Machine Learning Based Hardware Trojan Detection Using Electromagnetic Emanation

Junko Takahashi[1(✉)], Keiichi Okabe[1], Hiroki Itoh[1], Xuan-Thuy Ngo[2(✉)], Sylvain Guilley[2], Ritu-Ranjan Shrivastwa[2], Mushir Ahmed[2], and Patrick Lejoly[2]

[1] NTT Secure Platform Laboratories, Tokyo, Japan
junko.takahashi.fc@hco.ntt.co.jp
[2] Secure-IC, Cesson-Sevigne, France
thuy.ngo@secure-ic.com

Abstract. The complexity and outsourcing trend of modern System-on-Chips (SoC) has made Hardware Trojan (HT) a real threat for the SoC security. In the state-of-the-art, many techniques have been proposed in order to detect the HT insertion. Side-channel based methods emerge as a good approach used for the HT detection. They can extract any difference in the power consumption, electromagnetic (EM) emanation, delay propagation, etc. caused by the HT insertion/modification in the genuine design. Therefore, they can be applied to detect the HT even when it is not activated. However, these methods are evaluated on overly simple design prototypes such as AES coprocessors. Moreover, the analytical approach used for these methods is limited by some statistical metrics such as the direct comparison of EM traces or the T-test coefficients. In this paper, we propose two new detection methodologies based on Machine Learning algorithms. The first method consists in applying the supervised Machine Learning (ML) algorithms on raw EM traces for the classification and detection of HT. It offers a detection rate close to 90% and false negative smaller than 5%. For the second method, we propose a method based on the Outlier/Novelty algorithms. This method combined with the T-test based signal processing technique, when compared with state-of-the-art, offers a better performance with a detection rate close to 100% and a false positive smaller than 1%. We have evaluated the performance of our method on a complex target design: RISC-V generic processors. The three HTs with the corresponding sizes of 0.53%, 0.27% and 0.1% of the RISC-V processors are inserted for the experimentation. The experimental results show that the inserted HTs, though minimalist, can be detected using our new methodology.

Keywords: Hardware trojan · Electromagnetic · Side-channel analysis · Machine learning · Outliers detection

W. Meng et al. (Eds.): ICICS 2020, LNCS 12282, pp. 3–19, 2020.
https://doi.org/10.1007/978-3-030-61078-4_1

1 Introduction

1.1 Hardware Trojan Threat

The semiconductor industry has spread across borders in this time of globalization. Different design phases of an Integrated Circuit (IC) may be performed at geographically dispersed locations. Outsourcing the IC design and fabrication to increase profitability has become a common trend in the semiconductor industry. As more and more semiconductor companies are welcoming the outsourcing trend to be competitive, they are opening new security loopholes. One such threat that has come into light over the past few years is that of Hardware Trojan (HT). A HT is a malicious module inserted in an IC during the design or fabrication stage. Once inserted, a HT can perform dangerous attacks such as Denial of Service (DoS), leakage of sensitive data via circuit outputs, etc. [11]. It can be implemented in ASIC, microprocessor, microcontroller, GPU, DSP and also in FPGA bitstreams.

HTs can be inserted along the IC design flow from the specification phase to the assembly and the package phase. Different examples of the presence of HTs are discovered in different industrial applications. Skorobogatov et al. discovered an undocumented backdoor inserted into the Actel/Microsemi ProASIC3 chips (military grade chip) for accessing FPGA configuration [20] in 2012. Using this HT, an attacker is able to extract all the configuration data from the chip, reprogram crypto and access keys, modify low-level silicon features and finally access to the configuration bitstream or permanently damage the device. In 2014, the discovery of specific US-made components designed to intercept the satellites communications in France-UAE satellite has been reported in the news on www.rt.com. Different documents leaked in 2014 by NSA whistleblower Edward Snowden indicate that the NSA planted back-doors in Cisco routers and hence had been able to gain access to entire networks and all their users. Routers, switches, and servers made by Cisco are booby-trapped with surveillance equipment that intercept traffic handled by those devices and copy it to the NSA's network. And recently, in October 2018 Bloomberg reported that an attack by Chinese spies reached almost 30 U.S. companies, including Amazon and Apple, by compromising America's supply-chain technology. We can also find many other examples in the academic works such as in [11,12,16] etc. Because of its malicious and dangerous natures, a HT can create serious problems in many critical applications such as military systems, financial infrastructures, health applications, IoTs etc. Therefore, many national and international projects are launched to develop the countermeasures such as TRUST & Microsystems Exploration program (in USA), HINT (in Europe), HOMERE & MOOSIC (in France). This threat is also a big concern for all other countries.

1.2 Related Studies

Since HTs pose serious threats in the IC manufacturing, they have become a very important and key research topic. Covered areas are: threat analysis, HTs architecture, prevention and detection methods. Regarding HT detection, numerous

methods and approaches have been proposed in the state-of-the-art. To mention a few, optical methods [6,22], testing based detection methods [3,10], run-time based detection [17] or side-channel based detection methods [2,18,21]. Among these approaches, side-channel based detection methods seem to be the most suitable approach for various reasons. First of all, side-channel methods are non-invasive and unlike optical methods they do not require chip chemical preparation. Second, they can work without the need of additional logic for run-time detection. Third and most important, efficiency in detection is relatively high. The side-channel based detection methods can detect HTs even if they are not activated during the experimental process.

