Hardware Trojan Detection based on Testability Measures in Gate Level Netlists using Machine Learning

Thejaswini P, Anu H, Aravind H S, D Mahesh Kumar, Syed Asif, Thirumalesh B, Pooja C A, Pavan G R
Department of Electronics and Communication Engineering,
JSS Academy of Technical Education, Bangalore-560060, Karnataka, India

***Abstract*—Modern integrated circuit design manufacturing involves outsourcing intellectual property to third-party vendors to cut down on overall cost. Since there is a partial surrender of control, these third-party vendors may introduce malicious circuit commonly known as Hardware Trojan into the system in such a way that it goes undetected by the end-users' default security measures. Therefore, to mitigate the threat of functionality change caused by the Trojan, a technique is proposed based on the testability measures in gate level netlists using Machine Learning. The proposed technique detects the presence of Trojan from the gate-level description of nodes using controllability and observability values. Various Machine Learning models are implemented to classify the nodes as Trojan infected and non-infected. The efficiency of linear discriminant analysis obtains an accuracy of 92.85 %, precision of 99.9 %, recall of 80%, and F1 score of 88.8% with a latency of around 0.9 ms.**

***Keywords*—Hardware trojan; machine learning; controllability; observability; detection and mitigation**

Abbreviations	
A	Accuracy
P	Precision
R	Recall
F	F1–Score
TP	True Positive
TN	True Negative

I. Introduction

Hardware Trojans are modern-day system attacks that will cause prominent damage to the IC or system in numerous ways. Though the software is considered to be vulnerable, the underlying hardware is generally considered to be safe. However, research has shown that, due to the complex nature of the design, fabrication process, rapid prototype development, and distribution of the final product, new sources of attack are prominent [1]. Speeding up the development cycle and lowering R&D costs is the main goal of most manufacturing companies because the estimated R&D cost is up to $5 billion. Most companies cannot afford to invest such a huge amount from start to finish. So, companies frequently outsource fabrication to a third-party foundry, buy IP cores from third-party suppliers, and employ EDA tools from third-party vendors. Third-party suppliers can readily enter such a model, and the supply chain is currently deemed vulnerable to assaults like Hardware Trojan insertion, reverse engineering, IP theft, IC tampering, IC cloning, and IC overproduction, among others. Hardware Trojans are arguably the most concerning of them, and they have attracted a lot of attention. It will eventually change the functionality of the system and the user will be unable to take any action against it [2].

Hardware Trojans can be defined as malicious components introduced during the design, manufacturing, fabrication, testing, or development phase [3]. Once introduced they can be activated anytime, anywhere, and according to the attacker's interest. The activation mechanism divides Hardware Trojan into two groups: always on and triggered. Always-on Trojans are active as soon as the systems or designs are turned on, whereas triggered Trojans require the activation of some form of condition. A Hardware Trojan circuit is generally designed in two parts; a condition-based circuit (trigger) and an operation circuit (payload) which is interconnected via trigger net [4]. The Trojan will be triggered and activated when the predefined criteria is satisfied. The most dangerous part is that they can be inserted anywhere in the circuit, be it processor, IC power grid, IO, and there is no way to immediately know the source of the threat. By the time it is discovered and neutralized it may be too late as it can change the functionality of the circuit. They can also downgrade its performance, leak sensitive information and finally cause a Denial-of-Service attack [3]. Therefore, to find a solution for many such attacks, various researches are being conducted. Amongst them, the logic function test (LFT) is a traditional method [5]. Most of the existing methods of LFT do not effectively activate any potential hidden Trojans. Researchers use side-channel analysis to detect Hardware Trojans by modelling and analysing electromagnetic information generated during chip operation [6]. Traditional side-channel analyses' effect isn't sufficient, according to researchers [7], due to low sensitivity detection rates for big process fluctuations and a small Trojan footprint. Combining principal component analysis and linear discriminant analysis to analyse chip power is effective in evaluating Trojan detection accuracy [8]. Machine learning techniques for malware detection are the most successful state-of-the-art research topic because of their ability to keep up with malware evolution. They concentrate mainly on two areas, one is feature extraction, and the other is dimensionality reduction. Support Vector Machine algorithm seems to be the go-to algorithm for detection in multiple cases [9]. However other advanced algorithms are also being explored to trigger and detect a Trojan. Triggering a Trojan has an impact on the system's

power usage as well as the circuits' delay. This behavior is extensively used to study the power and delay characteristics of hardware Trojans in order to detect them[10].

Even though there is a possibility to detect the Trojan using the different techniques as stated so far, there is still scope for optimization. The contributions made by this paper are as follows:

1) We propose the use of gate-level descriptions of nodes using controllability and observability values to generate a dataset to detect the presence of Hardware Trojan.

2) Machine Learning techniques are used to detect the presence of Trojan.

3) Benchmark circuits-C17, C3540 and C432 are considered to validate our proposed technique.

4) We experimentally prove that our proposed model has an area and power reduction up to 75% and 80% respectively in Trojan free circuits. This performs better in comparison to other state-of-art techniques.

The rest of this paper is organized as follows. Section II and III analyze the related works and motivation. Section IV shows the proposed scheme. The experimental results and discussion are presented in Section V. Section VI shows the conclusion.

II. Related Works

Various techniques ranging from score-based classification [6] for identifying Hardware-Trojan-free or Hardware-Trojan-infected circuits without using golden model-based approach to deep learning techniques are researched. The side-channel analysis and detection method [11], uses dimensional reduction to detect HTs. This causes the loss of important feature information of Hardware Trojans after the principal component analysis or filtering process. To solve this problem, a Hardware Trojan detection technology is proposed based on Extreme Learning Machine (ELM), which can completely retain important information without any inaccuracies caused by modelling. Results show that detecting the Hardware Trojans only used about 0.15% of resources. The accuracy rate was about 90%.

