

Design/Practical Experience [CSN1020]

Department of Computer Science and Engineering

Final Report

Academic Year: 2021-2022

Semester: 2

Date of Submission of Report: 01 May 2022

Name: Risheek Nayak

Roll Number: B20AI058

Title: ML-based hardware Trojan detection

Authors: Risheek Nayak (B20AI058), Suyash Jaiswal (B20EE070), Likith Biyani (B20CS085), Divyanshi Singh Bora (B20EE018), Ghelani Shubham Bhaveshbhai (B20EE019)

Mentor's Name:

Binod Kumar

Assistant Professor in Electrical Engineering department at IIT Jodhpur (IITJ)

Abstract: We are creating a Machine learning model for hardware trojan detection with C++, Bash, and python as the languages for design.

INTRODUCTION: A Hardware Trojan (HT) is a malicious modification of the circuitry of an integrated circuit(IC). A hardware Trojan is completely characterized by its physical representation and its behavior.

Due to globalization the IC's prices have been reduced leading to the manufacturing of these by third-party vendors which leads to inserting hardware trojans in them and their activities remain unchecked.

DISADVANTAGES: The disadvantages of having an HT-infected IC are that they can cause malfunctions, destroy the IC products, and leak secret information.

OUR APPROACH: We hereby are extracting the features out of an infected and a normal IC and then training our model with those features and then predicting whether the new IC is provided to us then is HT positive or not. All of this is performed with the help of machine learning.

Work Done:

1. Reading the research paper:
 - To detect trojans employing a statistical correlation-based clustering.
 - Density-based algorithm (ordering points to identify the clustering structure optics)

- False-positive accuracy(~0.01)
 - Training the model with the help of a support vector machine.
 - A useful technique for data classification
 - Transforming data to SVM format
 - Then scaling
 - Model selection
 - RBF kernel-non linearly maps samples into a higher dimensional space
 - Cross-validation and grid search (for increasing the accuracy)
 - Training the model with the help of sci-kit learn.
 - It provides a wide variety of machine learning algorithms, both supervised and unsupervised.
 - It relies on the scientific Python ecosystem; it can easily be integrated into applications outside the traditional range of statistical data analysis.
2. Generating trojan files and features:
- Automated trojan file generation:
 - A python code that takes in .Bench/.txt file of the circuit as input and generates 100 trojan infected files as the output.
 - It inserts 1 or 2 trojan gates randomly in the copy of the original circuit file and returns the trojan file(copy).
 - The trojan added is validated using a small python code that returns the trojan added by the automated trojan adder code.
 - Features generation:
 - A python code that takes in .Bench/.txt file of the circuit as input and generates a text file of the features for each gate/DFF in the circuit and also the average values of all the features.
 - It consists of different functions made for the respective feature values to be generated for each gate/DFF in the circuit file.
 - The code for the above can be found at [https://github.com/likith-02/Hardware Trojan Detection](https://github.com/likith-02/Hardware_Trojan_Detection)
3. Simulation:
- Generation of input files:
 - Based on the number of files to be generated and the number of rows (number of cycles in the bench file + 1) and the number of cols (number of inputs in the bench file), a C++ program was written.
 - This C++ program returns files that contain randomly generated numbers of either zeros or ones.
 - Bash Scripting:
 - Learning Bash Scripting and downloading Virtual Box to run Ubuntu.

- Code for generating the Bash Script
 - Generating bash script to run the simulation for the trojan files.
 - Generating bash script to run the simulation for the non-trojan files.
 - Generating bash scripts to run convert the ff_values/ output_values to a table format.
 - These codes were written in C++ and will generate a .sh file containing the bash script.
 - Simulation:
 - Using the bash script generated, the simulation was done on the trojan files and the non-trojan files with 1000 different input files.
 - This simulation code was given to us by the mentor.
 - The simulation produced ff_values and output values files.
 - To convert the ff_values/ output values to a table format:
 - A C++ program was written which converts the ff_values/output values to a table format.
 - This code returns a new file.
 - <https://github.com/RisheekNayak/ML-based-Hardware-Trojan-Detection> Github repository for the above codes.
4. Generation of the features:
- Generated the features to create the final CSV file on which we had to work. It is a python code that takes .txt files as input and produces five features for a gate and gives us the CSV file as output. It was a code given by my mentor but we edited it and added our part to it so that it can read a large number of files at a time. Also, we generated the final CSV files to work with and tried various ML models on it.
- The five features are:
- Counting the total number of 1s across the cycle.
 - Counting on which cycle the 1st one is observed.
 - Counting the cycle from which consecutive runs of 1 are observed.
 - Counting number of occurrences of 5 consecutive runs of 1.
 - Counting maximum length of consecutive runs of 1.
5. ML:
- Link to Google colab: https://colab.research.google.com/drive/17-q_NEdK1cEK2rDXwRkNrORqXCdSqbXQ?usp=sharing
- Data Preprocessing:
 - Importing the libraries
 - Importing the dataset
 - Generation of the labels: We require two arrays, one for each flip flop and one for each circuit. This will contain zeros or ones representing non-trojan and trojan respectively.