In the state-of-the-art, different works have been proposed to detect purported HTs using side-channel analysis. In [18], the authors propose an Electro-Magnetic (EM) cartography detection method. The experiment has been performed on an FPGA and the detection method is based on the visual comparison of T-test coefficient between the genuine and infected design. In [8], the authors have used a golden chip-free EM side-channel methodology to detect the HT. Their technique has been limited to utilize the difference in the response between the simulated trace and chip's actual traces from the experiments. In [13], the authors propose a method based on the integration of sensor matrix used to measure the supply voltage in the circuit and T-test metric. The test is performed on a 128-bits AES and validated on a HT with an overhead of 3.2% of the target FPGA. Using the T-test, they obtained a success rate of 80%. In [24], the authors also propose a detection method based on a Ring Oscillators (ROs) matrix (used to measure the power) combined with supervised machine learning (ML) methods such as K-Nearest Neighbors and SVM. With this approach, they have a success rate greater than 88%.

1.3 Contributions

In this paper, we propose new HT detection methodologies based on the ML algorithms combined with the side-channel measurements. The first method consists in applying the supervised machine learning algorithms on the raw EM traces for the HT detection. And the second method consists in combining the Outlier detection algorithms with the T-test preprocessing technique for the HT detection. It presents several new advantages in comparison to those in the state-of-the-art. First, many papers used statistical metrics for the detection or the visual comparison between the genuine and infected designs [1,18]. However, these metrics are dependent on selected samples for the test. They also depend upon the measurement setups. For example, in the case of EM traces, the position of the EM probe affects the performance of the statistical metrics. Moreover, they need to decide manually a threshold for the detection using these metrics. So the selected samples and threshold can modify significantly the detection rate. There are also some works that have applied the classification ML methods for the HTs detection [15]. But the performance of these methods depends upon the dataset used for the training. With our new method, we can automatically detect the HTs without the need to pay attention to the selected samples and

threshold. Moreover, with the second method, we need only one dataset for the training phase. Then, we can test with all different datasets coming from genuine or HT designs. The method predicts if a test dataset is the same as the training dataset (Inlier/Genuine) or not (Outliers/HT). This can be very useful in the case where we have only the genuine dataset (from a genuine design or from simulation). It can also be applied in the case where we want to detect if two chip batches are the same or not. It can happen that a supplier ships two different chip batches for two different countries because of his/her government order for the goal of security and/or monitoring.

Second, for the HT detection, the detection rate is very important. All the proposed methods in the state-of-the-art have either no detection rate evaluation or a detection rate smaller than 70% even using the statistical approach. With our first method using the supervised ML algorithm, we obtain a detection performance of 90%. And with the second method, we propose to combine the processing method (T-test) and the outlier detection algorithms to obtain a very high detection rate (nearly 100%).

Third, all the methods described in the state-of-the-art are tested and evaluated only on some cryptographic co-processors such as AES or DES. In this paper, we show that our new method can be applied successfully on two complex and generic targets: **PicoRV** and **Freedom** RISC-V based processors. Three different HTs with the corresponding sizes of 0.56%, 0.27% and 0.1% are implemented on the DE1 SoC Cyclone V FPGA and the Arty-7 FPGA for the experimentation. Different outlier/novelty detection algorithms such as One Class SVM, Elliptic Envelope, Isolation Forest and Local Outlier Factor are also applied/evaluated for our methodology. The results have shown a considerable performance in the HT detection, i.e. with a probability of 100%. It validates the efficiency of our method for detecting even minuscule HTs (with an overhead of 0.1%).

2 Backgrounds

In this section, we have listed out different techniques that are central to the detection of HTs after EM measurements. These metrics form a very important step as they can enhance the reproducibility and robustness of the detection techniques.

2.1 T-Test Metric

T-test (or Student test) is a metric used in the field of statistics to detect if the mean of a population has a value specified in a null hypothesis or if the means of two different populations are equal. For the HT application, the T-test is already used in the state of the art to determine if the reference dataset and the dataset under test have the same means (no HT) or not (HT) using the following formula:

$$t = \frac{\mu_0 - \mu_1}{\sqrt{\frac{\sigma_0^2}{N_0} + \frac{\sigma_1^2}{N_1}}}$$

where μ_0 is the genuine sample mean, μ_1 is the HT sample mean. σ_0 is the genuine sample variance, σ_1 is the HT sample variance. N_0 is the cardinality of genuine set and N_1 is the cardinality of HT set. The T-test is also used for the side-channel analysis to break the cryptography IPs [7]. In this paper, we will evaluate the performance of the T-test metric based detection method for our test platform in order to show its limitation and drawback.

2.2 Supervised Machine Learning Method

Supervised learning method is used to map an input to an output based on known input-output pairs also called training database. Each input-output pair is composed of an input data and a desired output value. The supervised learning algorithm analyzes the training database in order to produce a model used for mapping new test data with the predefined outputs. An optimal trained model allows for the algorithm to correctly determine the class labels for unseen or undetected instances. The supervised ML algorithms are widely used for the classification and detection analysis. Here are some examples of supervised ML algorithms.

- **Support Vector Machine** analyzes data used for classification and regression analysis. Basically, the SVM constructs a hyperplane or a set of hyperplanes in a high dimensional space which can be used for classification, regression, or other tasks like outliers detection. During the training phase, the SVM tries to find the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin) [9].
- **Multi-Layer Perceptron** is a class of feed-forward artificial neural network (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a non-linear activation function. MLP utilizes a supervised learning technique called back-propagation for training a multi-layer Perceptron. It is a linear function that maps the weighted inputs to the output of each neuron.
- **Decision Tree Classification** algorithm creates tree models where the target variables can take a discrete set of values which are called classification trees. In these structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.
- **K-Nearest Neighbors** is a non-parametric method used for classification and regression. In K-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor.

