3-28-2023

Hardware Trojan Detection in Chips by Removing Dependencies Between Features in Machine Learning

Alfred Moussa
Boise State University

Nader Rafla
Boise State University

Hardware Trojan Detection in Chips by Removing Dependencies Between Features in Machine Learning

Abstract

Globally, there has been an increase in demand for System on Chip (SoC) applications, active medical implants, and Internet of Things (IoT) devices. However, due to challenges in the global supply chain, the design, fabrication, and testing of Integrated Circuits are often outsourced to untrusted third-party entities around the world rather than a single trusted entity. This situation presents an opportunity for adversaries to compromise the device's integrity, performance, and functionality by inserting malicious modifications known as Hardware Trojans (HTs) into the original design. HTs can also create a "backdoor" in the system for malicious alterations.

In this research, a solution to the issue of hardware trojan is presented through the utilization of machine learning models that rely on supervised and unsupervised learning. The proposed method involves providing the netlist features of the digital hardware design post-synthesis to the machine learning model and removing any interdependence between features to prevent overfitting of the training dataset. The supervised model showed a 99.2\% true positive and true negative rate, as well as an F-measure of 99.3\%, while the unsupervised model achieved a 99.5\% true positive rate with the use of random projection, thereby offering a more resilient machine learning-based method for detecting hardware trojans.

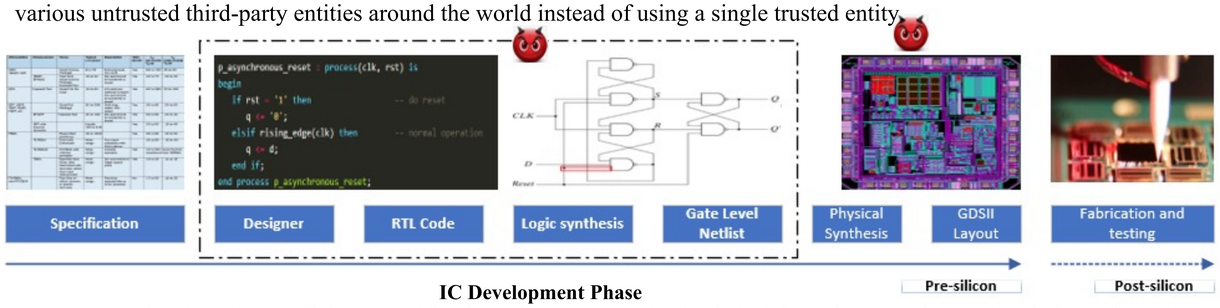
Hardware Trojan Detection in Chips by removing dependencies between features in Machine Learning

Alfred Moussa [1], Dr. Nader Rafla [2]
[1]*Boise State University, Boise, ID, 83725 USA.*

BOISE STATE UNIVERSITY

Introduction

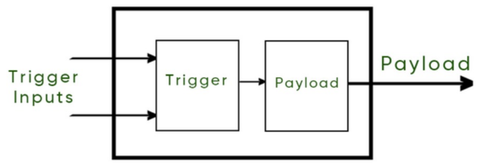
1. Globally, there has been increase in demands for system on Chip (SoC) applications, active medical implants and Internet of Things (IoT).
2. Due to the global supply chain challenges, Integrated Circuits processes of design, fabrication, and testing were outsourced to various untrusted third-party entities around the world instead of using a single trusted entity.



IC Development Phase

3.Hardware Trojan (HT) is a malicious modification of an Integrated Circuit (IC) intended to leak sensitive information, change the functionality of a system, degrade the performance, cause a denial-of-service (DoS), or leave a backdoor to the whole system.

Characterization of Hardware Trojan



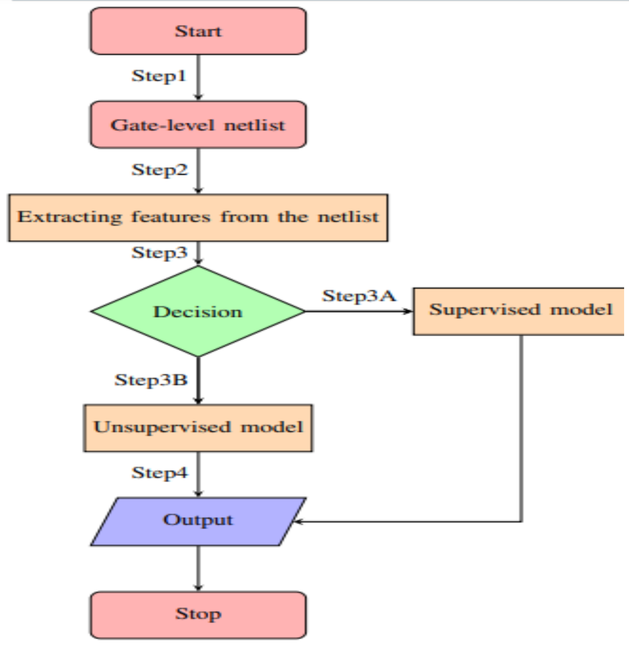
Basic structure of a Hardware Trojan

A typical Hardware Trojan consists of a trigger and payload circuit. Trigger monitors a rare call (signals) from the circuit and transforms unique signals from the circuit into an effective trigger for the payload as shown in Figure below. Payload of HT is the entire activity that triggers the execution of HT function.

