior under environmental variation are particularly important.

IV. RESULTS AND DISCUSSION

The proposed hardware-Trojan detection framework was evaluated through FPGA synthesis, behavioral simulation, and MATLAB-based unsupervised anomaly analysis. The supplied outputs verify the complete processing flow from frequency-domain EM feature acquisition to magnitude computation, peak extraction, anomaly scoring, and final Trojan decision. The Deep SVDD-like model was trained using only golden-device observations, while Trojan-contaminated samples were treated as previously unseen anomalies. The results show clear separation between normal and Trojan classes in both autoencoder reconstruction error and hypersphere-distance scores.

4.1. FPGA Device Mapping

The synthesized design was mapped successfully to the Xilinx Spartan-7 XC7S100-FGGA676-2 FPGA. The device view displays the clock regions X0Y0 through X1Y3 and confirms that the proposed detector can be implemented within the selected programmable fabric. The design occupies only a small portion of the available device, leaving sufficient capacity for the FFT front end, data-acquisition interface, multiple sensing channels, or additional classification logic.

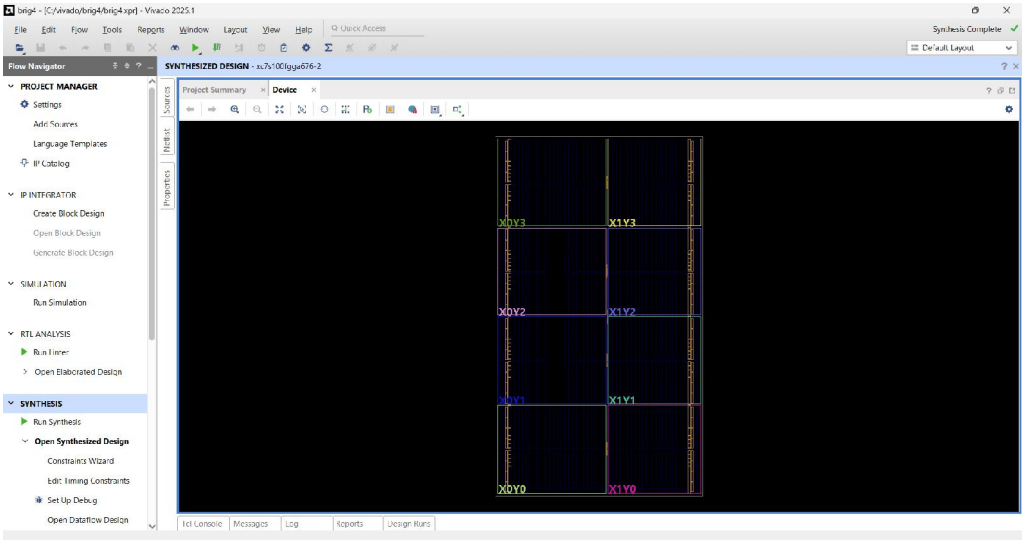

Figure 9. Synthesized FPGA device view for the EM side-channel Trojan detector.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

4.2. Power and Thermal Analysis

Vivado estimates the total on-chip power as 0.089 W. The report attributes 0.089 W, or 100%, to device-static power and reports 0.000 W dynamic power. The junction temperature is 25.2 °C at an ambient temperature of 25.0 °C. A thermal margin of 59.8 °C and an effective junction-to-ambient thermal resistance of 2.4 °C/W indicate safe thermal operation under the stated assumptions.