In this work, we will evaluate the performance of these methods on our platform.

2.3 Outlier/Novelty Detection Method

Outlier/Novelty detection are a sub ML class used to detect abnormal/unusual observations or data. Outlier detection uses an unsupervised learning process to detect the outliers and filter the impurities in a dataset. Novelty detection is a semi-supervised ML method used to form a dense cluster of the data as long as they are in a low density region of the training data, considered as normal in this context. For our HT detection method, the genuine datasets are used to train the model for outlier/Novelty detection algorithms. Once the model is fixed, we can test it with new data. If the new data is considered as an outlier, it means that this data is generated from a HT design, else this data is generated from a genuine design. For our method, the following algorithms have been tested/evaluated:

- **One Class SVM** this SVM is trained on data that has only one class, which is the "normal" class. It infers the properties of normal cases and from these properties, it is able to predict which test cases are unlike the normal case [4].
- **Isolation-Forest** builds a set of trees for a given data set. These trees are also known as iTrees form the basis of detection of anomalies. It isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature [14].
- **Elliptical Envelope** models the data as a high dimensional Gaussian distribution with possible co-variances between feature dimensions. It attempts to find a boundary ellipse that contains most of the data. Any data outside of the ellipse is classified as anomalous.
- **Local Outlier Factor** is an unsupervised anomaly detection method which computes the local density deviation of a given data point with respect to its neighbors. It considers the test samples as outliers that have a substantially lower density than their neighbors [5].

These outlier detection algorithms will be integrated in our new HT detection methodology. All these 4 algorithms will be tested in order to select the best ones for the HT detection scenario (Sect. 5.2).

3 Experimentation Platform

3.1 Target Designs

RISC-V processors are used as the reference design which embeds the HT to show the effectiveness of the detection method. RISC-V is an open-source hardware instruction set architecture defined by the University of Berkeley. In this project, we select two RISC-V implementations "PicoRV32" [23] and "Freedom E310" [19] for the experimentation. These two designs are 32-bit RISC-V processors. For the experiment, two test boards are also selected. The PicoRV32 processor is implemented and evaluated on the DE1-SoC board with a Cyclone V FPGA. And the freedom E310 processor is implemented and evaluated on the Arty-7 FPGA board.

3.2 Hardware Trojan Designs

As examples, to show the effectiveness of the proposed detection scheme, we implement different HTs with different sizes in the RISC-V processors. In the experiments, three HTs (HT1, HT2 and HT3) are inserted in two target designs. The HT1 and HT2 are inserted in the PicoRV32 design. The triggers of the HT1 and HT2 are based on specific DIV instructions. Once activated, the two HTs (HT1 and HT2) are able to modify arbitrarily the program counter of the processor. These two HTs are inserted at Place & Route level. It means that the reference design (without HT) and infected design (with HT) have the same layout except where the HTs are inserted. The overhead of the HT1 and HT2 are respectively 0.53% and 0.27% of the reference design (PicoRV32 processor).

The HT3 is inserted in the Freedom processor. The trigger of the HT3 is also based on the specific DIV instruction. Once activated, the HT3 modifies arbitrarily the privileged level of the processor hence performing a privilege escalation attack. This HT is inserted at the RTL by modifying directly the HDL code of the design. The overhead of the HT3 is 0.1% of the reference design (Freedom processor). The Table 1 resumes the 3 HTs used for the experiment.

Table 1. HT designs for the experimentation on RISC-V processors

	Target design	Insertion phase	Trigger	Payload	Overhead
HT1	PicoRV32	P& R	Specific Instruction	Modify PC	0.53%
HT2	PicoRV32	P& R	Specific Instruction	Modify PC	0.27%
HT3	Freedom	RTL	Specific Instruction	Modify privilege level	0.1%

3.3 Measurement Platform

Our detection method is applied and tested on the side-channel information/trace by deploying the EM measurement techniques. It measures the EM emanated trace of a test design and compares it with the reference trace (captured from a genuine design). Then we extract any difference purportedly created by the HT insertion. The EM acquisition platform is composed of the EM Langer probe (for EM signal capture), Langer preamplifier (for amplifying the EM signal), a 3D axis table (for cartography position) and a KEYSIGHT scope (for traces acquisition).

For the EM traces acquisition, we capture the EM emanation of the chip during its operation. In a real HT detection scenario, we cannot know which program (or which mechanism) is used to activate the HT. Therefore, we just use a normal test program that *does not activate* the HT but still we try *to detect* it using the EM side-channel analysis. In the test program, we execute just two sequences of 100 “NOP” and then 100 “incrementation” instructions. This program is used for traces acquisition and during all the tests—note that the three HTs are **never activated**.

Figure 1 presents the overview of the RISC-V development process and our cartography setup. We define the RISC-V design using an HDL (Verilog) implementation, then we synthesize and place-and-route the design to obtain the floorplan and finally generate the bitstream. After these design steps, we implement the bitstream in the corresponding FPGA circuits (Cyclone V for PicoRV32 and Arty-7 for Freedom).

