Hardware Trojan Identification Using Machine Learning-based Classification

Nur Qamarina Mohd Noor, Nilam Nur Amir Sjarif, Nurul Huda Firdaus Mohd Azmi,Salwani Mohd Daud and Kamalia Kamardin
Advanced Informatics School (AIS), Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia.
nilamnur@utm.my

***Abstract*—As Hardware Trojans (HTs) emerges as the new threats for the integrated circuits (ICs), methods for identifying and detecting HTs have been widely researched and proposed. Identifying the HTs are important because it can assist in developing proper techniques for inserting and detecting the treat in ICs. One of the recent method of identifying and detecting HTs in ICs is classification using machine learning (ML) algorithm. There is still lack of machine learning-based classification for HTs identification. Thus, a three type of ML based classification includes Decision Tree (DT), K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) are proposed for HTs identification. The dataset is based from the Trust-Hub. In order to improve the classification accuracy, the HTs are discretized based on their dominant attributes. The discretized HTs are classified using three machine learning algorithms. The results show that the DT and KNN learnt model are able to correctly predict about 83% of the test data.**

***Index Terms*—Classification; Hardware Trojan; Machine Learning; TrustHub.**

I. INTRODUCTION

The hardware Trojan (HT) is a new emerging type of hardware attack for integrated circuits (ICs) and has become an important research topic in recent years. The impacts from HTs are malicious such as leaking sensitive user information or disabling or altering the functionality of an IC. For instance, in 2007, an incoming air strike bypassed the Syrian radar which is due to the backdoor that were built into the system's chips [1]. In another case, it was exposed that HTs were directly implanted into USB protocol or port by the Quantum program of US National Security Agency (NSA) in 2014 [1] to acquire secret data from all over the world not only from Russian and China's military, but also from the trading information from EU and Mexican's law enforcement and drug cartel computer system.

The structure of HTs comprises of triggers and payloads. The triggers are defined as the mechanism to activate the HTs while the payloads are the resulting effects from the HTs. To avoid detection, the HTs are typically stay dormant in the IC until they are triggered by rare signals or events [1]. Upon occurrences of the specified signal or event, the activated payload circuit begins to implement malicious functions. The triggers of the HTs are usually intelligently designed which

they will not be induced during simulation or testing but only occur by covert field operation. In order to recognize these HTs, a number of typical HT-inserted benchmark circuits are developed in the TrustHub website [2]. The HTs benchmarks circuits database are created based on six factors: the insertion phase, abstraction level, activation mechanism, effect, location and physical characteristics. The insertion phase of the HTs may occur in either specification, design, fab, test or assembly. As for the abstraction level, the HTs may exist either at system or development, register-transfer level (RTL), gate or physical. The type of activation mechanism could be always-on with either being triggered internally or externally. The effect of the HTs could be either change functionality or degrade performance or leaking information or Denial of Service (DoS) attack. The component of ICs that contain HTs (location) can be either at processors, memory, power supply or clock grid. The physical characteristics of the HTs depends on either distribution, size, parameter, functionality or layout. All the elements of these six factors are used to assist the classification of HTs. By correctly identify types of the HTs, the techniques for the insertion and detection of these HTs can be methodically developed.

One of the techniques for identification and detection of HTs is the classification. The classification of HTs can be performed either using mathematical models such as algebraic matrix [3, 5] or machine leaning algorithms [7-12]. There are only few approaches of identifying HTs using machine learning- based classification have been developed such as in [7] compared to the approaches in detection of HTs [8-12]. This is because the identification of HTs are still immature since there are more HTs that are yet anticipated and recognized. As for the HTs detection, techniques such as frequency-domain power differences [9], reverse engineering (RE) [8] and macro synchronous micro asynchronous (MSMA) [10] are utilized. Then, the machine learning algorithms play role to enhance the detection by the above techniques. This process can be simplified by emphasizing on proper identification of the HTs using machine learning-based classification before the detection techniques are developed, thereby avoiding the redundant usage of machine learning based classification at the detection stage.