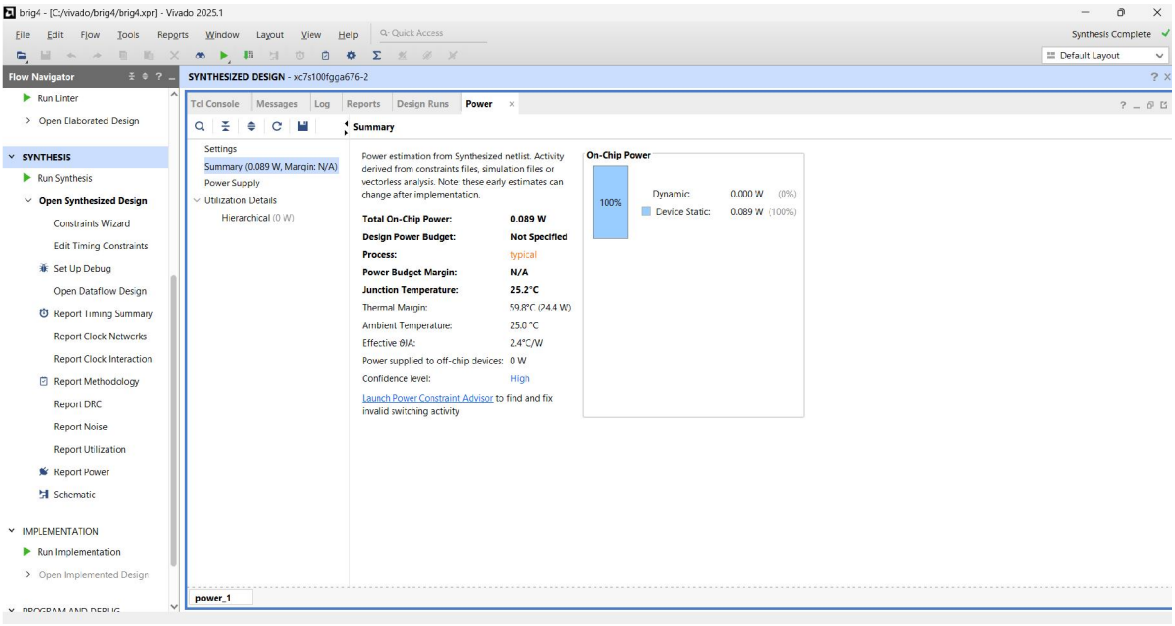

Figure 10. Synthesized-netlist power estimation report.

The reported zero dynamic power does not imply that active FFT and anomaly-scoring operations consume no switching energy. It indicates that the synthesized-netlist report was produced without realistic clock and signal transition activity. Therefore, 0.089 W should be treated as a static baseline. Post-route power analysis using SAIF or VCD activity generated from the testbench is required for an accurate active-power value.

4.3. RTL Architecture Verification

The elaborated schematic reports nine major cells and 393 nets. The input signals include the real and imaginary FFT components, FFT-bin index, frame-control signals, validity signals, clock, and reset. Two range-comparison paths determine whether each FFT bin lies inside the selected EM frequency band. The accepted complex FFT samples are then processed by the magnitude-squared unit. The peak extractor retains the most significant spectral peaks for the current frame, and the anomaly scorer converts the resulting feature vector into an anomaly score and a binary Trojan decision.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