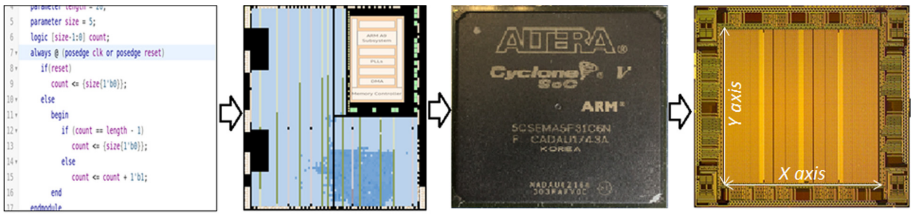

Fig. 1. Cartography overview

In this experiment, we have used the EM cartography in order to measure the EM emanations on each area of the FPGA chip (recall Fig. 1). One cartography consists of performing multiple measurements at several points on the target circuit. We perform a 2D cartography automated by an XY moving stage. In order to cover the whole FPGA, we perform N_x steps of 2 mm (for DE1 SoC board) and 1 mm (for Arty-7 board) along X-axis and N_y steps (2 mm for DE1 SoC and 1 mm for Arty board) along Y-axis. So for one cartography, we have $P = N_x \times N_y$ measurement points. For each measurement point, we have acquired N EM traces where N is the number of cartographies. Finally, each EM trace contains T temporal samples.

For the DE1 SoC board, we have performed $N = 50$ cartographies of size $N_x \times N_y$ where $N_x = 13$ and $N_y = 13$ (i.e., $P = 169$) for each design. The acquired traces consist in $T = 5000$ temporal samples. It means that, for each measurement point (between 169 positions) of cartography, we repeat the measurement 50 times and each time, we will storage an EM trace of 5000 samples. Figure 4 gives an overview of this dataset (on the left) where $N = 50$ represents the number of cartographies, 13×13 represents the N_x steps of 2 mm along X-axis and N_y steps along Y-axis of the cartography and 5000 is the amount of samples of each EM trace.

And for the Arty board, we performed $N = 50$ cartographies of size $N_x \times N_y$ where $N_x = 10$ and $N_y = 10$ (i.e., $P = 100$) for each design. The acquired traces consist in $T = 5000$ temporal samples.

4 Detection Results of State-of-the-Art Methods

Raw EM Traces Comparison. First straightforward approach is comparing the EM traces of genuine and infected designs and trying to detect visually the difference created by the HT insertion.

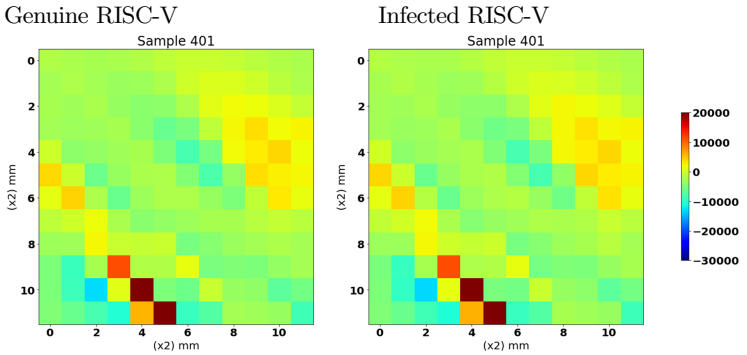

Fig. 2. EM cartography of Genuine PicoRV32 (left) and HT1 Infected RISC-V (right) for sample 401

Figure 2 presents an example of the EM cartography result for the temporal sample 401 of the genuine design (PicoRV32 on the left) and of the HT1 infected design (presented on the right). Here, we can notice that it is not possible to distinguish the difference created by the HT insertion. The same comparisons for other temporal samples and for other HTs (HT2 and HT3) give the same results. In conclusion, we cannot detect the HT inserted in the RISC-V processor just based on visual comparison of the raw traces.

HTs Detection Based on T-Test Metric. The second approach in the state of the art is using the T-test metric for the detection (See Sect. 2).

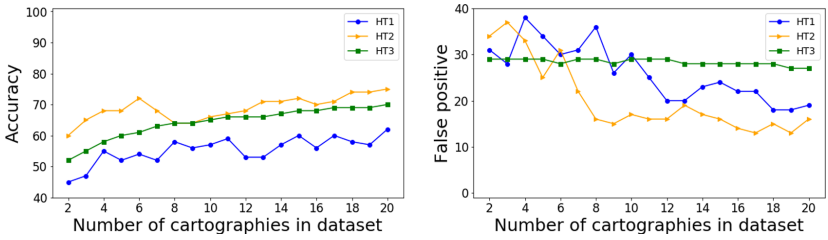

Fig. 3. HTs detection using T-test metric

In order to evaluate the performance of T-test detection methods, we evaluate its detection rate for different numbers of cartographies which are selected for

the T-test computation. The detection results of the T-test metric are presented in Fig. 3. It presents the detection rate and the false positive of the T-test based detection method for the 3 HTs in function of the number of cartographies used for the T-test computation. For the detection, we use a parameter c as the threshold coefficient for the detection. In each case, if the corresponding T-test value for one selected measurement point and one selected temporal sample is greater than $c = 2.0$ times of the reference T-test value, it can be considered as a HT. We can notice (in Fig. 3) that the detection results for all the 3 HTs are between 55% and 75% with a very high false positive rate (between 15% and 30%). So we can infer that the T-test performance is very low. Moreover, the performance of the T-test metric also depends upon the selected temporal sample for the test. In some specific samples, we can observe a big difference between the T-test coefficient of the genuine design and the infected design. But in many other samples, we cannot see the difference between them. The measurement point of the cartography can also impact the detection results.