In this paper, an identification of HTs using machine learning based classification is proposed. The identification of HTs are based on the benchmark circuits in Trust-Hub [2]. The procedures of developing the machine learning-based identification of HTs are divided into three steps. The first steps are initial learning where the class of the HTs are set to its default HT design name. Then, the discretize algorithm is used for learning the features. Three machine learning algorithms were used for classification of the HTs.

The rest of the paper is organized as follows: Section II presents related work for identification and detection of HTs. The proposed work of developing a machine learning-based classification identification of HTs is explained in Section III. Section IV discusses result of using proposed algorithm. Finally, Section V concludes the finding and analysis of the

result.

II. Related Works

To further understand the motivation behind this study, the literatures that discuss the classification techniques for identification and detection of HTs either using machine learning-based algorithm are reviewed as follows:

A. Classifying for the Identification of HTs

The first attempt of using classification technique to identify HTs is made by [4] where the HTs were classified based on five attributes: design cycle phase, abstraction level, triggers, effects and physical location. By organizing a HT design competition on tertiary level, the diverse set of HTs are compiled and classified based on these five attributes. This initial dataset of HTs is further standardized in [2] by developing vulnerability analysis flow and detectability metric. The HTs are implemented based on the hard-to-detect areas that is determined by the vulnerability analysis flow. The detectability of the HTs are evaluated using Trojan detectability metric. In [3], a classification technique using algebraic method was developed to identify the missing attributes in HTs dataset from TrustHub. It is claimed that by using this technique, all HTs in TrustHub are properly classified compared to the classification technique in [2] and [4]. This technique was then automated using online tool called Hardware Trojan System (HTS) [5].

Another classification technique called score-based classification are developed in [6] to identify HT-free or HT-infected circuit without using golden netlist. Two types of class: weak and strong are developed for score-based technique. In the weak classification, Trojan nets are classified into nine cases and each case is given a score. The nets are classified as Trojan nets if they have maximum score that is less than 3, maximum constant cycle that is more than 999996 cycles and the maximum score net count that is less than 5. This score-based technique is claimed to be able to detect all HTs in selected benchmark circuits in TrustHub compared to UCI and VeriTrust techniques. For classification technique that is based on machine learning algorithm, SVM is used in [7] to identify between normal and Trojan nets in a set of gate level netlist.

The features that are used to classify these nets are logic gate fan-in (LGFi), flip-flop input (FFi), flip-flop output (FFo), primary input (PI) and primary output (PO). These extracted features are learned using SVM based on three conditions: no weighting, static weighting and dynamic weighting. For no weighting, SVM learned the normal and Trojan nets as their default quantities.

For static weighting, the SVM learned the normal nets as their default quantities while the Trojan nets as their original quantity was multiplied by weight, W. For dynamic weighting, the SVM learned the normal nets as their default quantities while the Trojan nets as the quantity of normal nets. The accuracy of identifying the HTs are 80% or higher with dynamic weighting.

B. Classifying for the Detection of HTs

Compared to HTs identification, there are more literatures on developing machine learning-based classification for HT detection. In [8], an SVM-based approach was developed to assist RE in detecting HTs. This approach eliminated the last two steps: annotation and schematic creation in RE. The features were extracted from the first three steps of RE wihout labels. To solve this, one class v-SVM is used as the class for this training sample. This type of SVM has values between '0' and '1' and these values were determined by the decision boundary that closely surrounds the training sample. This approach achieved higher accuracy with higher v and lower noise margin.

Another machine learning-based classification for HTs detection is developed in [9] by converting the differences in power consumption between HT-free and HT-infected circuits into frequency domain and this converted power consumption was used as the training data using SVM. This technique is able to detect the all HTs in the AES circuit. A self- learning framework was developed in [10] to detect HTs in IC. The framework was constructed by integrating the MSMA detection technique with machine learning algorithms such as decision tree (DT), K-Nearest Neighbor (KNN) and Bayesian classifiers. The power, delay, current and frequency of the golden IC were extracted as attributes and were trained using the stated machine learning algorithms. The model was then used along with MSMA during the testing phase to detect HTs. The achieved accuracies using each model was relatively high where the accuracies were 95.19% using DT, 93.5% using BC and 93.12% using KNN.