Significant Effect of HT on chips

- In 2007, a suspected nuclear installation in Syria was bombed by Israeli jets because Syrian radar was crippled by a remote kill switch thru a backdoor in its commercial off-the-shelf microprocessor [1].
- In 2010, the U.S military bought over 59,000 microchips destined for installation in everything from missile defense systems to gadgets that tell friend from foe where they found an HT implemented on the chip giving the enemy a backdoor to their whole system [2]
- In 2012, Hardware Trojan backdoor existed in the Actel/Microsemi ProA-SIC3 chips used in the military-grade FPGAs. This HT added undesired additional JTAG functionality on the silicon itself that allowed the extraction of secret keys, enabling adversaries to modify the chip's configuration and gain control of the system [3].

HT Detection scheme



Unsupervised & Supervised Hardware Trojan detection flow diagram

Step 1: Input gate-level-netlist

Process started by synthesizing the hardware design from behavioral Verilog to structural Verilog in Cadence using the genus tool by writing a TCL script. It specifies timing constraints for the design such as:

- initializing the clock period to 20000 ps to set the operating
- frequency to 50MHZ, and defining the package used as 45nm technology.
- Specifying the characterization of timing and power for static timing analysis (STA).

Machine Learning Models for HT Detection

Step 2: Feature Extraction

The genus tool from Cadence was used to generate multiple reports that define the timing, power, and area of the design in an output as shown in Figures below.

```
==========================================================
  Generated by:           Genus(TM) Synthesis Solution 18.14-s037_1
  Generated on:           Sep 05 2022  04:53:35 pm
  Module:                 aes_128
  Operating conditions:   PVT_1P1V_0C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
  Description:            AES_100 (Trojan Free)
==========================================================


  Gate   Instances    Area       Library
-----------------------------------------------
DFFQXL          128   700.416    fast_vdd1v0
INVXL           128    87.552    fast_vdd1v0
SDFFQX1         128   963.072    fast_vdd1v0
-----------------------------------------------
total           384  1751.040


     Type       Instances   Area   Area %
-------------------------------------------
sequential            256 1663.488   95.0
inverter              128   87.552    5.0
unresolved             20    0.000    0.0
physical_cells          0    0.000    0.0
-------------------------------------------
total                 404 1751.040  100.0
```

Step 3:Decision

Step 3.A Supervised machine learning model

Step 3A.1: Dropping step

Step 3A.2: Data Shuffling - It is imperative to shuffle datasets during training to prevent the model from learning a definitive pattern.

Step 3A.3 MinMaxScaler: The estimator scales and translates each feature individually such that it is within the range (0,1) on the training set.

Step 3A.4 Random Forest classifier - A random forest classifier is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Step 3.B Unsupervised machine learning model

Step 3B.1 Removing Labels - The unsupervised learning model doesn't use labels to identify patterns. Therefore, insights tend to be less biased when they are removed from the data.

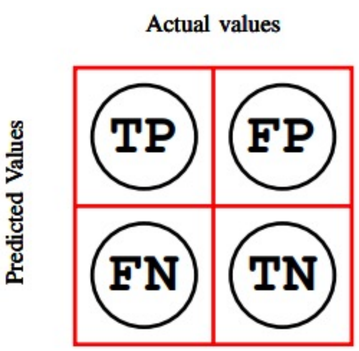
Step 3B.2 Data Shuffling - It is the same process as step 3A.2.

Step 3B.3: Random Projection - Used to reduce the dimensionality of the datasets in Euclidean space and guarantee similar embedding quality while being much more memory efficient and allowing faster computation on the projected data.

Step 3B.4: Random Forest classifier - The labels were removed prior to applying Random Forest classifier.

Step 4: Output

Visualizing output performance of our models are done through a confusion matrix. A confusion matrix is a performance measure for a Machine Learning classification problem when the output is more than one class



Conclusion

To sum up, using machine learning to detect hardware Trojans is difficult because of the large amount of data and the risk of overfitting. Overfitting can cause inaccurate results, so it's better to remove any linearity between features to improve the accuracy of the machine learning model.

Approach	N-Features	TN	FP	FN	TP	TPR	TNR	precision	F-measure
Supervised	9	280	2	5	622	99.2%	99.2%	99.6%	99.3%
Unsupervised	3	282	1	3	623	99.5%	99.6%	99.8%	99.6%

References

[1] S. Adee, "The hunt for the kill switch," in IEEE Spectrum, 2008.
[2]A. Rawnsley, "Fishy chips: Spies want to hackproof circuits," Wired, Jun. 24, 2011. [Online]. Available:https://www.wired.com/2011/06/chips-oy-spies-want-tohack-proof-circuits/
[3] S. Skorobogatov and C. Woods, "Breakthrough silicon scanning discovers backdoor in military chip," in Cryptographic Hardware and Embedded Systems – CHES 2012, E. Prouff and P. Schaumont, Eds. Springer Berlin Heidelberg, 2012.