In terms of router looping, traffic diversion, or core spoofing, a trojan attack corrupts the router packet [9] by changing the destination address. As a solution, SVM is used to increase detection accuracy. According to the estimates, the suggested security solution architecture achieves a 93 percent accuracy for seizure detection applications in 4.8μS.

Detection of Trojan using gate-level netlist based on observability and controllability analysis [4] produce sufficient results. When this technique was used on numerous trojans, the findings reveal that even in the worst situation, all Trojans are discovered effectively with zero false positive and negative rates in less than 14 seconds.

Using a specific gate-level netlist that specifies the Trojan nets in full, the review paper [12] covers extracting 51 gate-level Trojan features. The usage of an ensemble- based random forest classifier results in a true positive and true negative rate of 100 percent.

Implementation of SVM using five-dimensional vectors [13] classifies all the nets in an unknown netlist into Trojan affected and normal ones using the Trust-HUB benchmark. Not only SVM but various other machine learning algorithms like Decision Tree, K Nearest Neighbour [14] is applied to identify the Trojan. Further to increase classification accuracies, Hardware Trojans are discretized based on their dominant attributes. The results show that both Machine Learning algorithms when trained on a given dataset perform well. DT and KNN models can accurately predict about 83% of the test data.

In addition, existing deep neural networks security studies are not extensively conducted at these software algorithm levels [15] and the more realistic attacks by third-party vendors are not explored. So, it is successful in demonstrating how an attack is possible. Experiments reveal that the approach can quickly generate and activate a variety of Trojan attacks that can readily overcome existing defenses. This is very important in formulating a solution to the attack.

III. Motivation

The presence of Hardware Trojans (HTs) in circuits causes malfunction on various scales depending on the type of Trojan attack. Amongst the numerous existing state-of- art techniques, the FANCI [16] technique uses a coverage-like approach where it does not require access to any verification stimuli. But the attackers like third-party vendors being aware of this can make the Trojan look benign. The third-party inputs from on-chip IPs can be scrambled to suppress the Trojan triggers. But this would not work for analog triggers. So, the Side-channel techniques can be used to unmask Trojans injected by third parties [17], but this method is unusable until IC is manufactured and inside the supply chain. As a solution, the use of formal proofs enables to development of trusted IPs. However, this assumes that proof is sufficient to rule out injected Trojans and that IP specifications are known. Therefore, there is a need for better-advanced techniques to counter novel malware attacks. One such technique is Machine Learning (ML) [18].

The rise of Machine Learning has profound implications for many industries, including cyber security. ML-based anti-malware tools are generally believed to provide better detection of modern malware attacks and improve scanning methods. Machine Learning algorithms perform better against unforeseen threats as they can be trained to handle unknown potential threats. This is a major advantage over other techniques. They yield more accurate, efficient solutions. As a result, employing ML techniques proves to be beneficial.

IV. Methodology

The proposed Trojan detection technique is a combinational circuit [19] for avoiding unintended malicious activity is as shown in Fig. 1. Controllability and Observability are the two parameters considered for the detection and classification of Trojan in the given circuit [20].

To implement the proposed methodology, let us consider ISCAS benchmark circuits. These are implemented using Verilog. We have designed a Trojan threat model and applied it to ISCAS benchmark circuits. Then we generate a Gate Level

netlist. Using Python and gate-level netlist, we obtain values of two parameters, Controllability and Observability to detect the presence of Trojan in the circuit. Hence, we create a dataset using these values as input. The model is trained and tested by applying different Machine Learning techniques. The desired output is obtained and verified using functional verification and overhead analysis.

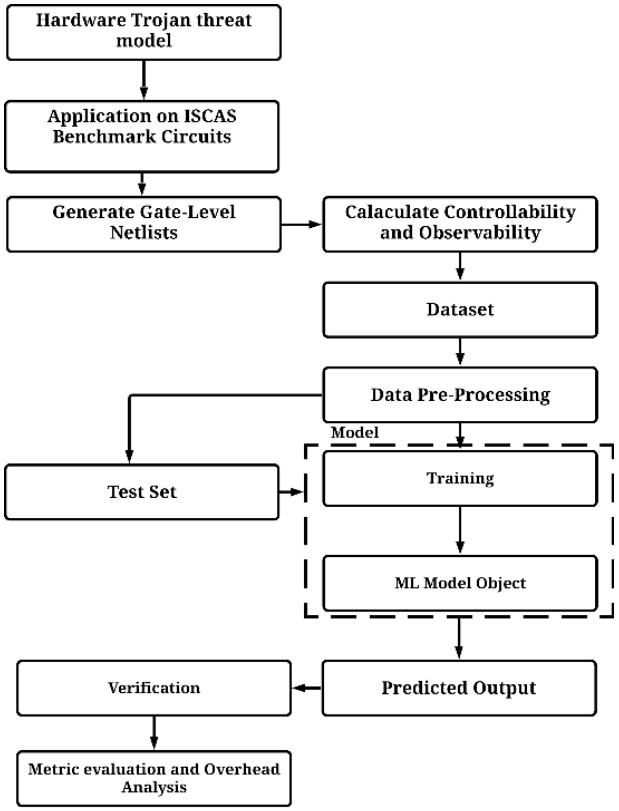

Fig. 1. Block diagram of proposed methodology

A. Design of Trojan Threat Model

A trojan is a unique circuit that performs specific malicious activity [21]. Trojan considered in our proposed work will change the functionality of the circuit. Fig. 2 shows the Trojan threat model designed for our work. Here, 4-bit Linear Feedback Shift Register (LFSR) is used to trigger the Trojan and NOT gate act as payload. One input to the comparator is from LFSR and the other input is a random number generated by the attacker. The output of the comparator is connected to the select line of MUX. If the output of the comparator is one, Trojan is triggered and vice versa.