In [11], a run-time Trojan detection architecture for custom many-core was developed using KNN, DT, Linear Regression (LR) and SVM. There are four features: source core number, destination core number, packet transfer path and total distance at each router hop that are extracted in order to detect communication-based HTs such as traffic diversion, routing loop and core spoofing. After all these features were extracted, they were trained using KNN, DT, LR and SVM to evaluate their accuracy in detecting all the communication-based HTs. From the learnt model, two analyses are performed: accuracy analysis where showed that SVM and DT is the best (94%- 100%) and hardware complexity analysis where showed that SVM was the best option in term of computation and memory requirement. Thus SVM is selected for developing the HTs detection architecture for many-core platform. This SVMbased technique had 93% accuracy in detecting the mentioned communication-based HTs. This technique was further enhanced in [12] to secure design from new attacks introduced at real-time. To serve this purpose, Modified Balanced Winnow (MBW), online machine learning algorithm was utilized using the attach detection module (ADM). This enhanced technique has 5% to 8% higher detection accuracy for communication-based HTs compared to SVM and KNN.

Based on the above discussion, it can be seen that there is still lack of machine learning-based algorithm for identifying the HTs compared to detecting HTs. However, the utilization of machine learning-based classification in HTs detection are complex since it is tailored to the detection techniques that are used. Thus, this process can be simplified given that proper HTs identification are performed using the machine learning- based classification prior to the HTs detection.

III. Proposed Work

The method of classifying for the identification of the HTs using machine learning algorithms begins by tabulating the HTs in the Trust-Hub benchmark [2] as the training data. The taxonomies are used as the attributes where their contents are

represented by either '1' to indicate TRUE or '0' to signify FALSE. As for the attributes with all null values, they are omitted from the training data before the learning is performed. Once the training data is ready, it is learnt using the MATLAB Classification Learner Apps. The accuracy of the model is observed and improved by discretizing the class according to similar attributes. The learning is repeated using MATLAB Classification Learner Apps until best accuracy is achieved. After the best accuracy is achieved, the selected models are used for predicting analysis of hardware Trojan [4].

A. Process of Initial Learning of Hardware Trojan

Using MATLAB R2015a and later, the Statistics and Machine Learning Toolbox provide an apps called 'Classification Learner' which allow user to train and validate using different types of classifiers. The Classification Learner Apps has four available classifiers such as decision trees, SVM, KNN and ensembles.

The process initial learning of Hardware Trojan data using the Classification Learner Apps requires four (4) steps. The steps include:

Step 1: The user is required to select the dataset

Step 2: A response features are selected from one of the attributes while the other attributes are set as the predictors.

Step 3: User are required to select a validation model. In order to guarantee that the best model performance is acquired, the validation model must be decided before the training is executed. There are two types of validation: cross and holdout. In this experiment, the cross validation is selected since the dataset is small.

Step 4: Next, the classifier is selected for training and predicting analysis of the feature of HTs. Once the training was done, the result is shown based on the confusion matrix.

IV. RESULTS AND DISCUSSION

Originally, there are 12 group of HTs according to design modules in the Trust-Hub benchmark [2]. The design modules that are injected with the hardware Trojan are AES, b15, b19, basic RSA, EthernetMAC, 8051microcontroller, multi- pyramid, PIC16F84, RS232, scan flip-flops, vga-lcd and WISHBONE conmax. Since the accuracies using each group of HTs as the class is moderate, then to improve the classification accuracy of the features, the discretization methods is applied for HTs. Table 1 tabulates the classes of discretized group of HTs.

