Machine Learning-based Classifier for Hardware Trojan Detection

line 1: 1st Given Name Surname   
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 4th Given Name Surname  
line 2: *dept. name of organization*  
*(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 2nd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 5th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 3rd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 6th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

Diagram

Description automatically generated*Abstract*—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. *\*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract*. (*Abstract*)

Keywords—component, formatting, style, styling, insert (key words)

# Introduction

## Motivation and Problem Statement

# Background

# Methods

The objective of this project is to distinguish Trojan Free (TF) RO frequencies and Trojan inserted (TI) RO frequencies. Although using a simple threshold like what we did in previous homework is tempting, unfortunately, the random process variation between each individual chip sample makes it unrealistic. As a result, machine learning classifiers are taken as potential solutions to our Trojan detection problem since we have both golden samples (Trojan free) and problematic ones (Trojan inserted) for all 33 chips.

Machine learning algorithms are mainly classified into two categories, which are supervised learning algorithms and unsupervised learning algorithms. For supervised learnings, both data and labels are known before training and testing. While in unsupervised learning, there’s no label at all, which means the hidden pattern is supposed to be found by the algorithm in the training process. In this project, both types will be used and compared for better performance. The procedure workflow is shown below.

## Data preparation

Diagram

Description automatically generated with medium confidence

The first step is to randomly select samples according to the project requirements. Take training sample size 6 as an example, random.sample(indices, num) method in Python is used to get random numbers from 1 to 33. Then decide whether the data type we need is Trojan free or Trojan inserted, if it is case 1 (the supervised learning), then half of the sample will be TI and the other half will be TF; While for case 3 (unsupervised learning), the type will be random by using random.choices(['TI','TF']. The rest of the data will be all test dataset for evaluations. The whole operations described below are packaged in our self-defined train\_test\_split() function.

## Classifier Selection and Design

## Case 1 – Supervised learning

1. *K-Nearest Neighbors*

As we are dealing with a binary classification problem, K-Nearest Neighbors (KNN) is naturally selected first for its simplicity in implementation. Beyond its simplicity, KNN also has a good performance in classification problems with irregular boundary for being a non-parametric method. We think it will be a good match to our problem since the RO frequencies are affected by multiple factors like the power supply, temperature variations, capacitance, threshold voltage, etc. and have very vague boundaries between samples. However, for large datasets with multiple features, KNN has its con in the computational complexity for it will store all the training data, and because of this, the prediction may be slower compared to other supervised learning algorithms. But in our case, the feature size for each row is 8 and we only have 33 chips, the impact of size of data is relatively mild.

The basic assumption of KNN is that similar data is close to each other, as a result, in order to decide which data point among the neighboring candidates is the closest to a given query point, distances will be calculated.

The most often used distance metric is the Euclidean distance, which is defined as the following:

For our project, the n equals to 8, since each chip has 8 RO frequencies. The main hyper-parameter in KNN is K, namely the number of neighboring data of a given point. We use sklearn packages in Python to implement KNN classification and change K from 2 to 10 with step size 1 to search for the K yielding the best training performance. Besides tuning the number of neighbors, leaf\_size parameter is also tuned in range of 2 to 30 with step size 2. This parameter has effects on construction time, query time and the memory size, which will mitigate KNN’s disadvantage in computational complexity.

1. *Logistic Regression*

Logistic regression algorithm is usually used in binary classification problems, and it is suitable for our supervised learning task, to decide whether the chip is Trojan Free, or Trojan inserted. This method is based on the concept of probability. Unlike linear regression, binary logistics regression transforms its outputs to probability space by using the logistic sigmoid function.

Like KNN, logistic regression also has its advantages in easy implementation, and it also can achieve good accuracy for simple data sets which are linearly separable. What’s more, logistics regression doesn’t require high computational power which outperforms KNN in this aspect. Yet logistic regression has its main limitation in its assumption of linearity between the dependent variables and the independent variables. In our dataset, we are not sure about the correlation between 8 ring oscillators frequencies.

## Case 3 – Unsupervised learning

1. *K-Means clustering*

K-Means is an unsupervised learning algorithm which aims to divide the input dataset to K different clusters. When a new data comes in, it will be distributed to the cluster whose center point is the closest to the new data. Then the center point will also be updated accordingly. Thus, this algorithm is very sensitive to outliers which is its biggest disadvantage. But this algorithm is still worthy of trying as a baseline for further optimization for its simple implementation in sklearn.

1. *DBSCAN*

Different from the partition-based clustering K-Means, DBSCAN is a density-based clustering method which stands for density-based spatial clustering of applications with noises. Considering the frequency of ring oscillators can be easily influenced by the environment, DBSCAN may be a good helper to eliminate the effects of these variations. To implement this algorithm, we first import the DBSCAN from the sklearn.cluster, and tune the eps (distance of neighbors) and min\_samples to get the best trained model.

Compared to K-Means, DBSCAN doesn’t need to specify the number of clusters beforehand and is robust to outliers. In our RO data, we have more trojan-inserted data than trojan-free in every sample chip (the ratio is 23:2), fortunately, DBSCAN has a great advantage in separating clusters of high density from those of low density, which makes it a candidate with large potential.

# Results

## Case 1 Results -

## Case 3 Results -

# Conclusion

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

##### References

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published**