HT-EMIS: A Deep Learning Tool for Hardware Trojan Detection and Identification through Runtime EM Side-Channels

Hanqiu Wang
Department of Electrical and Computer Engineering,
University of Florida,
Gainesville, FL, USA
wanghanqiu@ufl.edu

Maximilian Kealoha Panoff
Department of Electrical and Computer Engineering,
University of Florida,
Gainesville, FL, USA
m.panoff@ufl.edu

Shuo Wang
Department of Electrical and Computer Engineering,
University of Florida,
Gainesville, FL, USA
shuo.wang@ece.ufl.edu

Domenic Forte
Department of Electrical and Computer Engineering,
University of Florida,
Gainesville, FL, USA
dforte@ece.ufl.edu

ABSTRACT

Hardware Trojans (HTs) are malicious circuits planted in Integrated Circuits (ICs). Multiple techniques using Side-Channel signals to detect HTs have been developed over the past decade. However, most of this research focuses on HT detection. Few of them explore the possibility of either identifying different Hardware Trojans implemented inside ICs or detecting inactive HTs. We propose a runtime EM side-channel analysis workflow (HT-EMIS) that uses a convolutional neural network to address the shortcomings above. By analyzing EM side-channel leakage from an FPGA, our tool can identify known types of HTs implemented inside a design and reports whether they are inactive or active with 100% accuracy. Additionally, we are able to successfully detect new unseen HTs with this model in 98.7% of test cases, due to the fact that HTs inserted at the Register Transfer Level with similar triggers and payloads often have similar effects on a floorplan, and thus the EM radiation of a device.

CCS CONCEPTS

• **Security and privacy → Malicious design modifications**.

KEYWORDS

Hardware Trojan, Electromagnetic Side-channel Analysis, Deep Learning

ACM Reference Format:
Hanqiu Wang, Maximilian Kealoha Panoff, Shuo Wang, and Domenic Forte. 2023. HT-EMIS: A Deep Learning Tool for Hardware Trojan Detection and Identification through Runtime EM Side-Channels. In *Proceedings of the Great Lakes Symposium on VLSI 2023 (GLSVLSI '23), June 5–7, 2023, Knoxville, TN, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3583781.3590260

GLSVLSI '23, June 5–7, 2023, Knoxville, TN, USA

ACM ISBN 979-8-4007-0125-2/23/06…$15.00
https://doi.org/10.1145/3583781.3590260

1 INTRODUCTION

Hardware Trojans (HTs), are malicious circuits implanted in targeted integrated circuits (ICs), typically during fabrication or design by an adversary. HTs have been a growing concern in the hardware security community as third-party designs and manufacturing is increasingly used. An HT can be designed with different objects (i.e. *payloads*) such as extracting and leaking confidential information, disabling or paralyzing the system, and improper elevation of user access[1]. HTs can be either always active or only active when certain external or internal criteria (i.e. *a trigger*) occur. Properly designed triggers may allow HTs to avoid detection during testing [17]. An adversary can target different phases in an IC design's life cycle. In this work we examine the case of an adversary releasing 3rd Party Intellectual Property (3PIP) designs at the Register-Transfer Level (RTL) that a victim may include in their design. These are often obfuscated or locked to prevent leakage of the module, but as a result, they may contain code that the end-user is not able to verify manually. The trigger condition may also be rare enough that the Trojans escape activation during simulation, allowing HTs inserted at RTL to sneak onto a physical device. This work focuses on threats from the red encircled area in Figure 1, including RTL code and the netlist generation prior to implementation. However, our work can still detect active HTs when threats are from the orange encircled area which includes all phases in IC design. Simply put, if traces from the golden model are provided, then HT-EMIS is able to identify HTs. If not provided, HT-EMIS is limited to detecting active HTs of known types.

