Received: July 21, 2020. Revised: August 26, 2020. 318

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

An Optimal Feature Extraction using Deep Learning Technique for Trojan Detection and Validation using Game Theory

Priyatharishini Murugesan[1]* Nirmala Devi Manickam[1]

[1]*Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India*
* Corresponding author's Email: m_priyatharishini@cb.amrita.edu

Abstract: In modern electronic design, hardware Trojan has emerged as a major threat in the hardware security. To detect the hardware Trojan is a major problem in testing process because of their inherent concealed nature. In this work, we propose a deep learning-based Trojan classification approach, which extracts the optimal feature to indicate the nets affected by the Trojan module. In this approach, a handcrafted algorithm along with the structural report is also analyzed for extracting further features of the gate level netlist, which stamp out the requirement of golden chip. This detection technique is also validated using game theoretical approach, which is modelled as zero-sum game between the attacker and the defender. The Simulation is employed on ISCAS'85, ISCAS'89 and Trust-HUB circuits and the deep learning algorithm performs the best in detection and classification of Trojan type with an average True positive rate of 96.69% and an accuracy of 96.25%.

Keywords: Hardware trojan, Hardware security, Deep learning, Feature, Game model.

1. Introduction

Hardware designers need to depend on third party manufactures due to the technological growth of the Integrated circuit(IC). As a result, it makes easy for the attacker to intrude the design at various points. Such a violation in the security of the IC design may result in malicious modifications referred to as Hardware Trojans (HT) [1, 2]. In order to evade from the formal design testing and verification phase, the hardware Trojans are modelled in such a manner that they are stealthy in nature and triggered by rare events.

Detection of hardware Trojans and prevention of Trojan insertion by the adversary is a challenging task and addressed in the recent literatures. Various detection schemes are broadly classified into side channel analysis, online checker, functional test, runtime monitoring. In [3] a vulnerability analysis of the digital circuit is performed at the behavioural level circuit. The dummy scan flip flops are intruded in the design to improve the triggering frequency of the transition probability. The region-based approach is one of the classifications of side channel analysis [4, 5], in which the circuits under test are partitioned into blocks for detecting the Trojan. In side channel analysis, the frequency and power profile are considered, in which the fluctuation in these parameters for Trojan nets will be very low and is undetected during testing phase. The solution for the above drawback advances the scheme of online checker and here the Trojan prevention schemes obfuscate the original design from the attackers by inserting extra module to the design [6]. This process increases the trustworthiness, but the design and resource overhead are considered high.

Therefore, in order to address both the drawbacks, run time detection techniques is emerged with the concept of reversible logic incorporated in the design [7]. An optimal test pattern generation technique is presented in [8], which selects the sparse test set for detecting the HT. Hence to solve the drawback of above, the deep learning approach is emerged for extracting the optimal set of internal net features in a suitable manner to trigger the Trojan module and this approach is independent of test patterns.