One-Class SVM on EM Raw Traces [15]**.** This method consists is applying the One-Class SVM method on the raw power traces for the detection. They tested this method with the traces acquired from an AES design. In our case, we reevaluate the performance of this method with our EM traces of RISC-v design. For the test, we use 40 cartographies of reference RISC-V design for the training phase. And we use 10 cartographies of reference RISC-V design for the training phase and 40 cartographies of infected design for the test phase. With this dataset, we obtain a false positive and false negative of 40%. So for the moment, this method is not working with our design.

In conclusion, the statistical T-test metric and the One Class SVM based method have poor performances for the detection of these HTs and it also depends upon many parameters. In the following sections, we present our new methods based on ML algorithms to improve the detection rate.

5 New Methods Based on Machine Learning Algorithms

5.1 Supervised Machine Learning Based HT Detection Methods

Methodology. For the first method, we apply directly the supervised machine learning algorithms for the HT detection using EM raw traces. The methodology of this method is composed of the following steps:

1. Acquire the EM traces of reference design (EM_{ref}) and HT design (EM_{HT})
2. Use these EM traces (EM_{ref} and EM_{HT}) to train the supervised machine learning algorithms
3. Acquire the EM traces of test design EM_{test}
4. Apply the trained models on EM_{test}, the models will decide if the test design is the same than reference or HT design

Detection Results. Figure 4 presents the data format used for the ML based method for the HT1 and HT2 on DE1 SoC board. In this figure, N represents the number of cartographies, 13×13 represents the N_x steps of 2 mm along X-axis and N_y steps along Y-axis of the cartography and 5000 is the amount of samples of each EM trace. In our method, we have used the cartography of each sample as the input. Therefore, for one cartography, we have 5000 input vectors of length 169 (5000×169) for HT1 & HT2. For the HT3 on Arty board, we have 5000 input vectors of length 100 (5000×100).

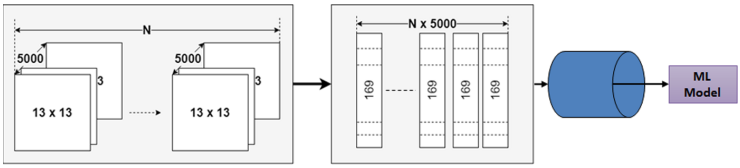

Fig. 4. Data preparation for the machine learning methods

Then we compute the detection results using all 5 supervised ML algorithms: Support Vector Machine (SVM), Multilayer Perceptron (MLP), Decision Tree Classification (DTC), Linear Regression Classification (LRC) and K-nearest neighbors classification (KNNC). In order to evaluate the performance of these methods, we evaluate the *detection probability* and *proportion of false positives* of the methods when we vary the number of cartographies used in the training phase. It means that we have to select a number of x cartographies amongst the 50 cartographies of genuine dataset and HTs (HT1 & HT2 and HT3) infected datasets. And the number of cartographies used for the detection phase is $50-x$. For this test, we vary the number of x from 1 to 22 cartographies (amongst the 50 cartographies of each datasets).

The results are presented in Fig. 5. In this test, the detection rate of SVM and MLP are greater than 90% or even above with varying number of cartographies in the training phase from 1 to 20 (refer to Fig. 5). And the false positive is smaller than 4%. (refer to Fig. 5). For the DTC, the detection rates for the HT1 and HT2 are greater than 95% with a false positive smaller than 1%. However, the detection rate for the HT3 is smaller than 80% with a false positive of 4%. With the LRC, the detection rate of the HT1 is good (greater than 90%) with a false positive of 3%. However, the detection results for HT2 and HT3 are poor (between 50% to 60%) with a false positive of 20%. For the last algorithm (KNNC), the detection rates of these 3 HTs are greater than 80% with a false positive smaller than 3%. So, for the moment the SVM and MLP based detection methods can detect the HTs with a good success compared to the T-test based method.

One of the drawbacks of the supervised ML based methods is that we need the dataset of the infected design. In the real case, it could be difficult to have these HT datasets. And for different HTs with different datasets, the detection