While most previous HT detection techniques have largely been successful, they often only detect if a Trojan is active [2] but not necessarily which Trojan design it is, if known. Such techniques with binary output including only anomaly undetected and anomaly detected leave no space for a flexible threat treatment plan. When threats are detected, the administrators have no options but shut down the whole system because they lack knowledge of the behavior of HT and thus cannot evaluate potential damage. Figure 2 covers the difference between our proposed technique and the standard response up until now. To address these shortcomings, we propose an Electro-Magnetic (EM) side-channel Hardware Trojan Identifying System, *HT-EMIS*. A model trained following

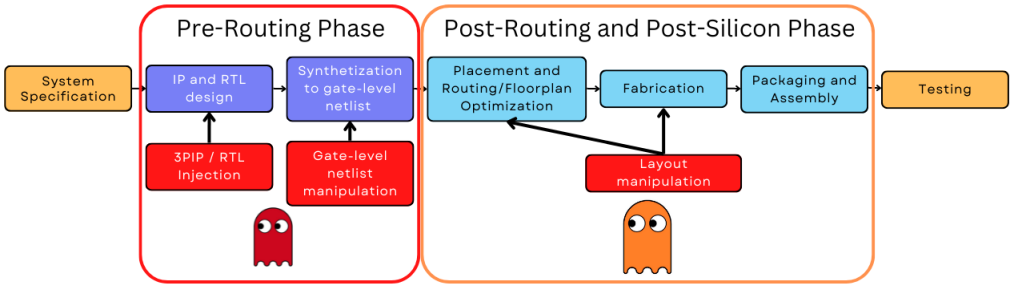

Figure 1: The threat model covered in this work. We are able to detect and identify HTs inserted during the red section, as well as detect active Trojans inserted during the orange.

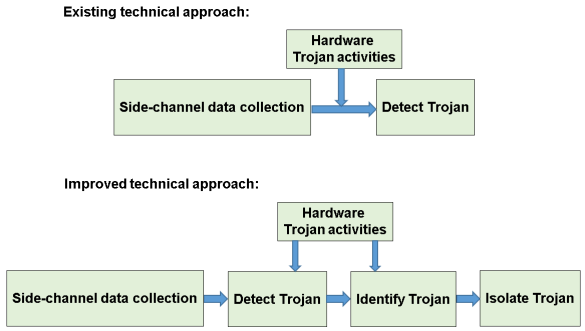

Figure 2: Improved workflow including HT Identification

this process can detect and identify different Hardware Trojans. Note this work assume we have traces from active HT design. Finding input vectors to trigger HTs is not covered by this work. The contributions of our proposed work are as follows:

- A method to train a deep learning model to identify the type and state of a Hardware Trojan from a single trace.
- A method to detect Trojans unseen during training at runtime using the above method.
- An explanation of how unseen HTs may be classified and detected based on their influence of a design's layout.
- A dataset library of EM side-channel measurements collected from a design containing a large number of both active and inactive HTs

The rest of this work is organized as follows: In Section 2 we introduce some important background and prior work in this area. Section 3 contains a detailed explanation of our proposed techniques and Section 4 presents our results, which are in turn discussed in Section 5. Finally, in Section 6 we conclude our work.

2 BACKGROUND

2.1 Hardware Trojan Detection Methods

As mentioned previously, detecting Hardware Trojans (HTs) has seen a large amount of prior interest. The majority of these detection methods require a golden model, (*i.e.* one known to be without HTs), as a reference. However, there are self-referring HT detection methods that use only one DUT. Instead of comparing patterns and features of an HT inserted Integrated Circuit (IC) and an HT-free IC, self-referring methods compare the patterns of the IC when a HT is inserted to determine if it is either active or inactive. He et al [4, 5] proposed a run-time EM side-channel self-referring detection method, in which an on-chip EM sensor is deployed and Euclidean distances of the collected EM traces are calculated and compared when HT is active and inactive. Xue et al [18] focused on the power side-channel and proposed a self-reference-based detection method. Their technique can detect very small HTs with a low Trojan-to-Main circuit area ratio. Hoque et al [6] improved the functional test and gave out a new self-reference method that can detect sequential triggered HT with rare activation events. As a comparison, our tool can work either with or without a golden reference.