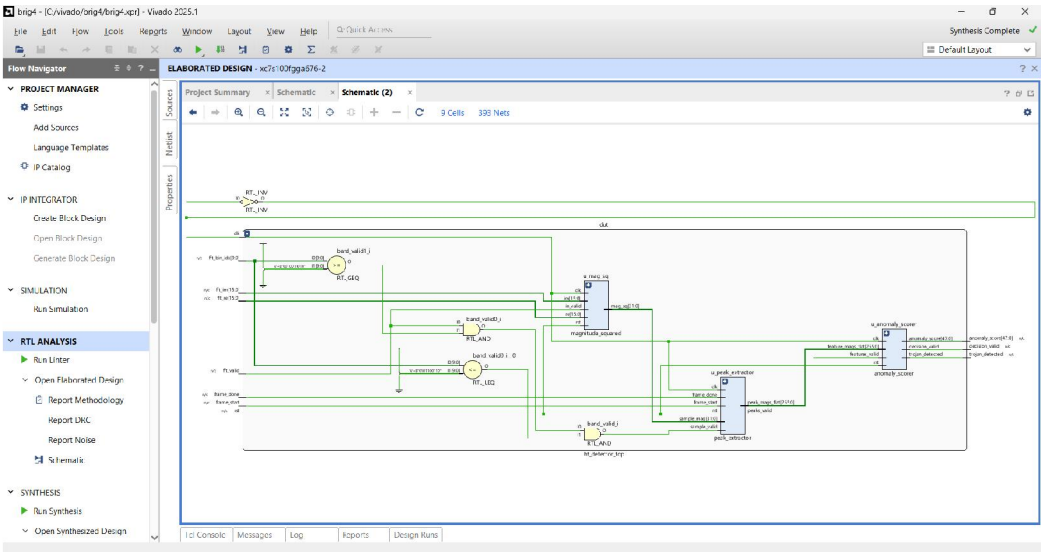

Figure 11. Elaborated RTL architecture of the EM side-channel detector.

For each accepted complex spectral sample, the energy feature is calculated as:

$$M[k] = (\mathrm{Re}\{X[k]\})^2 + (\mathrm{Im}\{X[k]\})^2$$

The magnitude-squared operation avoids the square-root circuit required for exact magnitude computation and therefore reduces FPGA resource usage. The peak-extraction block forms a compact feature representation from the strongest EM spectral components. The anomaly scorer compares this representation with the learned golden-device distribution. A score exceeding the selected threshold activates the Trojan-detected output.

4.4. Synthesis Status and Design Integrity

The synthesis process completed successfully with zero errors and zero critical warnings. Twelve ordinary warnings were reported. The synthesis report was written in approximately 19 seconds, the complete synthesis run required approximately 21 seconds, and peak memory usage was approximately 1.79 GB. These results confirm that the HDL architecture is synthesizable for the selected Spartan-7 FPGA.

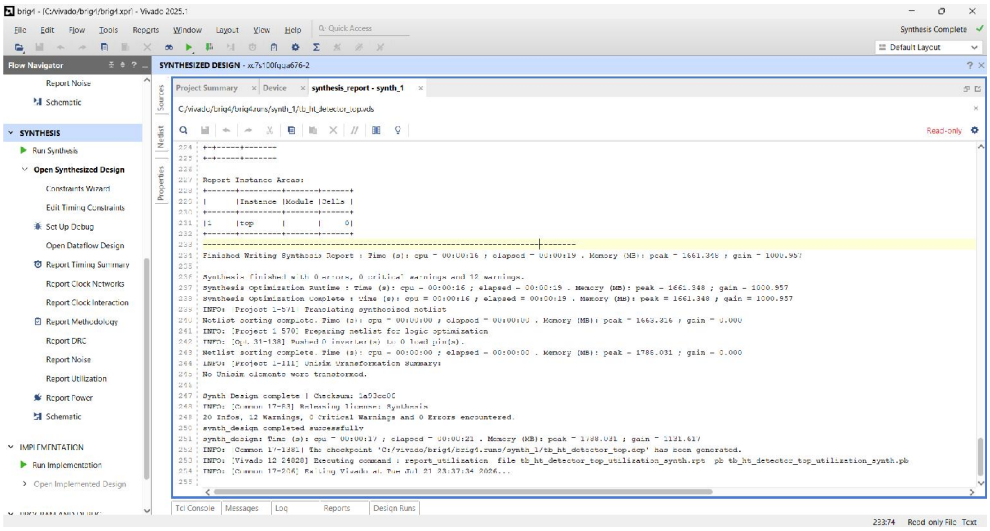

Figure 12. Vivado synthesis completion report.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, August 2026 Impact Factor: 8.2

The displayed report path contains the testbench name `tb_ht_detector_top`, and the visible instance-area entry reports zero cells. This suggests that the simulation wrapper may have been selected as the synthesis top or that non-observable testbench logic was optimized away. The elaborated schematic nevertheless confirms the detector hierarchy. For the final implementation report, `ht_detector_top` should be explicitly selected as the synthesis top, followed by utilization, timing, and post-route power analysis.