B. Benchmark Circuit Selection

The proposed method is implemented on a standard ISCAS benchmark circuit [22]. The ISCAS '85 benchmark circuits are combinational circuits that are used by researchers as the basis for performing analysis and comparing results. C17-NAND only circuit, C432-a 27 channel interrupt, and C3540–an 8-bit ALU are the three circuits considered in our work as depicted in Fig. 3, Fig. 4 and Fig. 5.

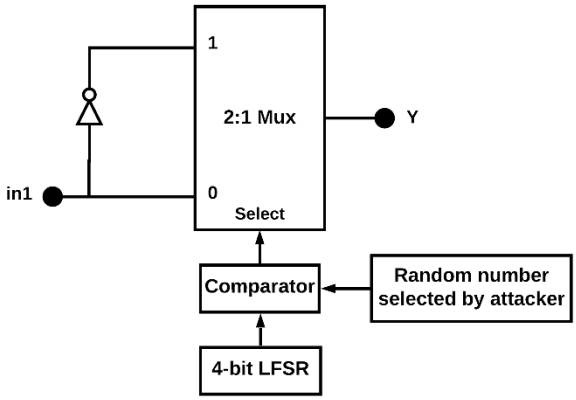

Fig. 2. Design of trojan threat model

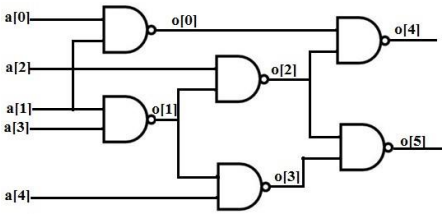

Fig. 3. C17 (NAND circuit)

C. Gate Level Netlist

Gate Level Netlist contains information regarding the logical connectivity of all standard cells and macros [23]. Gate level netlist of C17, C432, and C3540 is generated by synthesizing these circuits using the Cadence tool at 90nm technology. These gate-level netlists are used for calculating observability and controllability values.

D. Controllability and Observability Calculation

The presence of Trojans in the circuits is analyzed by calculating the controllability and observability values.

1) Controllability analysis: Nets with poor testability are identified using combinational controllability. The levels of controllability range from 1 to infinity. Because the possibility of detecting such a node is very low, and controlling that particular node is quite difficult, nodes with high controllability (CC) ratings are more susceptible to having a trojan inserted. All signals from primary inputs to primary outputs have their controllability values determined first. The circuit is initially levelled by giving each gate a level value [24]. Each gate's output controllability is then calculated. Controllability (CC) in general is expressed as:

$$CC(i) = \sqrt{CC0(i)^2 + CC1(i)^2} \quad (1)$$

where, CC0(i) is Combinational Controllability 0 and CC1(i) is Combinational Controllability 1

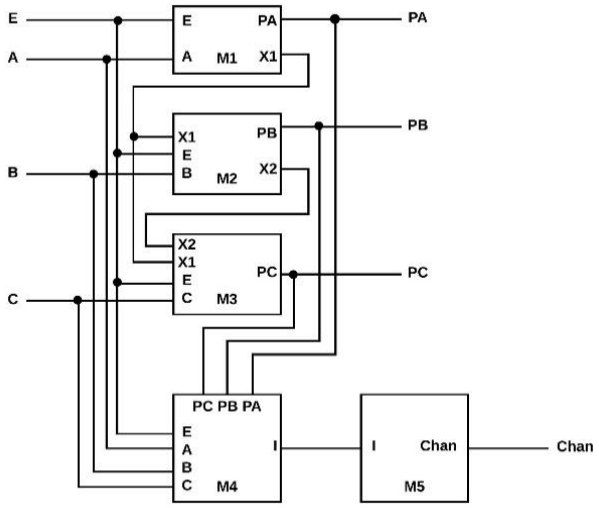

Fig. 4. C432 (27 channel interrupt controller circuit)

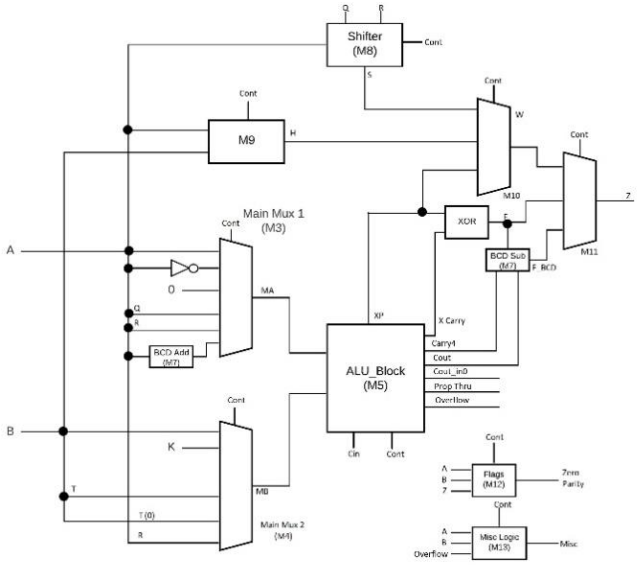

Fig. 5. C3540 (8-Bit ALU circuit)

2) Observability analysis: Observability is a measure of the ease (or difficulty) with which one can calculate the signal value at any logic node in the circuit by controlling its primary input and observing the primary output [4]. The observability values for all signals from primary outputs towards primary inputs are then calculated. The observability of one input of a AND gate with multiple inputs is given by

$$CO(s) = P(x,y)+1 \quad (2)$$

where, x=output observability and y= CC1 of other inputs. If 'U' is a primary output node of a digital circuit, then the combinational observabilities of node I are defined as, CO(U) = 0. Table II, Table III, and Table IV reports the controllability and observability values obtained.