All the AES-based HT except AES-T1800, AES-T1900 and AES-T500 are grouped with BasicRSA-T100 and BasicRSA-T300 in Class 1 based on their "Leak Information' attributes. On the other hand, AES-T1800, AES-T1900 and AES-T500 are grouped with all the b15-based HT, BasicRSA- T200 and BasicRSA-T400 in Class 2 based on their commonality in 'DoS' attribute. As for Class 3, all the scan flip-flops based HT, vga-lcd-T100 and all PIC16F84-based HT are grouped together based on their commonality in 'Processor' attribute. Class 4 has all MultPyramid-based HTs, EthernetMAC10GE-T700, EthernetMAC10GE-T710, EthernetMAC10GE-T720 and EthernetMAC10GE-T730 which are grouped based on their "Fab" attribute. For Class 5, it contains all MC8051-based HT, b19-T300, b19-T400 and b19-T500 which are grouped based on their "RTL" attribute. In Class 6, the EthernetMAC10GE-T700, EthernetMAC10GE- T710, EthernetMAC10GE-T720, EthernetMAC10GE-T730, b19-T100, b19-T200, RS232-T1000, RS232-T1100, RS232- T1200, RS232-T1300, RS232-T1400, RS232-T1500 and RS232-T1600 are grouped together based on their "Change Functionality" attribute. Finally, Class 7 comprises of RS232- T1700, RS232-T1800, RS232-T1900, RS232-T2000, RS232- T200, RS232-T300, RS232-T400, RS232-T500, RS232-T600, RS232-T700, RS232-T800, RS232-T900, RS232-T901 and all wb_conmax-based HT which are grouped based on "Internally Triggered" attribute.

Table 1 HTS Discretization

Discretized HT Class	Attributes	HT Group
Class 1	Leak Information	1. All AES-based HT except AES-T1800, AES-T1900, AES- T500 2. BasicRSA-T100 and BasicRSA-T300
Class 2	DOS	1. AES-T1800, AES-T1900, AES-T500 2. All b15-based HT 3. Basic RSA-T200 and BasicRSA-T400
Class 3	Processor	1. All flip-flop - based HT 2. VGA-LCD-T100 3. All PIC16F84-based HT
Class 4	Fab	1. All EthernetMAC10GE-based HT except EthernetMAC10GE-T700, EthernetMAC10GE-T710, EthernetMAC10GE-T720 and EthernetMAC10GE-T730 2.MultPyramid based HT
Class 5	RTL	1. b19-T300, b19-T400 and b19- T500 2. All MC8051-based HT
Class 6	Change functionality	1. EthernetMAC10GE-T700, EthernetMAC10GE-T710, EthernetMAC10GE-T720 and EthernetMAC10GE-T730 2. b19-T100 and b19-T200 3. RS232-T1000, RS232-T1100, RS232-T1200, RS232-T1300, RS232-T1400, RS232-T1500 and RS232-T1600
Class 7	Internal Triggered	1. RS232-T1700, RS232-T1800, RS232-T1900, RS232-T2000, RS232-T200, RS232-T300, RS232-T400, RS232-T500, RS232-T600, RS232-T700, RS232-T800, RS232-T900 and RS232-T901 2. All wb_conmax-based HT

Now that all the hardware Trojan are discretized according their dominance in certain attributes, they will be classified using the Classification Learner Apps tool to see whether their accuracies are improved or not. Table 2 shows the result accuracy for HTs. The table tabulates the achieved accuracy for each decision tree, SVM and KNN models. It can be seen

from the table that the best accuracy for the decision tree model is 76.1 %. As for the SVM model, the best accuracy is 70.7%, while the best accuracy for the KNN model is 71.7% These accuracies must be further improved to ensure the HTs features are classified correctly.

Table 2
Accuracy of HTs

Model	Classification Accuracy (%)
Decision Trees (medium split)	76.1
SVM (quadratic kernel)	70.1
KNN (weighted K=10)	71.1

A. *Hardware Trojan Classification using Discretization Algorithm*

Table 3 shows the comparison result classification based on discretization algorithm.