4.5. Behavioral Simulation and Trojan Decision

The behavioral waveform verifies end-to-end processing of the FFT-domain EM samples. The simulation uses an FFT size of 1024, a bin index width of 10 bits, 16-bit real and imaginary input data, 32-bit magnitude representation, eight selected peaks, and a 48-bit anomaly score. At the displayed cursor, the FFT real and imaginary components are 0x0014 and 0x0005, respectively, while the bin index reaches 0x1FF. The Trojan output is asserted, demonstrating that the computed anomaly score has exceeded the decision threshold.

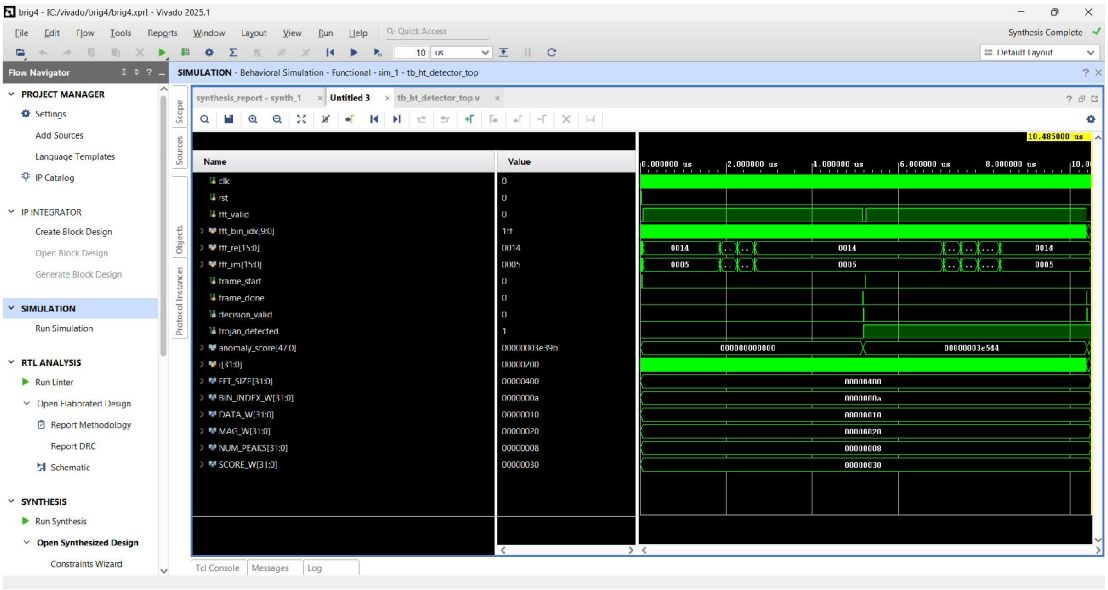

Figure 13. Behavioral simulation waveform showing anomaly scoring and Trojan detection.

The score waveform changes from a near-zero baseline to a large positive value during the processed frame, after which `trojan_detected` remains asserted. This behavior is consistent with an anomalous EM spectrum containing spectral peaks or feature relationships that differ substantially from the golden training distribution.

4.6. Golden-versus-Trojan Score Separation

The boxplots compare the autoencoder mean absolute reconstruction error and the Deep SVDD-like distance for golden and Trojan test samples. Golden samples exhibit a compact reconstruction-error distribution with a median near 0.65 and most values below the red decision threshold of approximately 0.83. Trojan samples have a markedly higher median, approximately 1.35, and a substantially wider distribution extending above 2.0. Although the lowest Trojan values approach or cross the autoencoder threshold, the overall class separation is clear.

The Deep SVDD-like score provides much stronger separation. Golden observations remain close to the learned center, generally around 0-2 distance units on the displayed scale, whereas Trojan observations cluster around approximately 44-52, with isolated values near 36 and 59. This large inter-class margin indicates that hypersphere-distance scoring is more discriminative than reconstruction error for the available EM side-channel features. The practically non-overlapping score distributions suggest high detection sensitivity and a low false-positive tendency on this test set.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