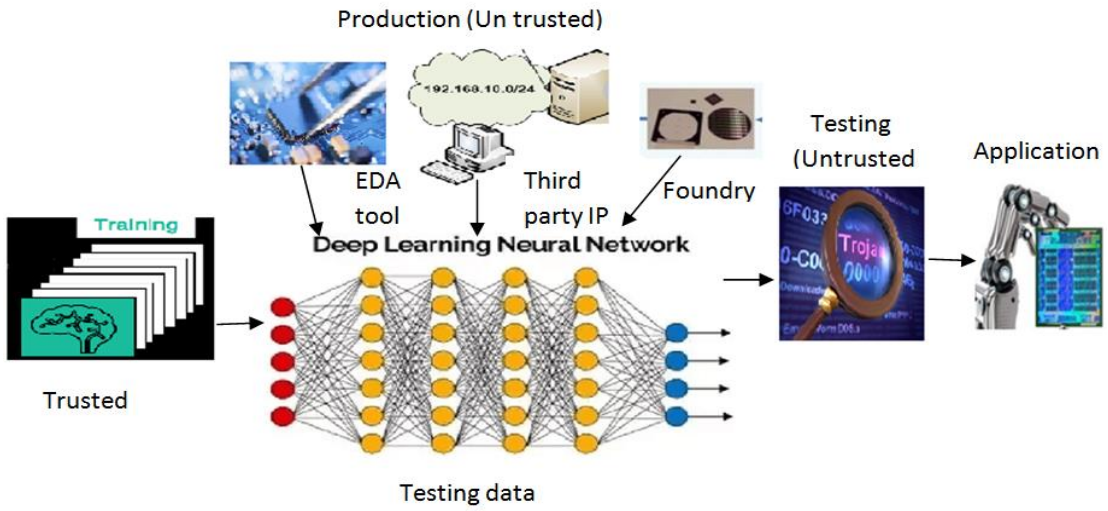


Figure. 1 Application scenario of deep learning technique

Deep learning algorithm is proven to be the best method of feature extraction for various real time applications in image processing, traffic identification and bio imaging. The Hardware Trojans can be embedded from register transfer level to designing gate level netlist, to packing the integrated chips are shown in the Fig. 1. In the design phase, the EDA tool is used by the IP suppliers and chip designers but the reliability of that tool is un-trusted. In the production phase, the un-trusted employee's involvement in design layout needs more attention. The threat models are inserted to change the functionality of the circuit by inserting, deleting or modifying the Trojan components to extract and control the original chip.

This paper presents an efficient feature extraction technique for detecting the hardware Trojan in the gate level net-list. This technique is referred to deep learning based hardware Trojan detection. Further it is validated using game theory. The proposed method comprises three different phases: Optimal Trojan feature extraction phase which extracts and encodes the Trojan features. Detection phase consists of learning phase to train the model and classification phase to precisely cluster the data, Validation phase to investigate the proposed detection scheme. The proposed methodology is evaluated on the benchmark circuits considering the performance metrics resembling True positive rate, True negative rate, Processing time, Probability of Defender, Expected pay off of defender, Attackers probability and Accuracy.

The main contributions of this work are as follows:

1) The proposed algorithm is attempted to extract the efficient features of Trojan nets from the large pool of nets in the circuit. This scheme extracts compact list of features, resulting minimal processing time for computation.
2) An algorithm is developed to suit any type of circuit by extracting the handcrafted features. A specific threat model is considered and the features are also extracted from synthesis tool.
3) An automated algorithm is developed along with K-means clustering to have a great impact on deciding the normal nets and the Trojan nets. On applying the proposed scheme to Trust- HUB circuits a high true positive rate is achieved and the results are compared.
4) A net scoring algorithm is developed to identify the efficient internal nodes in the net list and to generate the pay off matrix for validation.
5) The Trojan detection process is validated using game theory approach with a defender and adversary as a two-person strategic game model, to achieve high reliability.

The rest of the paper is organized as follows. The related work on different schemes for detecting hardware Trojan is described in section 2. In Section 3, the proposed deep learning approach for detecting the Trojan is presented. Section 4 describes the results and analysis for various ISCAS benchmark and Trust-HUB circuit. Finally, Section 5 concludes with some future scope on the related topic discussed in this paper.

2. Related work

The Trojan circuits are analyzed in various levels of the design, which include significant node selection and extraction of specific features of the

nets. Based on the outcome of these parameters, classification of the circuit under test is done using different classifiers such as support vector machines, random forest classifiers etc. The different types of hardware Trojans and its threat models are analyzed in [9-11]. In [11], hardware Trojan nets are identified by the proposed Support vector machine based classifier. The static detection process is discussed her, which does not require any test patterns for activating the Trojan model and also avoids the use of reference golden chip for the analysis. Low probability of detection rate when large data sets are involved is its limitations and it is also immune to noise signal. The Trojan features from the Trojan nets are extracted in [12] and a random forest classifier is applied to obtain the optimal set of Trojan features from the nets. Higher computational power is its limitations and requires more resources, since the model involves a lots of tree structure

Classification of Hardware Trojan using multi-layer neural network is discussed in [13]. The processing time is more to train the dataset for decision making which is its limitations since the hidden layers are based on feature set. In this paper [14], modelling an artificial neural network (ANN) is performed in-order to address the pattern recognition challenges. The statistical indicators metric is introduced for evaluating the performance of ANN models for various methods. The ANN model requires parallel processing power and is suitable for the numerical problems. Hence all the problems are to be converted to numerical values and feed to the model is the limitation of this method. In [15], a boundary net structure based Trojan classification, which used machine learning for initial classification of segregating the nets into Trojan and normal nets. The extraction of features is done based on the classification result are its limitations and the misjudgements of the nets are identified. An unsupervised learning approach is proposed in [16], in which local outlier factor (LOF) combined with Principle component analysis algorithm PL-HTD is adapted to visualize the Trojan nets. The theoretical approach is limited to the feature selection in different phase and is to be analysed. Hence the complexity of the system increases as scheme involves different filtering process. In [17-18], a deep learning algorithm is developed to overcome the limitations of machine learning, which extracts the features by itself and the nets are optimally classified based on the extracted parameter for large dataset with minimal computational complexity.