Table 3
Classification Accuracy with Discretization Algorithm

Model	Classification Accuracy (%)
Decision Trees (medium split)	87.0
SVM (quadratic kernel)	85.9
KNN (weighted K=10)	89.1

Based on Table 3, it shows that the result accuracy is improved for each decision tree, SVM and KNN models after applying the discretization algorithm with the three classifiers. It can be seen from the table that the best accuracy for the decision tree model now rises to 87.0%. As for the SVM model, the best accuracy is improved to 85.9%. The best accuracy for the KNN model is climbed to 89.1%. It can be seen from the table that all the accuracies for each classification model are improved by 14% to 25%. With these improved accuracies, the models are ready to be used for predicting analysis.

Then, for predicting analysis, a total of 37 HTs are extracted from ten finalists of Embedded System Challenge 2008 [4]. These finalists were asked to design HTs for crypto-hardware platform, Alpha that implemented the Advanced Encryption Standard (AES) on Digilent BASYS Spartan-3 FPGA board. The main processor on the board interacted with Alpha through an RS232 serial port. There are 256 shared secret keys are hard coded into Alpha. Leaking these secret keys or obtaining the unencrypted messages is the objective of the finalists as the attackers. The user of the device selected a private key using the switches. The encrypted data was sent through the RS-232 port. Alpha emulated a real world crypto accelerator, typically used to secure communications in a hostile environment.

All the HTs that are designed by the finalists are made for the predicting analysis. The data for predicting analysis will be based on the test data. All the features of HTs are classified with the three (3) classifiers: Decision Tree (DT), Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) with discretization algorithm. The label for predicted class comes from class attribute includes leak information, DoS, Processor, Fab, change functionality and internal triggered.

B. *Predicting Analysis based Decision Trees Classifier*

The result for predicting analysis using a test data is depicted in a Table 4. The table tabulates the dominant attribute matching of the predicted class for test data using decision tree learnt model. It can be seen that there are 26 rows of test data are classified as Class 1, 2 rows of test data are classified as Class 2, 1 row is classified as Class 5 and 7 rows are classified as Class 7. All 26 of test data rows that are predicted as Class 1 match the dominant attribute, 'Leak information'. This is also true for test data rows that are predicted as Class 2 and Class 5 where all of them match the dominant attributes, 'DoS' and 'RTL respectively. However, for test data rows that are predicted as Class 7, only 1 out of 7 rows matches the dominant attribute, 'Internally Triggered'.

Table 4
Dominant Attributes Matching based on Decision Tree (DT) Classifier

Predicted Class	Dominant Attribute for the Class	Match to Dominant Attribute
Class 1	Leak Information	26 out of 26
Class 2	DoS	2 out of 2
Class 5	RTL	1 out of 1
Class 7	Internally Triggered	1 out of 7

C. *Predicting Analysis Based SVM Classifier*

The result for predicting analysis using a test data is depicted in a Table 5. The table tabulates the dominant attribute matching of the predicted class for test data using SVM learnt model. It can be seen that there are 27 rows of test data are classified as Class 1, 5 rows of test data are classified as Class 2, 3 row are classified as Class 3 and 1 rows are classified as Class 6. From 26 of test data rows that are predicted as Class 1, 21 of them match the dominant attribute, 'Leak information'. As for test data rows that are predicted as Class 3, only 1 out of 3 rows matches the dominant attribute, 'Processor'. However, for test data rows that predicted as Class 2 and Class 6, none of them matches the dominant attributes, 'DoS' and 'Change Functionality'.

Table 5
Dominant Attributes Matching based SVM Classifier

Predicted Class	Dominant Attribute for the Class	Match to Dominant Attribute
Class 1	Leak Information	21 out of 27
Class 2	DoS	0 out of 5
Class 3	Processor	1 out of 3
Class 6	Change Functionality	0 out of 1

D. *Predicting Analysis based K-Nearest Neighbor*

The result for predicting analysis using a test data is depicted in a Table 6. The table tabulates the dominant attribute matching of the predicted class for test data using decision tree learnt model. It can be seen that there are 25 rows of test data are classified as Class 1, 2 rows of test data are classified as Class 2, 2 rows are classified as Class 5 and 7 rows are classified as Class 7. All 25 of test data rows that are predicted as Class 1 matches the dominant attribute, 'Leak information'. This is also true for test data rows that are predicted as Class 2 and Class 5 where all of them match the dominant attributes, 'DoS' and 'RTL respectively. However, for test data rows that are predicted as Class 7, only 1 out of 7 rows matches the dominant attribute, 'Internally Triggered'.