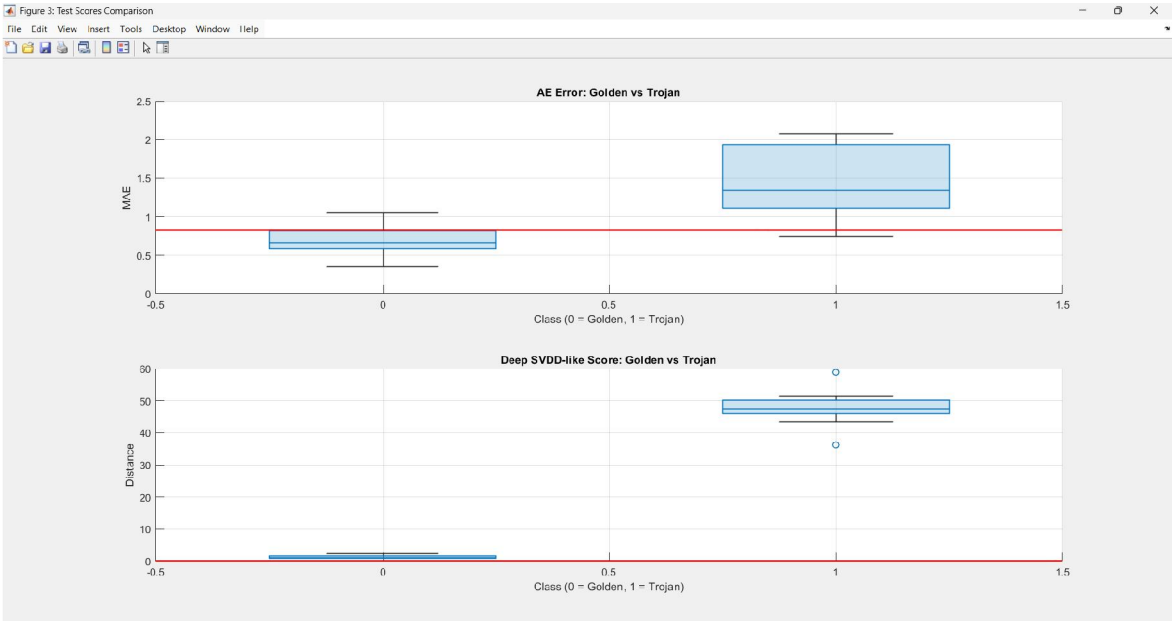

Figure 14. Golden-versus-Trojan comparison using autoencoder error and Deep SVDD-like scores.

4.7. Deep SVDD Training Distribution

The training-score histogram represents distances of golden samples from the learned Deep SVDD center. Most samples are concentrated between approximately 0.5×10^{-3} and 3.0×10^{-3}, with fewer observations extending toward 7.8×10^{-3}. The red vertical line, located near 8.1×10^{-3}, defines the anomaly threshold. All displayed golden training scores lie below this boundary.