In [19] a security based game model is developed, which aid to the detection process of hardware Trojan. But the utility function is required to identify the best set of strategy for the threat model which is its limitation. An iterated elimination of dominant strategy is preferred in which one player dominates the other player to yield better payoff. The testing of digital circuit is also performed by modelling the game theoretical approach in [20-21]. A zero sum game between the attacker and defender is designed for testing the Trojans, where the defender fails to detect Trojan pays a high fine value is its limitation. A novel game model is framed in [22] to analyze the relation between the manufacturer (attacker) and the IC testing unit (defender). In this paper [23], a strategic game is modelled with two players to determine the suitable strategy for both defender and intruder. A software framework is also presented, which provides the mathematical exploration and also provides the solution of the game.

3. Proposed methodology

A deep learning based hardware Trojan detection technique is proposed in this paper and the Fig. 2 illustrates the design flow of the scheme.

For the gate level net list, the algorithms are developed to generate the optimal features of the internal nodes that can probably excite the Trojan sites. Golden circuit net list and the Trojan infected net list are synthesized using Synopsys. The proposed deep learning approach computes the optimal internal net features for enhancing the detecting probability and it is also validated using game theory approach. The different phases of the proposed scheme are:

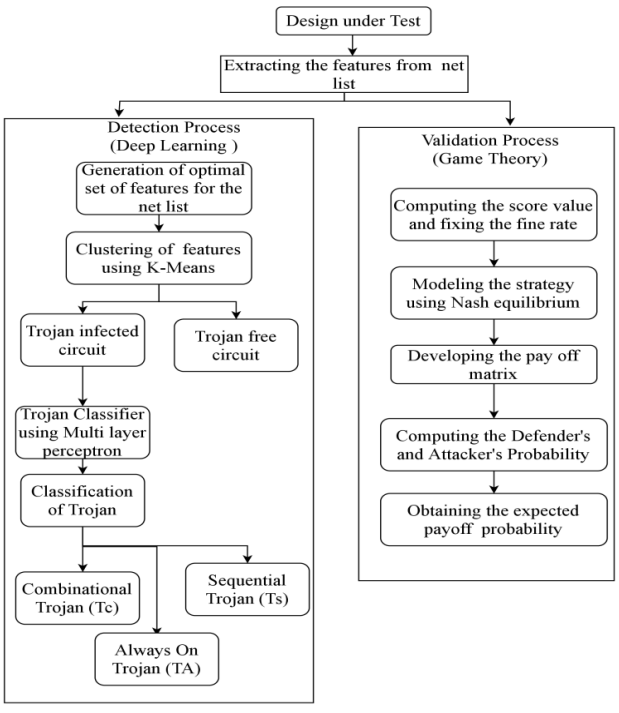


Figure. 2 The proposed methodology design flow

Optimal Trojan feature extraction phase, Detection phase and Validation phase.

3.1 Optimal trojan features extraction phase

In this phase, a gate level net- list for a specific net in ISCAS and Trust- HUB benchmark circuits are chosen for selecting optimal features. The complexity of the model is minimized by the efficient feature selection phase, which speeds up the learning phase of the classification algorithm.

3.1.1. Handcrafted feature extraction

The handcrafted algorithms are developed to extract the certain critical parameters like primary input, primary output, transition probability, level, connectivity and toggle rate.

Primary input (PI), primary output (PO):

The minimum level from the input node to the target net 'n' is PI and the minimum level in the design from the output node to the target net 'n' is PO.

Transition probability (TP):

The pseudo code 1 is developed to compute the transition probability of all net in the logic network. The nets with low transition probability are extracted, in which the malicious modules are more likely to be inserted.

Pseudo code 1 Determining Transition probability (TP) value

1: Scan the net list.
2: Export the PI, PO and the type of gate.
3: Initialize the primary input signal probability as 0.5
4: for each net in the net- list do
5: Determine gate type
6: Apply the probability equation at the output of each net
7: Compute the overall transition probability values for each net by
TP = Probability of logic 0 × Probability of logic 1.
8: Store TP value of each net in an array.
9: end for

Level (LE):

The number of logic gates that are cascaded between primary inputs to the target net'n'. This feature is more likely to increases during malicious module insertion.

Connectivity(CO):

Connectivity provides the total number of nodes to which a target net'n' is connected. If the target net triggeres on rare event, then this feature will impact the node associated with target net which may lead to functional change or leak of information.

Toggle rate(TO):

The toggle count features is the total number of transistions of a target net'n'for different range of test vectors. The net with maximum toggle rate are more sussecptable nodes and these nodes are more likely for trojan insertion.