Machine learning on Hardware Trojan detection is one trending topic in recent years. Our work also falls into this category. Supervised Learning methods in HT detection include architectures like Artificial Neural Networks (ANN), State Vector Machines (SVM), and K-Nearest Neighbor (KNN) [7]. Unsupervised learning methods examples include K-means clustering and other commonly used clustering methods. Sina et al in proposed a new convolutional neural network architecture that uses transfer learning to detect Trojans in new devices by looking at dominant features [2].

2.2 Hardware Trojan Design and Behaviors

HT threat and application scenarios are constantly evolving. The architecture of HT designs remains as it was a decade ago, consisting of a trigger and a payload [17]. A trigger is what activates a Trojan to deliver a payload. Typically this is a wire signal or a register value in the main circuit. The design goal of an HT trigger is to escape detection during testing while activating when desired. For example, the AES128 Trojan T1400 provided on Trust-hub [11, 12] has a rare sequential triggering condition that has a triggering possibility of only 8.6e-78 in the test phase. Such carefully designed triggers may nearly always escape conventional test vectors.

An example of this can be found in the newly developed analog HT trigger present in the A2 Trojan [19] requires a very small on-chip area and has a very low current profile in addition to its rare triggering condition. These features enable analog HTs to evade

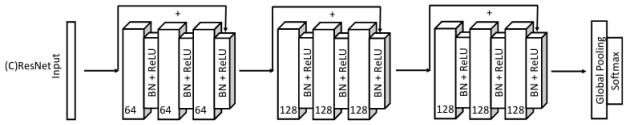

Figure 3: Figure 1C from [16], showing the network architecture. The unlabeled blocks are convolutional layers with number of filters equal to the number specified.

both functional tests and most side-channel analyses. It is difficult to implement analog HTs on FPGAs, so most research on HTs for FPGAs omit analog HTs.

HT payloads, on the other hand, are the components involved in completing a malicious goal, often using the fewest gates and the smallest area possible. These goals range from leaking confidential information to deceiving users by manipulating the output of the circuit. One example of this can simply be the denial of service (DoS) of a circuit, wherein it ceases to function, to a corruption of outputs or information leakage through a side channel. Trust-hub is a publicly available library of HTs that encompasses several types of triggers and payloads such as these [11, 12].

3 METHODOLOGY

In this study, a Convolutional Neural Network (CNN) was utilized to detect and classify hardware Trojans based on the electromagnetic (EM) signals emitted during device operation. A residual network structure, modified to handle 1-dimensional (1D) time series data [16], was employed to analyze EM signals from both compromised and uncompromised designs. Further information can be found in Section 3.1. The input data was classified into one of several classes, as described in Section 3.2. The results demonstrated the ability of the trained models to identify hardware Trojans they have never encountered during training by failing to classify them as belonging to a Trojan-free design.

3.1 Neural Network Architecture

The features of time series data were extracted by using the modified ResNet architecture from [15, 16]. The architecture consisted of three residual blocks, each with three Convolutional layers in 2D, along with batch normalization and raw activation functions. The features were fed into a fully connected layer with a softmax activation function. The detailed architecture is depicted in Figure 3. The choice of this architecture was motivated by the findings from [15, 16] which demonstrated its superiority over other Machine Learning methods for time series data in various tasks. The accuracy of this architecture was compared against other approaches in 4.2, where it was found to be the most accurate for the current task.