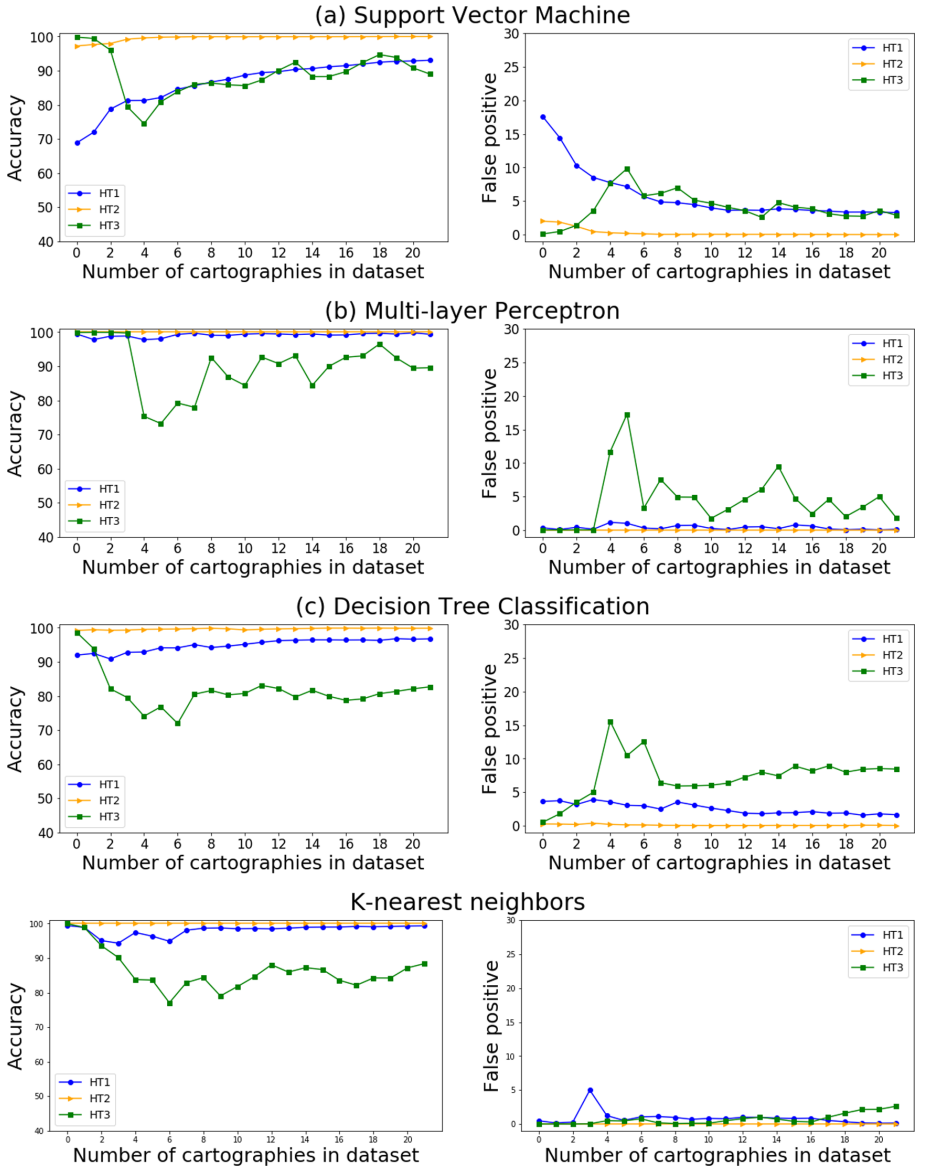

Fig. 5. HTs detection using supervised machine learning algorithms

results can be different. Each time, when we have a new HT design, we might need to re-train our model. For this reason, we have applied other detection methods based on the outliers detection.

5.2 Outlier Detection Based Method

Methodology. In order to solve the problems of supervised ML based detection method, we propose a new detection method using the novelty/outliers detection. The goal of these methods is to detect whether a new observation belongs to the same distribution as existing observations (an Inlier), or should be considered as different (an outlier). For the HT detection, we can apply these algorithms for two scenarios:

- When we want to detect if two different chip batches are the same or not. It can detect any difference between two chip batches.
- When we want to detect if a test data (from a test chip) belongs to the same distribution as the reference observations (from the reference chips). If the test data is detected as an outlier, we can conclude that there is some modification (HT) on the test chip.

For the first test, we apply the same methodology as the one used by the first proposed method using the supervised machine learning methods. But in this time, we replace the supervised methods by the outlier detection methods (Sect. 2.3). In this case, we use the raw EM values for the training phase and detection phase as described in Fig. 4. However, using the raw EM values as input, we obtained a low performance with a false positive and false negative between 60 to 40% depending on the selected outlier detection algorithms.

In order to increase the performance, we propose a new approach with the combination of T-test and outlier detection methods. With the raw data, the EM value of each sample could be very different. But, using the T-test for the pre-processing, we will evaluate only the variance at each sample therefore the data in different samples will have the same scale. In this case, we will use the T-test coefficients instead of raw EM values for the input of our outlier detection algorithms. The methodology of our method is composed of the following steps:

1. Acquire the EM traces of the reference design
2. Compute the T-test value of the reference design (T_{ref})
3. Train the Outliers detection algorithms using the T-test value (T_{ref})
4. Acquire the EM trace of test design
5. Compute the T-test value of the test design (T_{test})
6. Test the trained Outlier detection algorithms with (T_{test}) to decide if the test design is the same (or not) than the reference design

Detection Results. For the test, we evaluate the performance of these algorithms by varying the number of cartographies used for T-test computation from 3 to 22 cartographies (amongst the 50 cartographies of each dataset). Then we apply 4 outliers detection methods for our experimentation: One-Class SVM, Elliptical Envelope, Isolation Forest and Local Outlier Factor. The detection results of the 3 HTs using these methods are presented in Fig. 6. The obtained results show that the detection rates of all selected algorithms are greater than 95%. Particularly, the detection rate of One-Class SVM, Elliptical Envelope and Local Outlier Detection are close to 100%. However, the One-Class SVM has