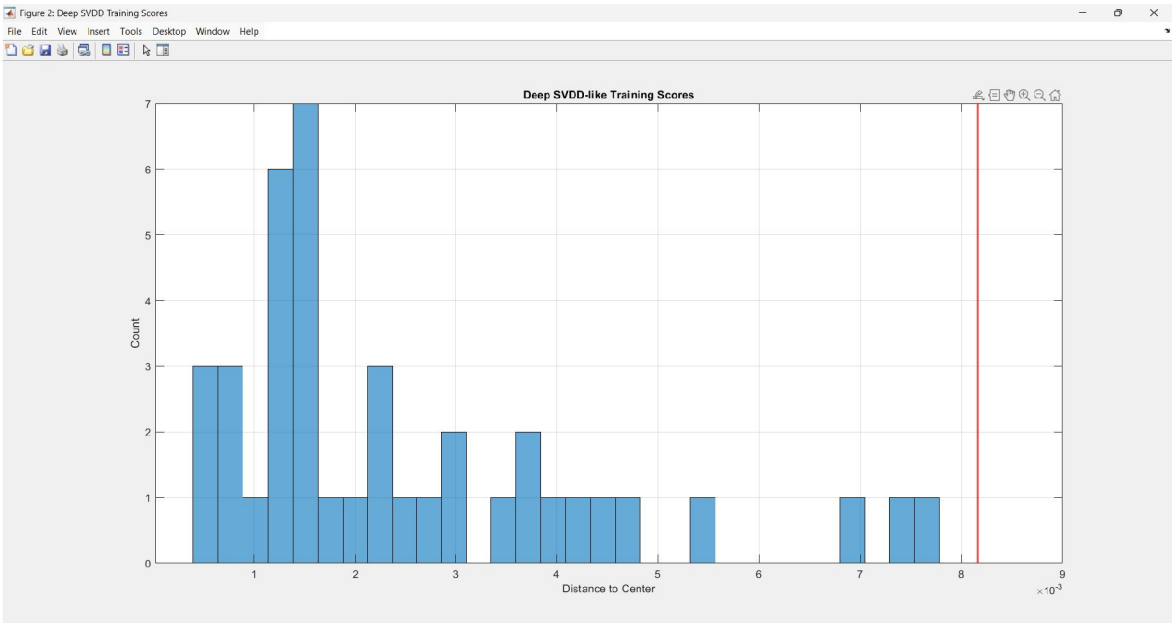

Figure 15. Distribution of Deep SVDD-like training scores and selected threshold.

DOI: 10.48175/IJARSCT-38083

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

The threshold placement beyond the largest observed golden distance creates a protective margin around the nominal feature hypersphere. A new observation is identified as anomalous when its latent-space distance exceeds the threshold. The test comparison demonstrates that Trojan scores are many orders of magnitude larger than the golden training distances shown in this histogram, thereby supporting robust unsupervised anomaly identification.

4.8. Autoencoder Feature-Learning Model

The autoencoder contains a 128-dimensional input layer, a 32-dimensional bottleneck representation, and a 128-dimensional reconstructed output. The encoder learns a compressed latent description of the normal EM feature vector, while the decoder reconstructs the original input. Training only with golden measurements forces the network to represent normal device behavior efficiently. Trojan-induced spectral changes are not represented well and therefore produce larger reconstruction errors or abnormal latent features.

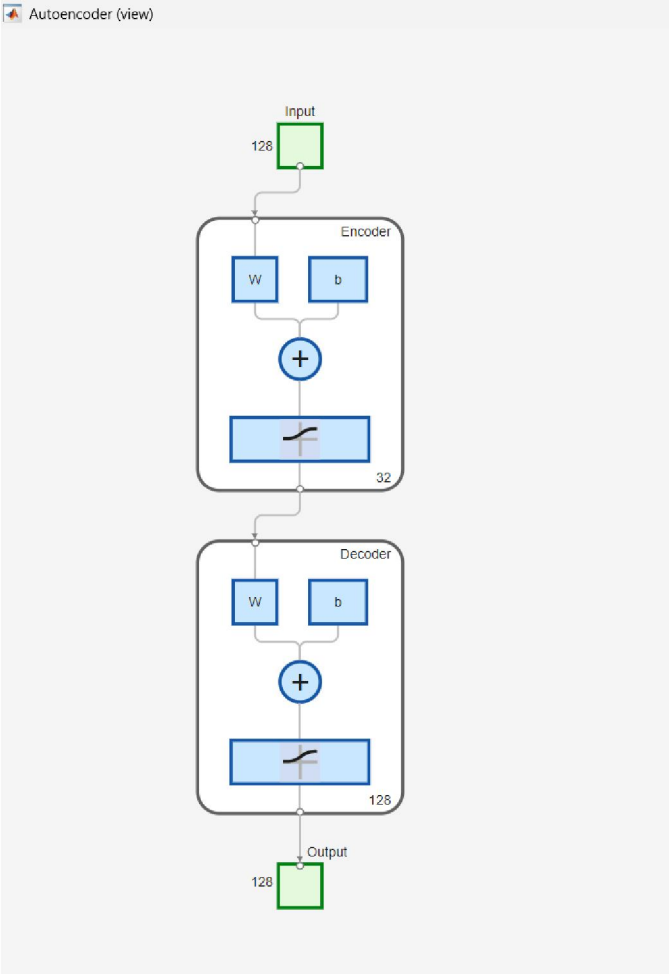

Figure 16. Autoencoder used to learn compact golden-device EM features.