3.2 Hardware Trojan Identification

Our dataset consists of three types of EM traces: one from a design without Trojans (Golden Model), one from a design with an inactive AES Trojan, and one from a design with an active AES Trojan. The Golden model is only needed when doing classification, but for simple HT detection, the Golden model is not needed. We classify these traces into different classes based on the HT model and state

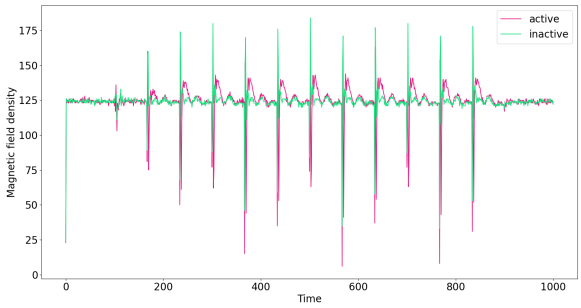

Figure 4: Differences in captured traces from active and inactive traces.

(Active or Inactive). This allows for both identification of the HT in the design and the state of the Trojan. The CNN model, as described in Section 3.1, can be trained using categorical cross-entropy loss to identify different Hardware Trojans. The results are presented in Section 4.2. Our EM trace dataset is obtained by collecting side-channel measurements from AES-128 designs with 16 HTs selected from Trust-hub [11, 12] and one customized HT. These HTs have different triggers and different payloads. The description of triggers and payloads can be found on the Trusthub website. The specifics of the trace collection are discussed in Section 4.1. The results demonstrate clear differences between EM traces of the same HT-infected design based on the state of the HT, as shown in Figure 4.

For Trojans which trigger on a certain input, we make the modification that once triggered the Trojan remains active. This modification was done to minimize the chance that our model overfits to correlate certain plaintext-HT pairs, and instead learns the EM effects of the Trojan directly.

3.3 RTL Trojan's Effects on Design Floorplan

In this work, we have found that inserting hardware Trojans in the Register Transfer Level (RTL) results in similar floorplans and these floorplans are significantly different from the layout of a Trojan-free design. To confirm this, we used a method that converts gate-level netlists with location information into an image and evaluated the similarity through keypoint detection using Scale-Invariant Feature Transform (SIFT)[8]. To generate these images, we assigned a unique color to each type of gate and then mapped each gate's location information to a pixel in an image. Finally, we used SIFT to determine the similarity between each design by detecting keypoints in each image and using a KNN matcher to find matching keypoints. The results of this analysis are presented in a figure.

To generate these images, we first identified all the unique types of gates in a design that contained location information using a Python script. Then, we generated a unique color in Red-Blue-Green (RGB) format for each type of gate. The location information of each gate was extracted and mapped to a pixel in the image, which was filled with the color for that gate.

Finally, we used SIFT to determine the similarity between each design. The OpenCV2's SIFT implementation was used to detect

key points in each image, and each other image was then analyzed to find their key points. Both sets of key points were fed into OpenCV2's default KNN matcher to find the matching key points if any. The number of matches was divided by the higher number of key points in each image set and multiplied by 100 to score the similarity between the two images. The results of this analysis are shown in Figure 5.

3.4 K-fold Verification on New HT Detection

Several factors modify EM leakage from a victim device as shown in equation 1: the location of the measuring EM probe denoted as $P(x, y, z)$, Switching behavior of the ith gate denoted as $S(A, B)$, and the floorplan shape denoted as F, affected by different HT types and sizes inserted. We discuss how HTs with similar triggers and payloads inserted at the RTL have similar effects on the floorplan previously in Section 3.3, and in this section we demonstrate how this can be used to enable models to correctly identify previously unknown HTs.

$$L_E = f\{P(x, y, z), S_i(A, B), F\} \quad (1)$$

To prove this, we perform a modified K-fold cross-validation [10] approach to isolate different HTs. Specifically, we create a 'fold' by splitting our dataset into two portions, one for training and one for testing. The training and testing datasets contain different Trojans, that is to say, no models of Trojans in the training dataset are in the testing set, and *vice versa*. This splitting process is repeated K times. We then train K models on their training sets and evaluate the matching test set. The results of this approach can be found in Section 4.3.