E. Machine Learning Algorithms and Data SetCreation

The use of machine learning algorithms aids in the better analysis of various Trojan threats because they are capable of processing massive amounts of data with greater precision. Using Python, dataset is created by evaluating the controllability and observability values received from the chosen benchmark circuits. The more advanced algorithms can be trained to detect any kind of Trojan across various platforms. Hence, we employ various Machine Learning Techniques (MLT) [25] along with our proposed technique to detect the presence of Trojan in the circuits by classifying the nodes as Trojan free and Trojan infected.

V. Experimental Results and Discussion

A. Functionality Verification

To understand the impact of Trojans on the circuits let us consider C3540, an 8bit ALU. C3540 consists of Mux, shift register, ALU_Core, xor gate, and BCD subtractor. Trojan can be introduced to any block in the circuit. To understand the effect of Trojan, we are introducing Trojan to the BCD adder block in C3540. Let us consider two and five as inputs to this BCD block. When this block is configured as adder, the output is six in the absence of Trojan. When Trojan is activated, the output changes. This can be observed in Fig. 11, where the Trojan activation at 72ns changes the output value from 6D to 83 and remains in the same state as long as Trojan is activated. Once the Trojan is deactivated, output changes to the original value. As the blocks in C3540 are cascaded as seen in Fig. 11, the final output of ALU will also be changed. This clearly demonstrates that, due to the presence of Trojans, functionality of the circuit will change. Similarly, functionality verification of C17 and C432 benchmark circuits are carried out for both the cases, with Trojan and without Trojan. The obtained simulation results are shown in Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10, and Fig. 11. Thus, this shows how the entire functionality of a circuit or a system change upon Trojan activation.

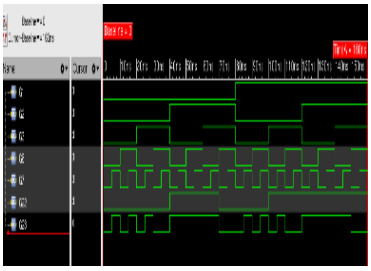

Fig. 6. C17 simulation result without Trojan

Fig. 7. C17 simulation result with Trojan

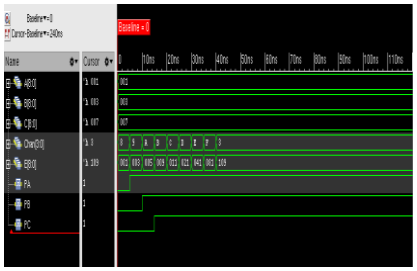

Fig. 8. C432 simulation result without Trojan

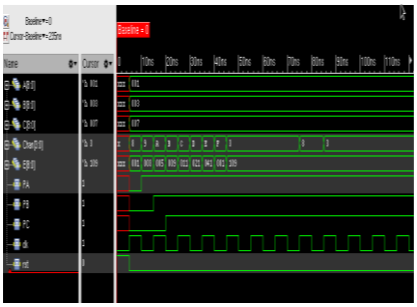

Fig. 9. C432 simulation result with Trojan

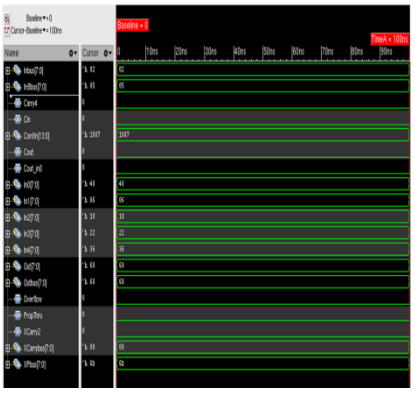

Fig. 10. C3540 simulation result without Trojan

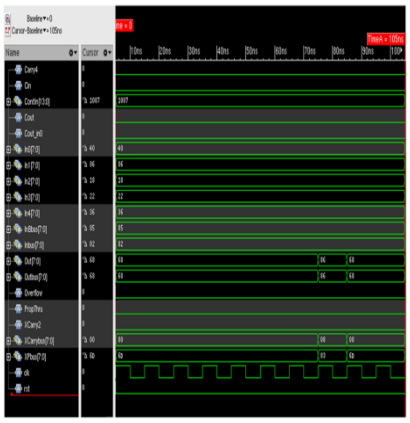

Fig. 11. C3540 simulation result with Trojan

B. Performance Analysis

Performance analysis of the benchmark circuits-C17, 432 and C3540 using area and power metrics are shown in Table I. The circuits are implemented in Verilog HDL using Cadence Incisive and synthesized in 90nm technology using Cadence Genus. It can be observed that, area of the circuits without Trojan is less compared to with trojan. It can be noticed that, power consumption of the circuits in the absence of Trojan is less. From Table I, it is evident that Trojan presence can also be determined by performing area and power analysis. This is due to the fact that, any new addition of unwanted components such as Trojan, increases the area and power consumption in a drastic way.