The mean absolute reconstruction error for a feature vector of dimension D is expressed as:

$$\mathbf{MAE} = \frac{1}{D}\sum_{i=1}^{D} | x_i - \hat{x}_i |$$

The Deep SVDD-like anomaly score can be represented as the squared distance from the learned center c:

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

$$s(\mathbf{x}) = \| \boldsymbol{\phi}(\mathbf{x}) - \mathbf{c} \|_2^2$$

The final decision follows:

Trojan = 1, if s(x) > τ; otherwise Trojan = 0

4.9. Autoencoder Training Reconstruction Error

The autoencoder training-error histogram shows that most golden reconstruction errors fall between approximately 0.21 and 0.80 MAE. A smaller number of samples extend toward 1.12. The red threshold is positioned near 0.83, above the majority of the golden training observations. This threshold therefore identifies samples with unusually poor reconstruction as potential anomalies.

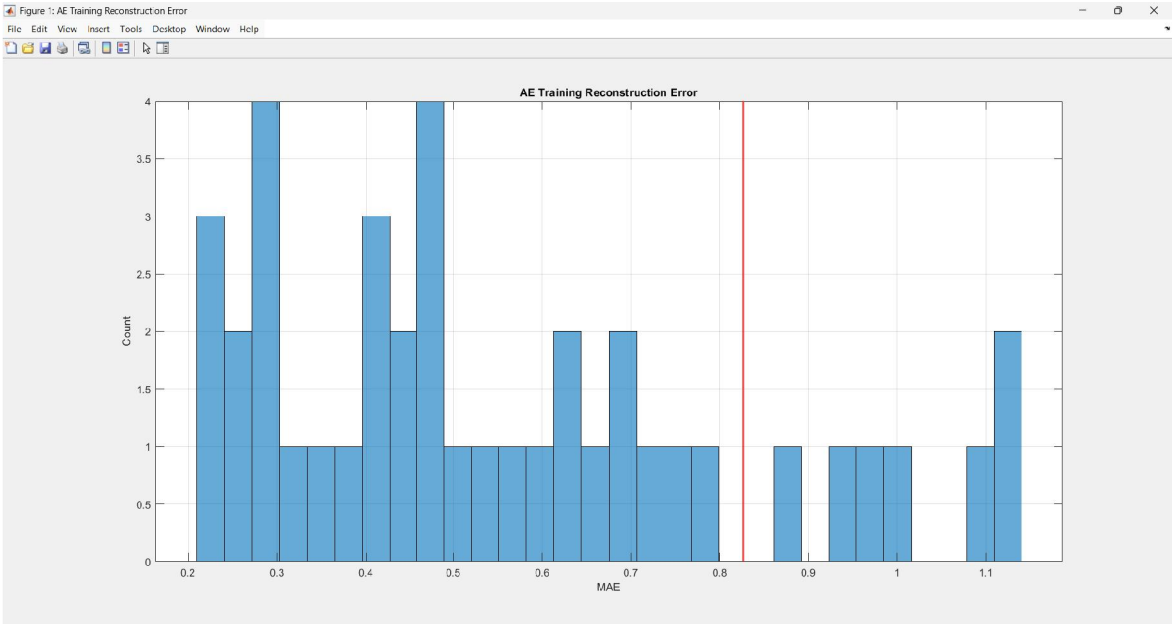

Figure 17. Autoencoder training reconstruction-error distribution and anomaly threshold.

Compared with the Deep SVDD result, the autoencoder error distribution has a longer upper tail, and a few normal training samples occur close to or above the selected boundary. Consequently, reconstruction-error detection may be more sensitive to normal variation and measurement noise. The combined use of autoencoder features and Deep SVDD distance is advantageous: the autoencoder reduces dimensionality and suppresses redundant information, while Deep SVDD provides a compact one-class boundary with stronger golden-versus-Trojan separation.