4 EXPERIMENTAL RESULTS

We utilize an automated EM side-channel measurement system, discussed in Section 4.1, to collect data from an AES-128 design [9] implemented containing several HTs on a Spartan-6 FPGA from the SAKURA-G test system [3]. The HT designs come from Trusthub [11, 12]. The specifics of data preparation is discussed in 3.2.

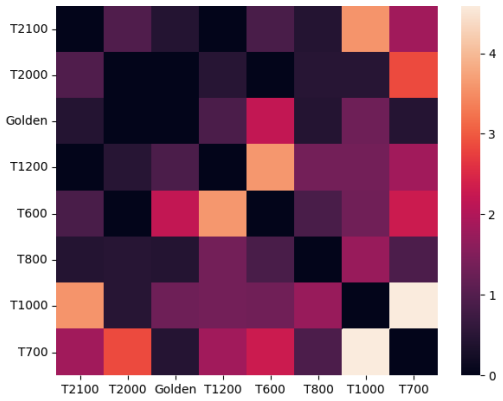

Figure 5: Similarity scores for each design netlist converted into an image. We manually set all designed comparisons to themselves to 0.

4.1 Data Collection

The automated EM side-channel measurement tool is used to collect electromagnetic (EM) side-channel measurements from the AES-128 design on the Device Under Test (DUT) using a Langer EMV LF1 probe. The DUT is the main FPGA chip on the SAKURA-G test board. The measurements are collected while the DUT encrypts random 128-bit plaintext generated by a laptop connected to the DUT via a serial connection. The AES128 encryption RTL used is the AES128 LUT core developed by Satoh Lab. An oscilloscope is used to capture the output of the probe and the traces are stored on the computer for analysis. The setup, including the 3D printer with the extruder replaced by the probe, is shown in Figure 6. The 3D printer is used to keep the probe location at a fixed location relative to the main FPGA on board so that the difference in the EM side-channel traces would not be caused by location changes of the probe. The location is near the center of the floorplan to increase the sensored magnitude of the traces.

Each trace is a 1000 sample recording of the magnetic field covering one complete encryption on the DUT. Each sample is an 8 bit unsigned value with 0 denoting -5mV and 255 denoting +5mV. The sampling rate is 5Gsa/s. Finally, the dataset is normalized such that the maximum of the dataset has the value 1 and the minimum has value 0. In the training dataset, Each Trojan/state combination consists of 1000 such traces, and the Golden Model has 4000 traces.

4.2 Deep Learning Model Selection and Algorithms

We begin by evaluating the success of various machine learning approaches to the problem of Hardware Trojan identification, including a K-Nearest Neighbor (KNN) model, a Multi-Layer Perceptron (MLP), and a Convolutional Neural Network (CNN) based on ResNet (as discussed in Section 3.1). We also evaluated two others, a KNN approach directly on the measured traces with 5 nearest neighbors, and an MLP with 4 fully connected layers and dropout layers to avoid overfitting [13], trained with a learning rate of 0.001 for 30 epochs. The CNN was trained for 10 epochs at the same learning rate. All training was done on a personal computer with a GTX 1070Ti graphic card. As shown in Equation 2, We define the sum of true positive cases and true negative cases over all test cases as the result accuracy, i.e. the rate of correctly identifying the

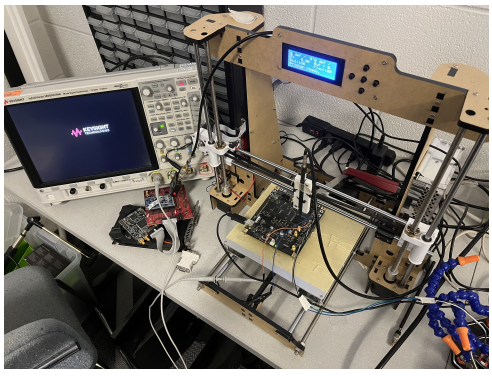

Figure 6: Data Collection Setup

Table 1: A brief comparison of different machine learning approaches to the problem of HT identification rate. The CNN outperforms the other approaches.