TABLE I. SYNTHESIS REPORT

Bench mark Circuits	Area(μm^2)		Power(μW)	
	Without Trojan	**With Trojan**	**Without Trojan**	**With Trojan**
C17	**16.62**	**227.07**	**0.425**	**6.76**
C432	**439.002**	**3279.648**	**15.205**	**107.09**
C3540	**4931.203**	**5196.118**	**185.38**	**194.27**

C. Controllability and Observability Calculation

Controllability and Observability are the parameters used for detection of Trojan in the benchmark circuits. Netlist obtained from synthesis of C17, C432 & C3540 is converted into benchmark formats. The benchmark formats are then fed as input to the python code that determines controllability and observability values. The output in the form of a text file is used to create datasets. Combinational controllability is used to identify nets which show difficulty in testability. The controllability values range from 1 to infinity. If 'I' is a primary input node of a digital circuit, then the combinational controllabilities of node 'I' are defined 1 i.e., CC0(I) and CC1(I)=1. Similarly, calculations of combinational controllabilities for various gates are shown in Fig. 12.

An Observability is simply a function of controllability, meaning that it is impossible to observe a given internal node if the circuit is not driven to a given state. The Observability values range from 0 to infinity. If 'U' is a primary output node of a digital circuit, then the combinational observabilities of node 'U' are defined as, CO(U) = 0

The formulation of combinational observabilities for various gates are shown in Fig. 13 where,

CC = Combinational Controllability

CO = Combinational Observability

Trojan value 0 = No Trojan detected

Trojan value 1 = Trojan detected

Table II, Table III, and Table IV represent samples of datasets generated using Controllability and Observability values obtained using the calculations for various gates mentioned in Fig. 12 and Fig. 13. The gate-level netlists are converted to benchmark codes that are fed into the python code that performs the calculations to output files with the testability measures.

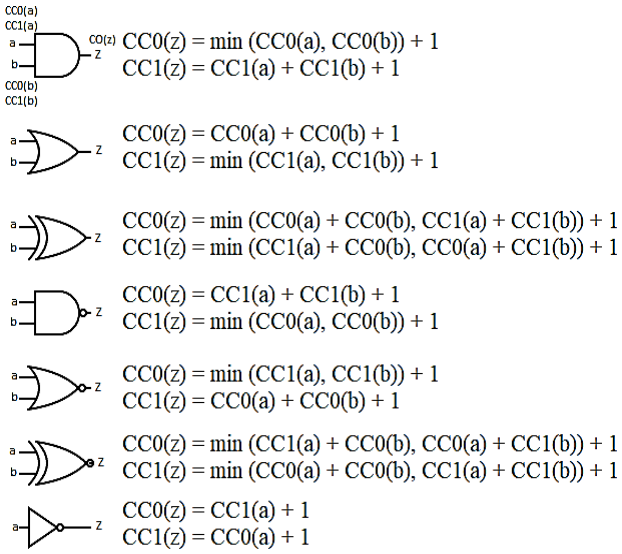

Fig. 12. Combinational controllability calculation for various gates

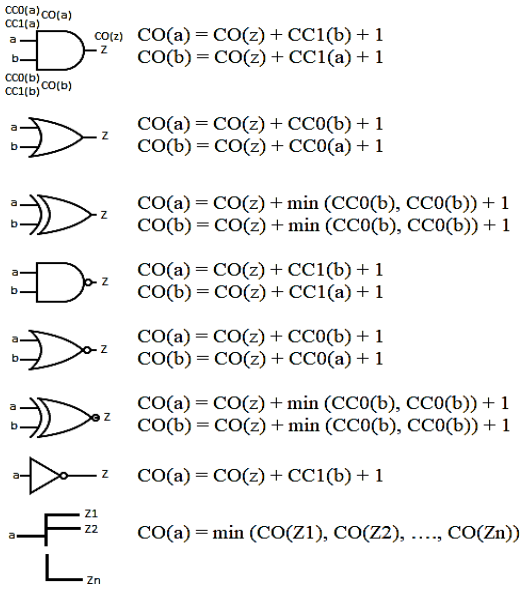

Fig. 13. Combinational observability calculation for various gates

TABLE II. C17 DATASET

Line Name	CC	CO	Trojan
G3gat	4.2	17	0
G1gat	1.4	5	0
G22gat	6.4	0	0
G10gat	3.6	3	0
G19gat	4.5	3	0
G3gat	4.2	54	1
G10gat	38.1	3	1
G19gat	4.5	3	1
G1gat	1.4	42	1
G22gat	7.1	0	1

TABLE III. C432 DATASET

Line Name	CC	CO	Trojan
G159	13.5	74	0
G165	13.5	74	0
G295gat	14.3	152	0
G1gat_1	1.4	38	0
. . .			
G236gat_0	14.2	157	0
G159	13.5	76	1
G165	13.5	76	1
G360gat	15.44	244	1
. . .			
G1gat_1	1.4	56	1
G236gat_0	14.2	163	1

TABLE IV. C3540 DATASET

Line Name	CC	CO	Trojan
G905	22.8	141	
G906	7.6	89	0
G116	5.6	409	0
. . .			
G353	11.7	0	0
G68	8.4	364	0
G905	126.3	141	1
G906	42.1	89	1
G116	5.6	384	1
. . .			
G1018	41.2	82	1
G625	42.1	4	1