In order to confirm the feature selection, Table 1 summarizes the average values of the above specified features for the genuine nets and Trojan nets for the benchmark circuit. The outcome of this phase will be a set of nets with efficient features. It is inferred that the average value of the Trojan nets seems to be larger when compared to genuine net. The nets with high fan in count are likely to be malicious nets, but some of the genuine nets also have maximum fan in count. Similarly, it is also observed that, the primary output is large for the genuine nets compared to Trojan nets. Hence only with the hand crafted features classifying the malicious nets are not sufficient, so the structural report is also analysed to extract the features relevant to the Trojan types.

3.1.2. Structural and power analysis:

The structural and power reports are generated using Synopsys tool to extract the relevant features for the genuine and malicious infected net-list.

Logic gate fan in (FI) and fan out (FO):

The total number of inputs of the logic gate away from the target net 'n'. The Trojan modules are activated only when it satisfies the triggering condition or on rare occurrence of the internal states. The total number of outputs of the basic gate away from the target net 'n'. Nets with maximum fan in and fan out are more likely to be infected nets but this itself is not enough to distinguish the Trojan nets from genuine nets.

Table 1. Average value of the feature

Net Type	LE	CO	PI	PO	Fan In	Fan Out
Trojan nets	4.5	1.34	4.5	3.5	1.25	1.5
Genuine nets	1.5	1.3	1.5	6.5	1.0	1.3

Table 2. Average value of the features based on synthesis tool

Circuit Net Type / Features	Genuine net	Trojan net
TN	0.29	0.46
SPR	0.48	0.64
TR	0.09	0.17
SP	0.007	0.043
LO	0.29	0.46
R	0.02	0.03
PN	2.1	2.7

Net load (LO) and pins (PN):

A device connected to the signal source consumes some power, which affects the circuit performance with respect to the primary output. The malicious module insertion may also lead to variation in the net load feature and the pins in the design.

Static probability (SPR):

This feature refers to the target net 'n' pertaining to the expected state of logic -0 or logic-1 and the maximum value of static probability are more prone for Trojan sites because of the rare occurrence of the logic state.

Resistance(R) and switching power (SP):

A resistive element connected between the load and the power source, which is used to monitor the power performance for varying load conditions.

Table 2 intimates the average value of the original nets versus Trojan nets for the features obtained as a result of applying the netlist to the structural analysis using the synthesis tool. It is observed that, the values in genuine nets are comparatively lesser than malicious nodes. The outcome of this phase will be a set of nets with efficient features which are more feasible to classify the anomaly in the design and are referred as Optimal Trojan Feature set (OTF).

3.2 Detection process

The OTF for each target net 'n' are extracted, but it is hard to fix the threshold value for the features to classify the malicious net from Trojan infected. Hence a deep learning based malicious module detection method is proposed, where it can automatically learn and extract the efficient features for the malicious and genuine nets from the OTF. It also classifies an unknown gate level net-list into a set of Trojan nets and genuine net using multilayer perceptron classifier. The Pseudo code 2 describes the flow of the proposed detection and classification of Trojan, which is composed of learning phase and classification phase.

Pseudo code 2 Detection and Classification of Trojans using Deep learning algorithm

1: Read the features list of the circuit.
2: Optimize the training features and set to X-train
3: Obtain the unsupervised feature representation of the data as input to auto encoder.
4: Generate a neural network that are densely connected for encoding the features
5: Generate a neural network that are densely connected for decoding the features
6: Decode the training data for extracting best set of features
7: Apply k-means algorithm for detecting the presence of Trojan.
8: Visualize the clusters with its labels
9: Implement Multilayer perception for classifying multiple Trojan types.

Learning phase:

In learning phase, the deep learning architecture is learned by many known malicious nets and genuine net by using the extracted feature values. For each net *'n'* the feature value are extracted and it is considered in a thirteen dimensional feature vector 'y_n', which is provided to the input layer of the deep learning architecture. The back propagation algorithm is employed in order to approximate the feature vector by minimizing the error value. The two level hidden layer is developed in the architecture, which densely connect the feature vectors of the input layer to that of the output layer. The feature vectors are encoded in the hidden layer using auto encoder and the encoded features are also labelled using k-means clustering algorithm. The labelled feature data pool are provided to the multilayer perceptron(MLP) classifier for training.