Trojan Type	KNN	MLP	CNN
Golden Model	75%	0%	100%
T100 (Active)	82%	100%	100%
T200 (Active)	100%	100%	100%
T400 (Inactive)	39.5%	100%	100%
T400 (Active)	100%	100%	100%
T500 (Active)	100%	100%	100%
T600 (Active)	100%	100%	100%
T600 (Inactive)	29.5%	0.9%	100%
T700 (Inactive)	100%	100%	100%
T700 (Active)	100%	100%	100%
T800 (Inactive)	100%	100%	100%
T800 (active)	100%	100%	100%
T900 (Inactive)	66.5%	0%	100%
T900 (Active)	100%	100%	100%
T1000 (Inactive)	86.5%	100%	100%
T1000 (Active)	100%	100%	100%
T1100 (Inactive)	0%	78.5%	100%
T1100 (Active)	50.5%	100%	100%
T1200 (Inactive)	100%	100%	100%
T1700 (Inactive)	100%	100%	100%
T1700 (Active)	100%	100%	100%
T1800 (Inactive)	100%	100%	100%
T1800 (Active)	100%	100%	100%
T1900 (Active)	100%	33.5%	100%
T2000 (Inactive)	75%	100%	100%
T2000 (Active)	100%	100%	100%
T2100 (Inactive)	50%	100%	100%
T2100 (Active)	100%	100%	100%
T2 (Inactive)	100%	100%	100%
T2 (Active)	100%	100%	100%
Mean (All HTs)	81.5%	79.18%	**100%**

Trojans.

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN} \quad (2)$$

And the results of these can be found in Table 1, which clearly shows the Residual Model outperforming the other two. The loss was adjusted to reflect unequal class distribution as discussed in Section 4.1 where appropriate.

4.3 New Trojan Detection

This approach can detect unseen new Hardware Trojans based on whether they are classified into belonging to the golden model or a known Trojan. When the deep learning model is trained on the data from a set of Trojans and an EM side-channel trace of an unseen new Trojan is input into the model, as long as the new trace is not categorized into the golden class, the model detected anomalous Trojan-relate behavior. We define the 'false negative rate' in this instance to then be the percentage of traces from designs with Trojans that are classified as being without any HTs.

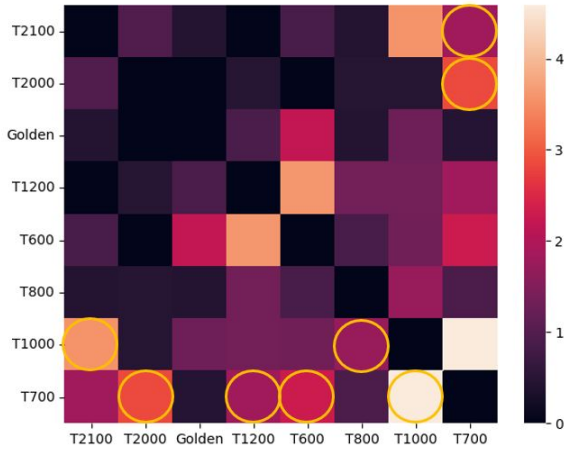

Figure 7: A modified heatmap showing what traces from designs with trojans along the horizontal are classified into.

We first evaluate if a new Trojan can be detected on the full dataset using the K-fold verification mentioned in Section 3.4. Our model achieves a 0% false negative rate on all Trojans selected from Trusthub following this methodology.

We select a subset that contains traces from inactive T2100, T2000, T1200, T1000, T800, T700, and T600 designs for visualization. The result is shown in Figure 7. The circles indicate what category a new unseen Trojan along the horizontal axis would be classified into when testing. When evaluating T1200 as an unknown Trojan, the model has a 10% false negative rate because only a subset of EM traces of Trojans was used for the training. Otherwise, the model achieves a false negative rate of 0% when using the other Trojans as new Trojan test cases.