D. Machine Learning Techniques (MLT)

MLT are used for detection of Trojans in the benchmark circuits. Controllability and Observability values calculated for C17, C432 and C3540 are used as input to create dataset. Upon pre-processing the data, dataset is split into training set and test set. This data set is used on various MLT for classifying the nodes as Trojan free and Trojan infected. Scatter plot shown in Fig. 14, Fig. 15 and Fig. 16 better visualizes the results. C17 circuit is easier to classify as with Trojan and without Trojan but C432 circuit has considerable overlapping and is harder to classify. Thus, non-linear ML models have to be used in order

to separate this nonlinear data. Hence, we propose an MLT model which is a combination of LDA and Naive Bayes. Comparative study of all these MLT along with the proposed model in terms of accuracy, precision, F1 score and Recall is carried out for C17, C432 and C3540 circuits (see Fig. 17 to 22). It is evident from Table V, Table VI and Table VII that our proposed LDA+ Naive Bayes model is best among other MLT with the latency of around 0.9 ms in comparison with the other state-of-art techniques.

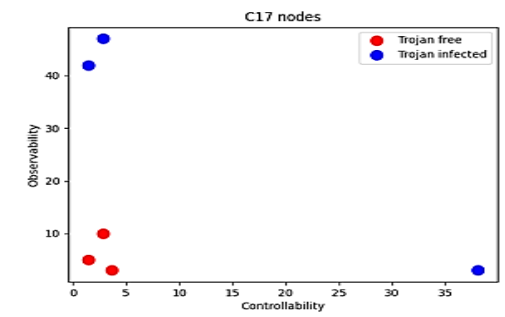

Fig. 14. Observability (Y) vs controllability (X) scatter plot results of C17 nodes

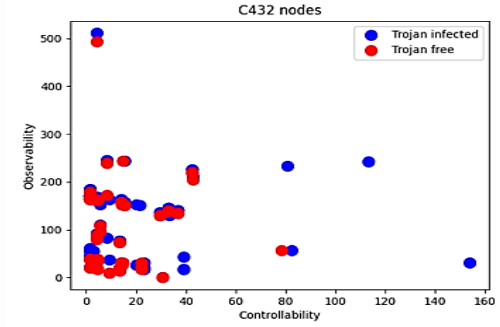

Fig. 15. Observability (Y) vs controllability (X) scatter plot results of C432 nodes

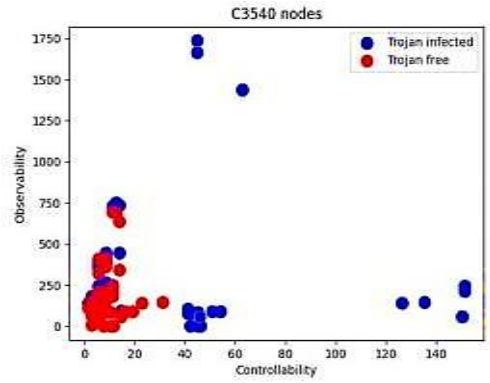

Fig. 16. Observability (Y) vs controllability (X) scatter plot results of C3540 nodes