V. CONCLUSION

This paper presented an electromagnetic side-channel framework for unsupervised hardware Trojan detection using Deep SVDD. The method converts near-field EM measurements into stable frequency-domain representations through windowing, FFT, and magnitude-squared processing. Dominant clock-related spectral peaks provide a compact feature vector that reduces memory and computation. Deep SVDD learns only the normal hardware distribution and detects unknown modifications through distance-based anomaly scoring. The separation of MATLAB-based offline training from Verilog-oriented online inference enables both flexible model development and practical real-time deployment. The proposed system is therefore suitable for production screening, runtime trust monitoring, and golden-chip-limited scenarios. Future work may include multi-probe fusion, adaptive thresholding, domain adaptation across devices, environmental compensation, and FPGA or ASIC prototype validation.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

REFERENCES

[1] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," 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," Computer, vol. 43, no. 10, pp. 39–46, 2010.
[3] R. S. Chakraborty, S. Narasimhan, and S. Bhunia, "Hardware Trojan: Threats and emerging solutions," in Proc. IEEE High Level Design Validation and Test Workshop, 2009, pp. 166–171.
[4] S. Narasimhan, X. Wang, D. Du, R. S. Chakraborty, and S. Bhunia, "Multiple-parameter side-channel analysis for hardware Trojan detection," IEEE Trans. Computers, vol. 62, no. 11, pp. 2183–2195, 2013.
[5] J. Rajendran, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in Proc. DAC, 2012, pp. 83–89.
[6] Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," in Proc. IEEE HOST, 2008, pp. 51–57.
[7] J. He, Y. Zhao, X. Guo, and Y. Jin, "Hardware Trojan detection through EM side-channel analysis," IEEE Trans. VLSI Systems, vol. 25, no. 10, pp. 2939–2948, 2017.
[8] Y. Zheng, S. Bhunia, and S. Yang, "Self-referencing based Trojan detection without golden chip," IEEE Trans. CAD, vol. 35, no. 1, pp. 37–48, 2016.
[9] B. Schölkopf et al., "Support vector method for novelty detection," in Advances in Neural Information Processing Systems, 1999.
[10] L. Ruff et al., "Deep one-class classification," in Proc. ICML, 2018, pp. 4393–4402.
[11] M. A. Kramer, "Autoassociative neural networks," Computers & Chemical Engineering, vol. 16, no. 4, pp. 313–328, 1992.
[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, pp. 533–536, 1986.
[13] S. Sankaran et al., "Deep learning-based approach for hardware Trojan detection," in Proc. IEEE iSES, 2021.
[14] K. G. Liakos et al., "Machine learning for hardware Trojan detection: A review," in Proc. PACET, 2019.
[15] Z. Huang et al., "A survey on machine learning against hardware Trojan attacks," IEEE Access, vol. 8, pp. 10796–10826, 2020.
[16] A. Sumarsono and Z. Masters, "Application of LSTM autoencoder in Trojan detection," in Proc. IEEE CCWC, 2023.
[17] S. Faezi, R. Yasaei, and M. A. Al Faruque, "HTNet: Transfer learning for Trojan detection," in Proc. DATE, 2021.
[18] J. Takahashi et al., "Machine learning-based Trojan detection using EM emanation," IEICE Trans., 2022.
[19] L. N. Nguyen et al., "Backscattering side-channel for Trojan detection," IEEE Trans. VLSI, vol. 27, no. 7, pp. 1561–1574, 2019.
[20] H. Li, Q. Liu, and J. Zhang, "A survey of hardware Trojan threat and defense," Integration, vol. 55, pp. 426–437, 2016.
[21] J. He et al., "Golden chip-free Trojan detection leveraging EM fingerprinting," IEICE Electronics Express, 2019.
[22] K. Xiao et al., "Hardware Trojans: Lessons learned after one decade," ACM TODAES, vol. 22, no. 1, 2016.
[23] S. Wang et al., "Trojan detection based on ELM neural network," in Proc. ICCCI, 2016.
[24] M. Priyatharishini and M. N. Devi, "Deep learning-based malicious module identification," Measurement, vol. 194, 2022