Classification phase:

In classification phase, the deep learning algorithm considers the extracted features as the training data and classifies the unkown gatelevel netlist into genuine nets and set of Trojan nets.In order to normalize the training data an unsupervised autoencoder algorithm is developed and the the structure of the auto encoder is created using a densensly connected encoder with Relu as activation function. The data from the encoder is provided to densely connected decoder which uses sigmoid as an activation fuction to decode it. The clustering of decoded data into as a genuine net or trojan net based the extracted features of the unknown net-list is performed using K-means algorithm. The result of the classification phase will be clustering the extracted features into original net or trojan nets. The

set of features is passed on to the multilayer perceptron only if the classifier identifies the extracted features as Trojan nets and it clusters the trojan infected nets into *combinational, sequential and always on Trojan*. Thus, the detection happens between genuine or Trojan and also between the type of Trojans there by increasing the overall reliability of the system.

3.3 Game theory based validation

The proposed methodology in the Fig. 2 is validated using the using the game theoritical approach for testing the hardware trojan.

Modelling the stratergies

In this game model, a decision making process is performed on an non- coorporative stratergies with two players attackers and tester. The attacker is the one who inserts the hardware trojan module and to minimize the validation, they insert single trojan at a time. The certain types of trojan like combinational, sequential and always on trojan classes are chosen for analysis. The attackers stratergies considered as inserting any one of the trojan from the specified trojan classes. Thus the attackers have three strateries (Y) that are denoted as TC, TS, TA. Thus the corresponding values for the attackers stratergy for different Trojan classes are represented as S1,S2,S3 and are computed from the score algorithm.The attackers fine value(F) is also decided from score computaion and a thershold value (ΔF) is also fixed,which is imposed on the attacker when a trojan is identified. This model is also considered as a rational approach, where the attacker intrusion of malicious circuit minimizes the likelihood for detection and the defender aims for maximizing the the payoff value.

Generating the payoff matrix

A mixed strategy Nash equilibrium is employed in-order to get the solution for the game model and also it allows us to determine the two player's frequency of choosing the strategies at equilibrium condition. A payoff matrix is generated using the score value of the Trojan inserted circuit along with the fine values to be imposed on the attackers. For this illustration, the score value of the attacker's strategy is computed from the score algorithm which is performed by adding all the features for each net. The maximum value (M), minimum value (N) from the nets is considered and the score value is clustered into number of attacker strategy groups by Eq. (1).

$$Si = \left[\frac{M+N}{Y} \times (i)\right] \; for\; i = 1,2,..Y \qquad (1)$$

A sample circuit of ISCAS 85' C17 bench mark circuit the maximum net value M=7, minimum net value is N=4, the score value is computed by equation 1 to obtain the clusters S_1 =4, S_2=8, S_3 =12. In this model, the defender considers a mixed stratergy in which two trojan types are tested simultaneously out of three trojan classes and therefore the defender has $3C_2$ possible stratergies represented as T_CT_S, T_ST_A, T_CT_A. The fine value F is also decided for the corresponding circuit based on the Eq. (2).

$$Fine(F) = max(Si) + \Delta F \qquad (2)$$

where ΔF is choosen as 2 inorder to get maximum positive payoff for the defender. The attackers and defenders payoff for various fine values are also analysed. The Table 3 specifies the pay off matix generated for the sample ciruits of C17 and C6288 of ISCAS'85 benchmark circuit.

Determining the expected pay off

The game model is incorporated using mixed strategy Nash equilibrium and the best response of the defender to the attacker intruding a sequential Trojan of type TS is to play TCTS or TSTA. So that the malicious module can be identified and the attacker can be imposed with a fine 'F' to get a negative pay off value to the attacker and positive pay off value to the defender. If the defender choice is to play TSTA then the attacker's best choice is to play TC and go undetected. Thus the pure stratery nash equilibrium is not suitable for the game model shown in Table 3 because of its circular reasoning. The defender chooses the mixed stratergy to get the maximum payoff against the attacker, who seeks to intrude trojan that is undetectable with maximum impact. From the pay off matrix, the probability is computed by the online mixed stratergy solver for the defender and attacker. The probability of defender's strategies TCTS, TSTA, TCTA are represented as p1, p2, 1-p1-p2 and that of attackers strategies TC, TS, TA are represented as q1, q2, 1-q1-q2. The expected payoff values for the player's are calculated using the Eq. (3).