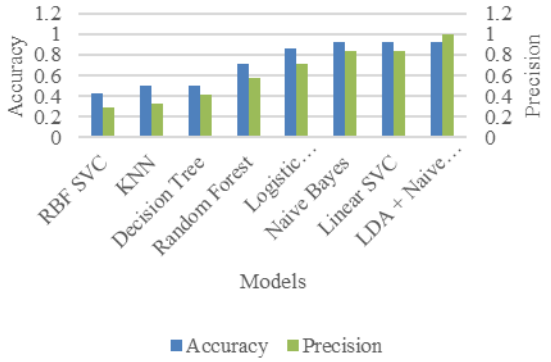

Fig. 17. C17 accuracy and precision metrics

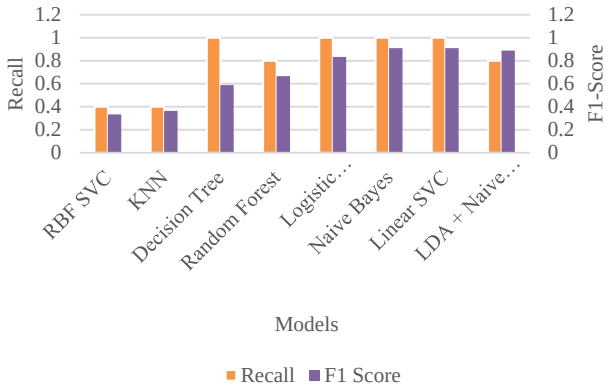

Fig. 18. C17 recall and F1-score metrics

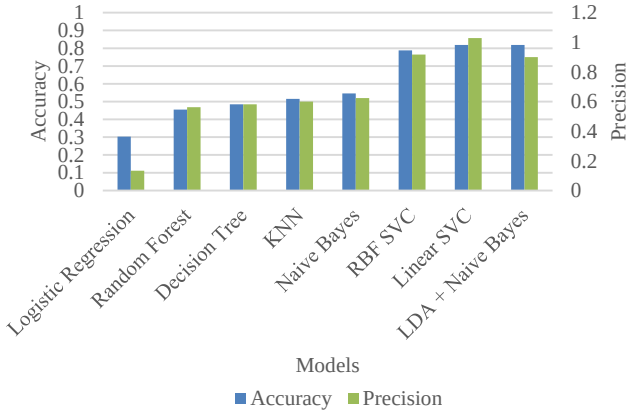

Fig. 19. C432 accuracy and precision metrics

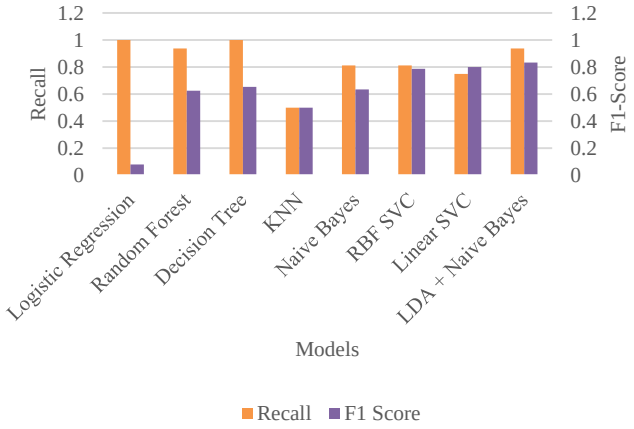

Fig. 20. C432 recall and F1-score metrics

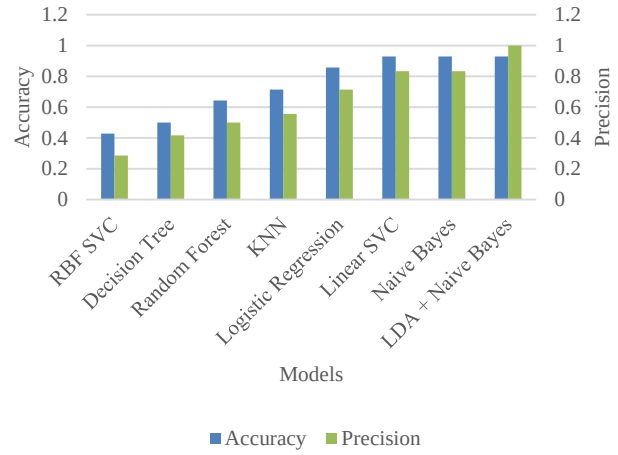

Fig. 21. C3540 accuracy and precision metrics

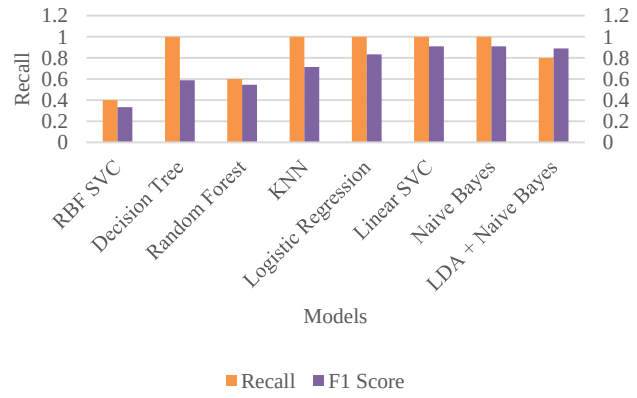

Fig. 22. C3540 recall and F1-score metrics

The Performance Metrics in Table V, Table VI, Table VII use the following parameters:

TABLE V. C17 PERFORMANCE METRICS

Model	A (%)	P (%)	R (%)	F (%)	TP	TN	Time (ms)
RBF SVC	43	29	40	33	4	2	0.9
KNN	50	33	40	36	5	2	3.9
Decision Tree	50	42	100	59	2	5	0.9
RandomForest	71	57	80	67	6	4	17.9
Logistic Regression	86	71	100	83	7	5	0.9
Naive Bayes	93	83	100	91	8	5	0.9
Linear SVC	93	83	100	91	8	5	0
LDA + Naive Bayes	93	100	80	89	9	4	0.9

TABLE VI. C432 PERFORMANCE METRICS

Model	A (%)	P (%)	R (%)	F (%)	TP	TN	Time (ms)
Logistic Regression	30	11	100	8	9	1	0.0
Random Forest	45	47	94	63	0	15	0.9
Decision Tree	48	48	100	65	0	16	0.0
KNN	52	50	50	50	9	8	1.9
Naïve Bayes	55	52	81	63	5	13	0.0
RBF SVC	79	76	81	79	13	13	11.9
Linear SVC	82	86	75	80	15	12	0.5
LDA + Naïve Bayes	82	75	94	83	12	15	1.9

TABLE VII. C3540 PERFORMANCE METRICS

Model	A (%)	P (%	R (%)	F (%)	TP	TN	Time (ms)
RBF SVC	43	29	40	33	4	2	0
Decision Tree	50	42	100	59	2	5	0
Random Forest	64	50	60	55	6	3	9.9
KNN	71	56	100	71	5	5	2.9
Logistic Regression	86	71	100	83	7	5	0.9
Linear SVC	93	83	100	91	8	5	0
Naïve Bayes	93	83	100	91	8	5	0.9
LDA + Naïve Bayes	93	100	80	89	9	4	0.9

VII. CONCLUSION

Considering the various threats to the manufacturing process of an IC/system like functionality change, IC tampering, third party vendor attacks etc, a new Hardware Trojan detection technique using Machine Learning is proposed. Controllability and Observability analysis was performed using Gate Level netlists. Based on the values obtained, the various Machine learning models were able to distinguish the nodes in three of the benchmark circuits used as Trojan free or Trojan infected. Amongst them, our proposed model i.e., LDA + Naïve Bayes performed the best when compared to other state-of-art techniques with an accuracy of 92.85%, precision of 99.9

%, recall of 80% and F1 score 88.8%. The latency of the proposed technique was around 0.9ms. Along with this, the Simulation and Synthesis reports obtained using 90nm technology of Cadence tool also proved that presence of Trojan increasingly affects the system in terms of area and power.

REFERENCES

[1] K. Madden, J. Harkin, L. McDaid, and C. Nugent, "Adding Security to Networks-on-Chip using Neural Networks," 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 2018, pp. 1299-1306.