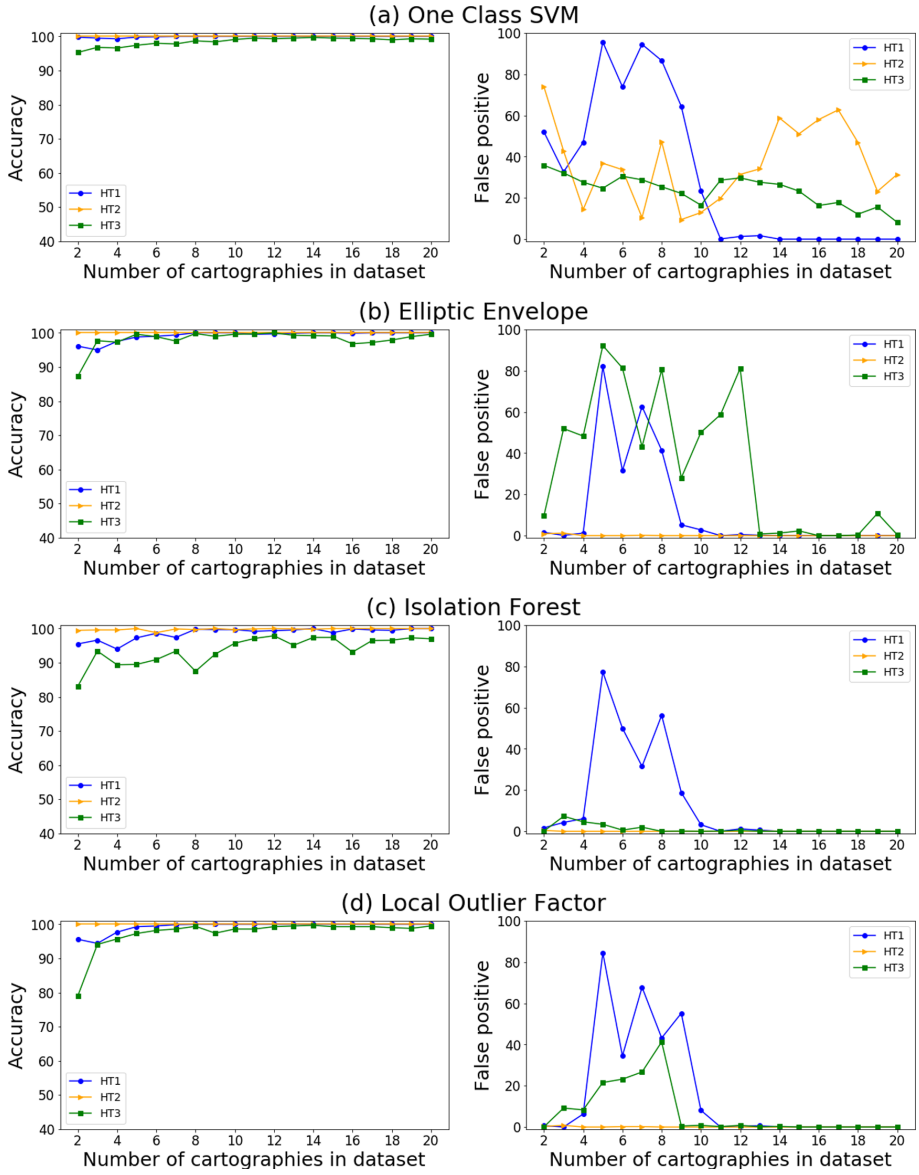

Fig. 6. HTs detection using outliers detection

a poor false positive rate (between 10% to 30% for HT2 and HT3) compared to other algorithms (nearly 1%). So the results show that we can detect effectively all these 3 HTs using the Elliptical Envelope and Local Outlier Detection algorithms. We can also notice that the new detection methods are much more efficient than T-test and supervised ML based detection methods. Figure 2 shows the performance comparison of our new method with those in the state of the art.

Table 2. Comparison of detection methods

	Method	Target	HT Size (%)	Detection rate
State-of-the-art	Raw trace comparison [21]	RISC-V	0.53, 0.27, 0.1	nc
	T-test [1]	RISC-V	0.53, 0.27, 0.1	70%
	One-Class SVM [15]	RISC-V	0.53, 0.27, 0.1	60%
This paper	Supervised ML methods	RISC-V	0.53, 0.27, 0.1	≈ 90%
	Test & outlier detection methods	RISC-V	0.53, 0.27, 0.1	≈ 100%

6 Discussion and Perspectives

As described in the introduction section, the HT is a big and serious challenge at the moment. Many methods and techniques are studied and proposed for the detection, but there is no universal method that can detect all HTs for the moment. Because of the complexity of the HT, the combination of different techniques may be required in order to increase the coverage of the detection.

In this paper, we study the HT detection based on the EM traces during the operation of the circuit or device. This can be useful to detect the HT during the testing phase. Application of ML is a new trend in the field of security in general and also in the hardware security. For the moment, the supervised ML algorithms based method is efficient to detect the HT. However, these algorithms require the dataset for genuine design and also for all HT designs. It could be interesting to evaluate the performance of the supervised ML based methods on new HTs which are not taken into account during the training phase. For the second method using the Outlier Detection algorithms such as Isolation Forest and Local Outlier detection, we obtain promising results comparing to those in the state of the art (Table 2). So a study of the performance of this method for HT detection could deserve more attention. For the future work, it could be interesting to test the performance of the outliers detection algorithms using the simulated traces as the reference for the training phase instead of the real traces coming from golden chip. We can apply this metric for a large HT database in order to have a complete evaluation of its performance. We need also to evaluate our methodology against the process variations.

This method could be very useful to classify and highlight the difference of two chip batches or to detect if a test dataset belongs to the same distribution as reference dataset or not. It can be applied to detect the HT that may have been inserted directly by the chip vendors. In this scenario, the chip vendors produce different chip versions for different clients because of the pressure coming from their government. If we can obtain the chips from different clients, we will detect the difference of these chips. In these scenarios, it raises a great suspicion about the genuineness of the delivered product, and the buyer may raise complains towards the provider and in turn require explanation about the dubious quality of the product. It can be also used as a forensic tool to detect similarities between two designs from different manufacturers, hence detecting if a manufacturer has reverse-engineered the design of another company and embedded in their circuit.

