4 Machine Learning-based Classifier for Hardware Trojan Detection

Main Objective In this project, you must design machine learning (ML) based classifiers to detect a hardware Trojan.

Background Detecting malicious modifications (“Trojans”) in a fabricated chip is difficult. The designer does not know the size, type, location, trigger, or payload of a potential Trojan, and there is a lot of complex logic on a chip that must be analyzed before the chip can be declared Trojan-free. One class of Trojan detection mechanisms analyzes variations in power consumed by a chip. Since any chip activity consumes power, unexpected activity or variations may indicate the presence of Trojans.

In [1], the authors put identical ring oscillators in different locations of a fabricated chip to monitor the power supply network at runtime. The key idea behind this is that RO frequencies are affected by the power supply and temperature variations (runtime activity), capacitance, threshold voltage, etc. variations (from the manufacturing process). Since wires are not perfect superconductors (they have finite resistance and inductance), any Trojan switching will draw power in one part of the on-chip power network and also cause a small, temporary voltage dip in a different part of the circuit. This voltage drop causes a temporary drop in RO frequency when the Trojan is active. If the resulting drop in RO frequency can be detected, then the Trojan can be detected.

However, the real-life scenario is much more complicated. There is no single frequency that works as a pass/fail threshold for detection: ROs experience random process variations and, therefore, produce slightly different frequencies even with similar design and operating conditions (recall the RO frequency data in homework 2), so it is difficult to compare RO frequencies from different chips even if they are both Trojan-free. Also, normal switching activity can cause RO frequency variations as large as, or even larger than, Trojans. Thus, smart classifiers are required. This is why you will design a machine learning classifier to distinguish trojan-infected from trojan-free chips.

Project Goal

1. Choose two cases from the following and implement them:
   - Case 1. You have some known samples of both Golden chips and Trojan-inserted chips. (i.e., RO data from both types of chips are known), and both can be used for training the classifier. (However, the type of Trojan is unknown.)
   - Case 2. You only have some known samples of Golden chips, i.e., only golden data can be used for training the classifier.
   - Case 3. You have completely unidentified samples (no knowledge about the samples, whether they are golden, Trojan-inserted, or a mixture of both) to train the classifiers.
2. Implement two different classification techniques of your choice.
3. Evaluate your classifier accuracy with the given sample size (33 total samples available):
   - 6 samples (Case 1: 3TF and 3TI; Case 2: All 6TF; Case 3: All 6 Unknown)
   - 12 samples (Case 1: 6TF and 6TI; Case 2: All 12TF; Case 3: All 12 Unknown)
   - 24 samples (Case 1: 12TF and 12TI; Case 2: All 24TF; Case 3: All 24 Unknown)
   - All remaining samples are used for evaluation.

   For each case, choose the samples for the training set randomly (20 trials) and use the remaining chips/samples for evaluation. Run your classification/detection and collect results (20 trials) for each Trojan type (23 total), and report the average accuracy result (i.e., % of successful detection).

Software Requirements

- Scripting language for analyzing/visualizing your results. Python, MATLAB, or others.

- Dataset: ROFreq.zip (on Canvas) contains 33 CSVs (Chip1, Chip2, Chip33) that refer to data from 33 chips. Each spreadsheet has the following specifications:
  - Each row indicates Trojan Free/ Trojan-inserted data.
  - Trojan Free (golden) data in row 1 and row 25
  - Trojan-inserted data in rows 2-24
  - Each column indicates the frequency of RO# in the network (RO1-RO8). For details, check Ref. [1].

Report to submit.

- IEEE conference format. (4-6 pages)
- The report should include (tentative marks dist.):
  1. Top Sheet with group members names and ID
  2. Introduction, motivation, and problem statement. (10%)
  3. Classifiers selection and design (40%)
     - Reason of choice, Brief description, Pros/Cons
     - Short workflow/algorithm, model architecture, and parameters, etc.
  4. Classifier evaluation Results (35%)
     - For each classifier, provide the following: Working flow-chart/algorithm and used boundary conditions, parameters, etc., as necessary.
     - For each problem case, provide the following:
       * Average true positive and false positive rates over the 20 trials (figures, tables, etc.).
       * Average training time (if appropriate) and evaluation time (if appropriate).
       * Comparative results
         - Among different sample sizes
         - Among different cases (if applicable)
       * You can also provide any other necessary data to justify your implementation. Some sample tables/plots are given for your convenience.
  5. Conclusion and personal comments (10%)
  6. Clarity, organization, etc. (5%)

Submission guidelines

- Submit a .zip file with name: Project#Group#
- Include all of the following:
  - Report.pdf
  - Working directory for your analysis
    * readme.txt (provide necessary instruction for running your code)
    * All source files
    * Data set

References

1. Shane Kelly, Xuehui Zhang, Mohammed Tehranipoor, and Andrew Ferraiuolo, Detecting Hardware Trojans using On-chip Sensors in an ASIC Design. Journal of Electronic Testing 31, no. 1 (2015): 11-26.