[2] Noor, N. Q. et al. “Hardware Trojan Identification Using Machine Learning-based Classification.” Journal of Telecommunication, Electronic and Computer Engineering 9 (2017): 23- 27.

[3] Z. Huang, Q. Wang, Y. Chen and X. Jiang, "A Survey on Machine Learning Against Hardware Trojan Attacks: Recent Advances and Challenges," in IEEEAccess, vol. 8, pp. 10796-10826, 2020.

[4] H. Salmani, "COTD: Reference-Free Hardware Trojan Detection and Recovery Based on Controllability and Observability in Gate-Level Netlist," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 2, pp. 338-350, Feb. 2017.

[5] J. Cruz, F. Farahmandi, A. Ahmed, and P. Mishra, "Hardware Trojan detection using ATPG and model checking," in 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), 2018: IEEE, pp. 91-96.

[6] Y. Xiang, L. Li and W. Zhou, "Random Forest Classifier For Hardware Trojan Detection," 2019 12th International Symposium on Computational Intelligence and Design (ISCID), 2019, pp. 134-137, doi: 10.1109/ISCID.2019.00037.

[7] Y. Huang, S. Bhunia, and P. Mishra, "MERS: statistical test generation for side-channel analysis based Trojan detection," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016: ACM, pp.

[8] J. He, Y. Zhao, X. Guo, and Y. Jin, "Hardware trojan detection through chip-free electromagnetic side- channel statistical analysis," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25,no. 10, pp. 2939-2948, 2017.

[9] A. Kulkarni, Y. Pino and T. Mohsenin, "SVM-based real-time hardware Trojan detection for many-core platform," 2016 17th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, 2016, pp. 362-367.

[10] A. Amelian and S. E. Borujeni, "A side-channel analysis for hardware Trojan detection based on path delay measurement," Journal of Circuits, Systems and Computers, vol. 27, no. 09, p. 1850138, 2018.

[11] F. K. Lodhi, I. Abbasi, F. Khalid, O. Hasan, F. Awwad and S. R. Hasan, "A self-learning framework to detect the intruded integrated circuits," 2016 IEEE International Symposium on Circuits and Systems (ISCAS), 2016, pp. 1702-1705, doi:10.1109/ISCAS.2016.7538895.

[12] K. Hasegawa, M. Yanagisawa and N. Togawa, "Trojan-feature extraction at gate-level netlists and its application to hardware-Trojan detection using random forest classifier," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, 2017, pp. 1-4.

[13] K. Hasegawa, M. Oya, M. Yanagisawa and N. Togawa, "Hardware Trojans classification for gate- level netlists based on machine learning," 2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS), Sant Feliu de Guixols, 2016, pp. 203-206.

[14] Noor, N. Q. et al. “Hardware Trojan Identification Using Machine Learning- based Classification.” Journal of Telecommunication, Electronic and Computer Engineering 9 (2017): 23- 27.

[15] T. Liu, W. Wen and Y. Jin, "SIN2: Stealth infection on neural network — A low-cost agile neural Trojan attack methodology," 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), Washington, DC, 2018, pp. 227-230.

[16] A. Waksman, M. Suozzo, and S. Sethumadhavan, “FANCI: identification of stealthy malicious logic using boolean functional analysis,” in Proc. ACM SIGSAC Conference on Computer and Communications Security (ACM-CCS), 2013, pp. 697–708.

[17] A. Moradi, D. Oswald, C. Paar, and P. Swierczynski, “Side-channel attacks on the bit stream encryption mechanism of Altera Stratix II,” Proc. ACM/SIGDA Int. Symp. F. Program. Gate arrays - FPGA ’13, p. 91, 2013.

[18] R. Elnaggar and K. Chakrabarty, “Machine Learning for Hardware Security: Opportunities and Risks,” Journal of Electronic Testing, vol. 34, no. 2, pp. 183- 201, Apr. 2018, doi: http://doi.org/10.1007/s10836-018-5726-9.

[19] M. Banga, M.S. Hsiao, “A Region based Approach for the Identification of Hardware Trojan,” in Proc. IEEE Intl. Symp. On Hardware Oriented Security and Trust (HOST’08), pp. 40-47, Jun. 2008, doi: 10.1109/HST.2008.4559047.

[20] Goldstein, Lawrence H., and Evelyn L. Thigpen. "SCOAP: Sandia controllability/observability analysis program." Proceedings of the 17th Design Automation Conference. 1980.

[21] J. Wang, S. Guo, Z. Chen and T. Zhang, "A Benchmark Suite of Hardware Trojans for On-Chip Networks," in IEEE Access, vol. 7, pp. 102002- 102009, 2019.

[22] Trust-Hub.org

[23] Martin, Antonio J. Lopez. "Cadence design environment." New Mexico State University, Tutorial paper (2002): 35.

[24] Reshma K., Priyatharishini M., Nirmala Devi M. (2019) "Hardware Trojan Detection Using Deep Learning Technique". In: Wang J., Reddy G., Prasad V., Reddy V. (eds) Soft Computing and Signal Processing. Advances in Intelligent Systems and Computing, vol 898. Springer, Singapore.

[25] K. G. Liakos, G. K. Georgakilas, S. Moustakidis, P. Karlsson and F. C. Plessas, "Machine Learning for Hardware Trojan Detection: A Review," 2019 Panhellenic Conference on Electronics & Telecommunications (PACET), 2019, pp. 1-6, doi: 10.1109/PACET48583.2019.8956251.