Table 3. Pay off matrix generation for hardware trojan detection

Bench mark Circuit		Defender (C17)			Defender (C6288)		
		T_C T_S	T_S T_A	T_A T_C	T_C T_S	T_S T_A	T_A T_C
Attacker	T_C	-F,F	12,-12	-F,F	-F,F	63,-63	-F,F
	T_S	-F,F	-F,F	8,-8	-F,F	-F,F	42-42
	T_A	4,-4	-F,F	-F,F	21,-21	-F,F	-F,F

$$EP = (S1 \times p1 \times q1) + (S2 \times p2 \times q2) + (S3 \times (1 - p1 - p2) \times (1 - q1 - q2)) \quad (3)$$

4. Result and analysis

The proposed methodology is validated using ISCAS'85, ISCAS'89 and Trust-HUB circuits. An optimal feature set is generated using deep learning algorithm for the circuit under test and the nets with varying feature values are likely to be Trojan triggered nets. The threat models intruded into the benchmark circuit are combinational (Tc), sequential (Ts) and always on Trojan (TA) modules. These threat modules alter the circuit functionality when it is triggered on rare conditions. Synopsys DC compiler is used to synthesis the infected design and the golden design.

Table 4 illustrates the comparison of different features values of the genuine nets and Trojan infected nets, obtained as a result of intruding a ring oscillator based Trojan module in the design. The outcome of extraction phase will be selecting a set of Trojan nets with varying feature value for the benchmark circuits. It is observed that when the Trojan module is intruded in the design, the feature values of the corresponding Trojan nets are varying with respect to the normal nets. Thus a sample of C17 and C432 circuit is shown along with its Trojan nets represented as NT, which shows the optimal features are extracted for detecting the Trojan module in the design for high reliability.

The maximum values of the extracted features are listed in Table 5 and the analysis is done by inserting the three different types of Trojan individually in the benchmark circuit. It is observed that the maximum values of the extracted feature of the Trojan circuits T_C, T_S, T_A is comparatively higher than the genuine circuits. It is observed that the variation in extracted optimal feature values indicate the presence of Trojan module in the design.

Table 6 reveals the processing time for executing the deep learning algorithm, achieved as a result of inserting (i) Tc, Ts, T_A Trojan modules individually and (ii) all the Trojans at the same time for the benchmark circuits. It is observed that for the C7552 circuit which has large number of primary input and nets compared to other circuits consume only an average processing time of 9.02 s and for S13207 it takes 13.56 s. Thus the Table 6 depicts that the average processing time required for computing the deep learning algorithm for complex circuits is less. It is also inferred that the extracted optimal features require less processing time, with minimal computational complexity for detecting the hardware Trojan.

The Fig. 3 shows the total number of nets affected by the insertion of Trojan module for ISCAS'85, ISCAS'89 and Trust-HUB circuits. The combinational (Tc), sequential (Ts) and always on Trojan (T_A) are designed in order to observe the feature values for specific trigger conditions

Table 4. Comparison of different features values of the genuine nets and trojan infected nets

Benchmark circuit	Nets	LE	CO	PI	PO	FI	FO	LO	RE	PN	TN	SPR	TO	SPO
C17	N1	0	1	0	3	1	1	0.24	0.02	2	0.243	0.247	0.0974	0.0058
	NT1	**0**	**1**	**0**	**8**	**1**	**1**	**0.24**	**0.02**	**2**	**0.243**	**0.616**	**0.123**	**0.0073**
	N19	2	1	2	1	1	1	0.24	0.02	2	0.243	0.616	0.123	0.0073
	NT19	**7**	**2**	**7**	**1**	**2**	**1**	**0.54**	**0.04**	**3**	**0.544**	**0.616**	**0.123**	**0.0164**
	N23	3	1	3	0	1	1	0.24	0.02	2	0.243	0.568	0.1447	0.0086
	NT23	**8**	**1**	**8**	**0**	**1**	**1**	**0.24**	**0.02**	**2**	**0.243**	**1**	**0**	**0**
C432	N118	1	1	1	16	1	1	0.24	0.02	2	0.243	0.284	0.1881	0.0112
	NT118	**6**	**2**	**6**	**21**	**2**	**1**	**0.24**	**0.02**	**2**	**0.243**	**0.5**	**0.1**	**0.006**
	N119	1	2	1	16	2	1	0.54	0.04	3	0.243	0.5	0.1	0.006
	NT119	**6**	**4**	**6**	**21**	**4**	**1**	**0.54**	**0.04**	**3**	**0.544**	**0.5**	**0.1**	**0.0133**
	N123	1	2	1	16	2	1	0.54	0.04	3	0.243	0.5	0.1	0.006
	NT123	**6**	**2**	**6**	**21**	**2**	**1**	**0.54**	**0.04**	**3**	**0.544**	**0.5**	**0.1**	**0.0133**
	N163	2	2	2	15	2	1	0.24	0.02	2	0.243	0	0	0
	NT163	**7**	**2**	**7**	**20**	**2**	**1**	**0.24**	**0.02**	**2**	**0.544**	**0.76**	**0.0949**	**0.0126**