7 Conclusion

In this paper, we have proposed new HT detection methodologies using Machine Learning algorithms. Our methodology allows having an automatic method which is independent of the selected test samples. For the first method, we applied the supervised machine learning algorithms for the HT detection. The results show that we can obtain a detection rate of 90% with a false positive of 5% (with Support Vector Machine and K-nearest Neighbors) compared to the T-test (detection rate of 70% and false positive of 30%). For the second method, we applied the Outliers Detection algorithms combined with the T-test metric for the HT detection. This method can also detect different HT designs even with those which are never discovered (or never used for the training phase) unlike the statistic and supervised machine learning based methods. For this paper, we apply our methodology on the EM cartography traces and on a generic purpose processor RISC-V. Three HTs with different sizes of 0.53%, 0.27% and 0.1% are inserted for the experimentation. The results show that, using the Elliptical Envelope and Local Outlier Factor algorithms, we can detect the HT with a detection rate of 100% and a false positive smaller than 1%.

References

1. Balasch, J., Gierlichs, B., Verbauwhede, I.: Electromagnetic circuit fingerprints for hardware trojan detection. In: 2015 IEEE International Symposium on Electromagnetic Compatibility (EMC), pp. 246–251, August 2015
2. Banga, M., Hsiao, M.S.: A novel sustained vector technique for the detection of hardware trojans. In: International Conference on VLSI Design, pp. 327–332. IEEE (2009)
3. Banga, M., Hsiao, M.S.: ODETTE : a non-scan design-for-test methodology for trojan detection in ICs. In: International Workshop on Hardware-Oriented Security and Trust (HOST), pp. 18–23. IEEE (2011)
4. Bounsiar, A., Madden, M.G.: One-class support vector machines revisited. In: 2014 International Conference on Information Science Applications (ICISA), pp. 1–4, May 2014
5. Chiu, A.L.M., Fu, A.W.C.: Enhancements on local outlier detection. In: Seventh International Database Engineering and Applications Symposium, Proceedings 2003, pp. 298–307, July 2003
6. Courbon, F., Loubet-Moundi, P., Fournier, J.J., Tria, A.: A high efficiency hardware trojan detection technique based on fast SEM imaging. In: Nebel, W., Atienza, D. (eds.) Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE 2015, 9–13 March 2015, pp. 788–793. ACM, Grenoble (2015)
7. Ding, A.A., Chen, C., Eisenbarth, T.: Simpler, faster, and more robust T-test based leakage detection. In: Standaert, F.-X., Oswald, E. (eds.) COSADE 2016. LNCS, vol. 9689, pp. 163–183. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-43283-0_10
8. He, J., Zhao, Y., Guo, X., Jin, Y.: Hardware trojan detection through chip-free electromagnetic side-channel statistical analysis. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **25**(10), 2939–2948 (2017)

9. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intell. Syst. Appl. **13**(4), 18–28 (1998)
10. Jha, S., Jha, S.K.: Randomization based probabilistic approach to detect trojan circuits. In: Proceedings of the 2008 11th IEEE High Assurance Systems Engineering Symposium, HASE 2008, pp. 117–124. IEEE Computer Society (2008)
11. Jin, Y., Kupp, N., Makris, Y.: Experiences in hardware trojan design and implementation. In: Proceedings of the 2009 IEEE International Workshop on Hardware-Oriented Security and Trust, HOST 2009, pp. 50–57. IEEE Computer Society, Washington, DC (2009)
12. King, S.T., Tucek, J., Cozzie, A., Grier, C., Jiang, W., Zhou, Y.: Designing and implementing malicious hardware. In: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, LEET 2008, pp. 5:1–5:8. USENIX Association, Berkeley (2008)
13. Lecomte, M., Fournier, J., Maurine, P.: An on-chip technique to detect hardware trojans and assist counterfeit identification. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **25**(12), 3317–3330 (2017)
14. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 413–422, December 2008
15. Liu, Y., Jin, Y., Nosratinia, A., Makris, Y.: Silicon demonstration of hardware trojan design and detection in wireless cryptographic ICs. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **25**(4), 1506–1519 (2017)
16. Muehlberghuber, M., Gürkaynak, F.K., Korak, T., Dunst, P., Hutter, M.: Red team vs. blue team hardware trojan analysis: detection of a hardware trojan on an actual ASIC. In: 2nd International Workshop on Hardware and Architectural Support for Security and Privacy (HASP 2013), pp. 1:1–1:8. ACM, New York(2013). http://dx.doi.org/10.1145/2487726.2487727
17. Ngo, X.T., Danger, J.L., Guilley, S., Najm, Z., Emery, O.: Hardware property checker for run-time hardware trojan detection. In: 2015 European Conference on Circuit Theory and Design (ECCTD), pp. 1–4, August 2015
18. Rad, R., Plusquellic, J., Tehranipoor, M.: Sensitivity analysis to hardware trojans using power supply transient signals. In: Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, HST 2008, pp. 3–7. IEEE Computer Society, Washington, DC (2008)
19. SiFive: Source files for SiFive's Freedom platforms, 29 November 2016. https://github.com/sifive/freedom
20. Skorobogatov, S., Woods, C.: Breakthrough silicon scanning discovers backdoor in military chip. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 23–40. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33027-8_2
21. Söll, O., Korak, T., Muehlberghuber, M., Hutter, M.: EM-based detection of hardware trojans on FPGAs. In: 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 84–87, May 2014
22. Torrance, R., James, D.: The state-of-the-art in IC reverse engineering. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 363–381. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04138-9_26
23. Wolf, C.: PicoRV32 - A Size-Optimized RISC-V CPU. https://github.com/cliffordwolf/picorv32
24. Worley, K., Rahman, M.T.: Supervised machine learning techniques for trojan detection with ring oscillator network. In: 2019 SoutheastCon, pp. 1–7, April 2019