- Encoding: Label Encoder was used to do the following.
- The method used for prediction:
 - In a given circuit, if any one of the gates produces a positive result for a trojan, we have come to the conclusion that the circuit will have a trojan.
- Various classification techniques:
 - Logistic Regression:
 - Training on the training set and then testing on the test set. The following confusion matrix was obtained:

```
Confusion Matrix
[[ 0 200]
 [ 0 400]]
```

- Naive Bayes Classifier:
 - Training on the training set and then testing on the test set. The following confusion matrix was obtained:

```
Confusion Matrix
[[ 0 200]
 [ 0 400]]
```

- Decision Tree Classifier:
 - Training on the training set and then testing on the test set. The following confusion matrix was obtained:

```
Confusion Matrix
[[ 0 200]
 [ 0 400]]
```

- Random Forest Classifier with n_estimators 100 and criterion as 'entropy'
 - Training on the training set and then testing on the test set. The following confusion matrix was obtained:

```
Confusion Matrix
[[ 0 200]
 [ 0 400]]
```

- Using the ROC curve to find the best threshold with the model Random Forest Classifier:
 - Training on the training set and then testing on the test set. The following confusion matrix was obtained:

```
Confusion Matrix
[[ 0 200]
 [ 58 342]]
```

- XGBoost with ROC curve threshold:
 - Training on the training set and then testing on the test set. The following confusion matrix was obtained:

```
Confusion Matrix
[[ 0 200]
 [ 0 400]]
```

Analysis:

1. As we can see from the confusion matrix, the non-trojan circuits which have been falsely accused as trojans are very high.
2. This was because if only one of the gates was falsely accused as trojan positive, the entire circuit was concluded as trojan infected.

Conclusion:

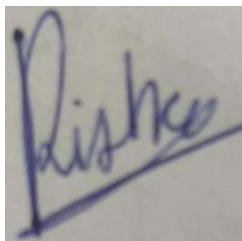
1. Poor accuracy due to high false accusation rate.
2. False Negative rate was low (i.e., no trojan infected circuit was categorized as non-trojan)
3. Random Forest Classifier with ROC curve threshold performed poorly among the other models as it predicted few trojan infected circuits as non-trojan.

References:

- B. Cakir and S. Malik, "Hardware trojan detection for gate-level ics using signal correlation based clustering,"
- C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.
- M. Oya, Y. Shi, M. Yanagisawa, and N. Togawa, "A score-based classification method for identifying hardware-trojans at gate-level netlists," in Proc. Design, Automation and Test in Europe (DATE), pp. 465–470, 2015.
- "Hardware Trojan Detection Using Machine Learning" research paper from University of Windsor

- C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification."
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vander-plas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duches-nay, "Scikit-learn: Machine learning in python," The Journal of MachineLearning Research

Declaration: I declare that no part of this report is copied from other sources. All the references are properly cited in this report.



Signature of the Student



Signature of the Supervisor

Supervisor's Recommendation for the Evaluation

Please tick any one of the following

- ☒ 1. The work done is satisfactory, and sufficient time has been spent by the student. The submission by the student should be evaluated in this term.
2. The work is not complete. Continuity Grade should be given to the student. The student would need to be evaluated in the next semester for the same Design Project with me.
3. The work is not satisfactory. There is no need for evaluation. The students should look for another Design Credit Project for the next semester.
4. [Other Comment, if 1-3 are not valid] _____



Signature of the Supervisor