Table 5. Maximum values of the extracted feature of the benchmark circuit

Bench mark circuit	LE	CO	PI	PO	FI	FO	LO	RE	PN	TN	SPR	TO	SPO
C17	3	2	3	3	2	1	0.54	0.04	3	0.544	0.753	0.1447	0.0164
T_C	3	2	3	3	3	1	0.86	0.07	4	0.544	0.749	0.1921	0.0159
T_S	5	4	5	5	4	1	1.2	0.09	5	1.203	0.757	0.1437	0.0358
T_A	8	2	8	8	2	2	0.54	0.04	3	0.544	1	0.1432	0.0164
C432	17	9	17	17	19	1	9.88	0.76	20	9.876	0.921	0.2784	0.493
T_C	18	9	18	18	20	1	9.88	0.76	22	9.876	1	0.2801	0.493
T_S	18	9	18	19	21	1	9.88	0.76	22	9.876	1	0.2737	0.4869
T_A	17	9	17	17	19	2	9.88	0.76	20	9.876	1	0.2804	0.493
C1908	35	25	35	35	25	1	15.22	1.18	26	15.218	1	0.7531	1.3853
T_C	35	16	35	37	26	1	15.22	1.18	27	15.218	1	0.7531	1.3853
T_S	37	16	37	40	28	2	15.22	1.18	28	15.218	1	0.7506	1.4025
T_A	40	16	40	40	26	1	15.22	1.18	26	15.218	1	1.014	1.3853
C6288	119	34	119	119	16	1	7.48	0.58	17	0.863	1	0.8035	0.1699
T_C	120	36	120	120	16	1	7.48	0.58	17	0.863	1	0.8035	0.1699
T_S	122	16	122	122	16	2	7.48	0.58	19	1.951	1	0.8077	0.1704
T_A	124	16	124	124	16	1	7.48	0.58	18	1.203	1	0.8047	0.1701

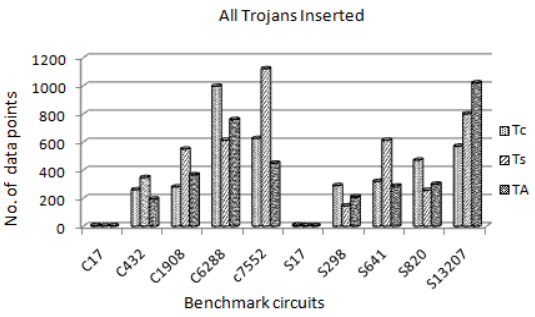
Table 6. Processing time for executing deep learning algorithm

Bench mark circuit	No. of inputs	Normal nets	Processing Time(s)			
			Tc	Ts	T_A	T_c T_s T_A
C17	5	11	0.4	0.42	0.42	0.48
C432	36	196	1.69	1.71	1.75	1.80
C1908	33	913	1.69	1.71	1.75	1.80
C6288	32	2445	2.17	2.76	2.80	2.83
C7552	207	3720	8.95	9.0	9.03	9.12
S27	4	14	0.1	0.14	0.18	0.23
S298	3	133	0.3	0.32	0.34	0.38
S641	35	433	2.78	2.89	3.10	316
S820	18	309	1.78	1.89	2.10	2.13
S13207	62	8141	13.1	13.2	13.8	14.0

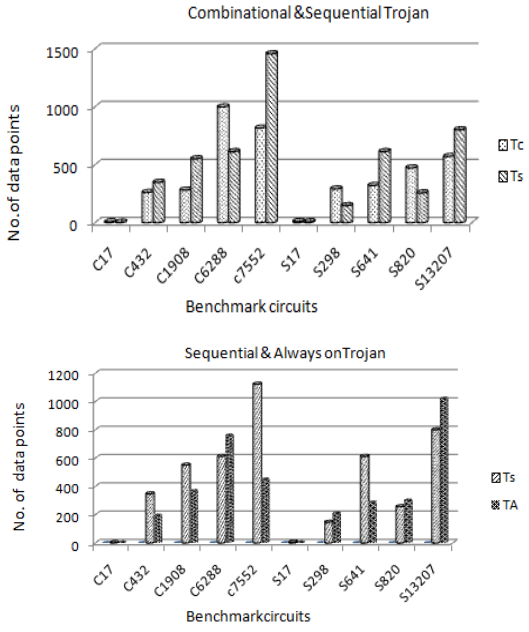
to validate the presence of Trojan. The Fig. 3 (a) shows the number of affected nets by inserting different types of Trojan simultaneously to the bench mark circuit. It is observed that for C7552 circuit, the number of nets affected by Tc type Trojan are 625, Ts type are 1122 and T_A type Trojan are 450. Thus the extracted features from the deep learning algorithm classify the intruded Trojans into its corresponding Trojan types. Similarly Fig. 3 (b) shows the number of nets affected by simultaneously inserting (i) combinational & sequential Trojan, (ii) always on & sequential Trojans modules in the circuit. It is observed that the nets are clustered only in its corresponding Trojan type. The circuit under test is introduced with only one Trojan type and is shown in Fig. 3 (c). Thus it is inferred that the optimal features are extracted, as a results the data sets are classified to the corresponding Trojan types inserted in the design which provides high reliability of the system.