Table 6
Dominant Attribute Matching based KNN

Predicted Class	Dominant Attribute for the Class	Match to Dominant Attribute
Class 1	Leak Information	25 out of 25
Class 2	DoS	2 out of 2
Class 5	RTL	2 out of 2
Class 7	Internally Triggered	1 out of 7

V. CONCLUSION

Based on the results, it is concluded that the machine-learning-based classification was successfully developed for HTs identification. About 83% of test data were successfully predicted by both DT and KNN algorithm based on the dominant attributes for each class. As for the SVM, it successfully predicted about 63% of the test data. These results match with the classification accuracies of the learnt models where SVM model had less accuracy than DT and KNN model

ACKNOWLEDGMENT

The authors would like to thank Ministry of Higher Education (MOHE) and Universiti Teknologi Malaysia (UTM) for their educational and financial support. This work is conducted under CyberPhysical Systems Research Group (CPS RG) and funded by Universiti Teknologi Malaysia (GUP Tier 2 Grant no. Q.K130000.2638.12J34).

REFERENCES

[1] H. Li, Q. Liu, and J. Zhang, "A survey of hardware trojan threat and defense," *Integration, the VLSI Journal*, vol. 55, pp. 426-437, Sep. 2016.

[2] H. Salmani, M. Tehranipoor, and R. Karri, "On design vulnerability analysis and trust benchmark development," in *IEEE 31st International Conference on Computer Design*, 2013, pp. 471-474.

[3] S. Moein, S. Khan, T.A Gulliver, F. Gebali, and M.W El-Kharashi, "An attribute based classification of hardware Trojans," in *IEEE 10th International Conference on Computer Engineering and System*, 2015, pp. 351-356.

[4] H. Salmani, M. Tehranipoor, and R. Karri, "On design vulnerability analysis and trust benchmark development," in IEEE *31st International Conference on Computer Design*, 2013, pp. 471-474.

[5] N. Houghton, S. Moein, F. Gebali, and T. A. Gulliver, "An automated web tool for hardware classification," in *CSREA International Conference on Embedded Systems, Cyber-physical Systems & Applications*, 2016, pp. 89-94.

[6] M. Oya, Y. Shi, M. Yanagisawa, and N. Togawa, "A score-based classification method for identifying hardware Trojans at gate level Netlist," in *IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, pp. 465-470.

[7] K. Hasegawa, M. Oya, M. Yanagisawa, and N. Tagawa, "Hardware Trojans classification for gate level Netlist based on machine learning," in *IEEE 22nd International Symposium on On-line testing and Robust System Design (IOLTS)*, 2016, pp. 203-206.

[8] C. Bao, D. Forte, and A. Srivastava, "On application of one-class SVM to reverse engineering-based hardware Trojan detection," in *IEEE 15th International Symposium pn Quality Electronic Design*, 2014, pp. 47-54.

[9] T. Iwase, Y. Nozaki, M. Yoshikawa, and T. Kumaki, "Detection technique for hardware Trojans using machine learning in frequency domain," in *IEEE 4th Global Conference on Consumer Electronics*, 2015, pp. 185-186.

[10] F.K Lodhi, I. Abbasi, F. Khalid, O. Hassan, F. Awwad, and S.R Hassan, "A self-learning framework to detect the intruded integrated circuits," in *IEEE International Symposium on Circuits and Systems, 2016*, pp. 1702-1705.

[11] A. Kulkarni, Y. Pino, and T. Mohsenin, "SVM-based real-time hardware Trojan detection for many-core platform," in *IEEE 17th International Symposium on Quality Electronic Design*, 2016, pp. 362-367.

[12] A. Kulkarni, Y. Pino, and T. Mohsenin, "Trojan detection framework through machine learning," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2016, pp. 120-123.