By comparing Figure 5 and Figure 7, the relationship between SIFT similarity and Trojan classification should be clear. Specifically, when the model is presented with a trace from an unseen HT, it is most likely to be classified as containing a HT with the most similar design. The theory behind is explained in Sections 3.3 and 3.4.

Furthermore, when an unknown Trojan with low confidence scores(less than 0.7) to the Trojans that were already trained in the machine learning model, it is possible to classify it as an unknown Trojan. Therefore, the proposed approach can also detect unknown Trojans.

5 DISCUSSION

It is believed that the power grid of an IC, which has the most significant current flow inside, generates the majority of EM leakage of a chip [14]. However, power side-channel signals and EM

side-channel signals have a key difference. EM side-channel traces depend on the distance from each signal source (e.g. the distance from the corresponding wires of the power grid), which are in turn affected by the floorplan in FPGAs or placement and routing (P&R) in Application Specific Integrated Circuit (ASIC) designs while power side-channel only depends on the delay of each signal to the power measuring point.

To explain this, let us consider Equation 3 which describes the power side-channel. $S_i(A, B)$ denotes the switching behavior of each gate and D denotes the delay of each gate to the power sensing node. Compared with the Equation 1, we can draw the conclusion that EM side-channel leakage is floor-plan sensitive as Equation 1 depends on F, which is absent in Equation 3. This observation leads to the conclusion that when Hardware Trojan is inactive, and thus when the power consumed by the trigger circuits is small, the power side-channel may not detect the difference caused by Hardware Trojan. However, the EM side-channel can be due to the floor-plan change modifying the EM side-channel. We plan to conduct further research to explore this in our future work.

$$L_P = f\{S_i(A, B), D_i\} \tag{3}$$

One criticism of the above would be that the Hardware Trojans provided on Trust-hub are large enough to have a substantial effect on the floorplan, unlike many other HTs. Specially designed analog HTs like A2 Trojan [19] is small enough to avoid detection based on size and power and concealing enough to pass most test phases. This concern naturally brings up a fundamental question to all EM side-channel analysis waiting to be answered, which is What is the real potential of EM side-channel analysis in HT detection or how much difference in a design's size and power is needed for EM side-channel HT detection to work.

If we utilize the findings in this paper to answer the above question, a workflow can be developed to systematically evaluate EM side-channel analysis or any other side-channel methods. The workflow would consist of two parts:

(1) A data collection phase, where the floorplan of a given circuit, like AES128, is manually manipulated by adding gates and changing routing. Thus, we gradually decrease the similarity between the design before and its following versions. Every time we change the circuits, we collect EM traces or data from other side-channels.
(2) Then in a data analysis phase, the collected traces are used to train a CNN model. Among the test subsets that reach a base line accuracy, we select the subset with the highest similarity to the original design, or in other words, having the least modifications between the two designs included.

Following this method, it should be possible to find the minimum amount of alteration that EM side channel can detect, to determine the limitations to assurance of EM side channel methods when evaluating a design.

6 CONCLUSION

In this paper, we train a convolutional neural network on EM side-channel traces collected from an FPGA and successfully use the trained model to identify known types of Hardware Trojans, whether they are active or inactive, as well as detect the presence of Hardware Trojans unseen during training. We evaluate the hypothesis that HTs which make similar modifications to the floorplan will generate similar EM leakages. The theory is supported by a subset of AES128 Trusthub Hardware Trojans. We also briefly introduce a methodology to evaluate the minimum amount of floorplan change that EM side-channel analysis can detect, which we plan to evaluate in our further experiments.

7 ACKNOLEDGEMENT

This work was supported by the Office of Naval Research under Award Number N00014-19-1-2405.

REFERENCES

[1] Swarup Bhunia, Michael S Hsiao, Mainak Banga, and Seetharam Narasimhan. 2014. Hardware Trojan attacks: Threat analysis and countermeasures. *Proc. IEEE* 102, 8 (2014), 1229–1247.
[2] Sina Faezi, Rozhin Yasaei, and Mohammad Abdullah Al Faruque. 2021. HTnet: Transfer Learning for Golden Chip-Free Hardware Trojan Detection. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1484–1489. https://doi.org/10.23919/DATE51398.2021.9474076
[3] Hendra Guntur, Jun Ishii, and Akashi Satoh. 2014. Side-channel attack user reference architecture board SAKURA-G. In *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*. IEEE, 271–274.
[4] Jiaji He, Haocheng Ma, Yanjiang Liu, and Yiqiang Zhao. 2020. Golden chip-free trojan detection leveraging Trojan trigger's side-channel fingerprinting. *ACM Transactions on Embedded Computing Systems (TECS)* 20, 1 (2020), 1–18.
[5] Jiaji He, Yiqiang Zhao, Xiaolong Guo, and Yier Jin. 2017. Hardware trojan detection through chip-free electromagnetic side-channel statistical analysis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25, 10 (2017), 2939–2948.
[6] Tamzidul Hoque, Seetharam Narasimhan, Xinmu Wang, Sanchita Mal-Sarkar, and Swarup Bhunia. 2017. Golden-free hardware Trojan detection with high sensitivity under process noise. *Journal of Electronic Testing* 33, 1 (2017), 107–124.
[7] Zhao Huang, Quan Wang, Yin Chen, and Xiaohong Jiang. 2020. A Survey on Machine Learning Against Hardware Trojan Attacks: Recent Advances and Challenges. *IEEE Access* 8 (2020), 10796–10826. https://doi.org/10.1109/ACCESS.2020.2965016
[8] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.
[9] Sumio Morioka and Akashi Satoh. 2002. An optimized S-Box circuit architecture for low power AES design. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 172–186.
[10] Payam Refaeilzadeh, Lei Tang, and Huan Liu. 2009. Cross-validation. *Encyclopedia of database systems* 5 (2009), 532–538.
[11] Hassan Salmani, Mohammad Tehranipoor, and Ramesh Karri. 2013. On design vulnerability analysis and trust benchmarks development. In *2013 IEEE 31st international conference on computer design (ICCD)*. IEEE, 471–474.
[12] Bicky Shakya, Tony He, Hassan Salmani, Domenic Forte, Swarup Bhunia, and Mark Tehranipoor. 2017. Benchmarking of hardware trojans and maliciously affected circuits. *Journal of Hardware and Systems Security* 1, 1 (2017), 85–102.
[13] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.
[14] Akihiro Tsukioka, Karthik Srinivasan, Shan Wan, Lang Lin, Ying-Shiun Li, Norman Chang, and Makoto Nagata. 2019. A fast side-channel leakage simulation technique based on IC chip power modeling. *IEEE Letters on Electromagnetic Compatibility Practice and Applications* 1, 4 (2019), 83–87.
[15] Zhiguang Wang and Tim Oates. 2015. Imaging Time-Series to Improve Classification and Imputation. https://doi.org/10.48550/ARXIV.1506.00327
[16] Zhiguang Wang, Weizhong Yan, and Tim Oates. 2016. Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline. https://doi.org/10.48550/ARXIV.1611.06455
[17] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor. 2016. Hardware Trojans: Lessons Learned after One Decade of Research. *ACM Trans. Des. Autom. Electron. Syst.* 22, 1, Article 6 (may 2016), 23 pages. https://doi.org/10.1145/2906147
[18] Hao Xue and Saiyu Ren. 2017. Self-reference-based hardware Trojan detection. *IEEE Transactions on Semiconductor Manufacturing* 31, 1 (2017), 2–11.
[19] Kaiyuan Yang, Matthew Hicks, Qing Dong, Todd Austin, and Dennis Sylvester. 2016. A2: Analog malicious hardware. In *2016 IEEE symposium on security and privacy (SP)*. IEEE, 18–37.