The accuracy for evaluating the deep learning algorithm for the benchmark mark circuit is manifested in Fig. 4. For the circuit C7552, the accuracy is 95.4% for Trojan Type Tc, 95% for Type Ts, 95.25 for Trojan Type TA and has an accuracy of 97.31 % with all types of Trojan inserted. It is inferred that the accuracy for the different Trojan types inserted simultaneous is high compared to the single Trojan insertion. Hence the extracted features by the deep learning algorithm is capable of classifying and detecting multiple Trojans in the design with an average accurate rate of 96.02% for ISCAS'85 and ISCAS'89 circuits.

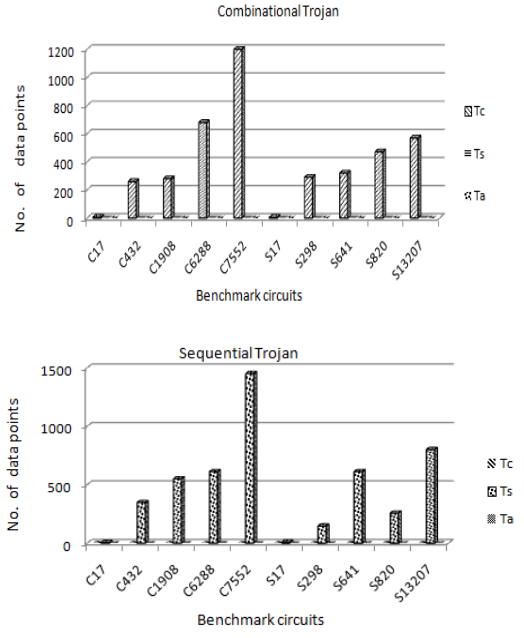
The attackers probability of the game theorical approach with high score value on combinational type trojan is shown in Fig. 5. It is observed that the probability of the attacker inserting the hardware trojan minimizes the score value of corresponding



(a)



(b)



(c)

Figure. 3 Number of data points infected by the trojan: (a) all types of trojan inserted, (b) two types of trojan inserted, and (c) single trojan inserted

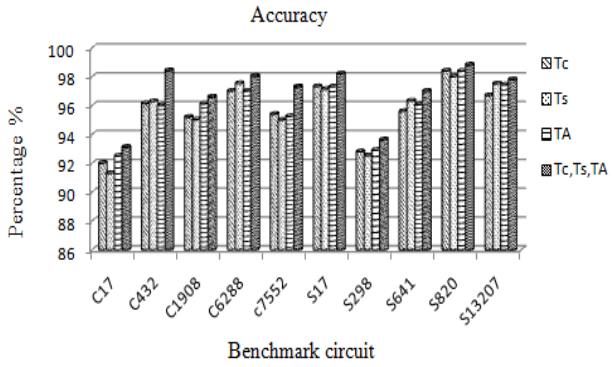


Figure. 4 Accuracy of the deep learning algorithm with different trojan types

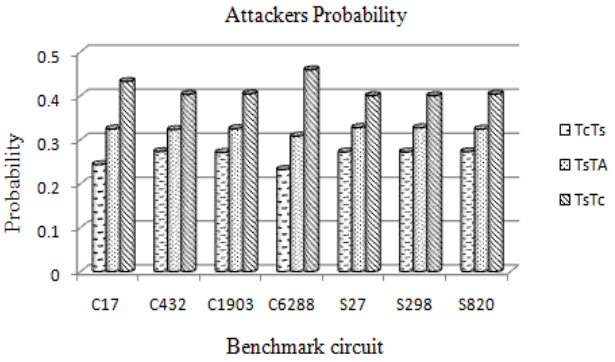


Figure. 5. Mixed strategy Nash equilibrium for attacker

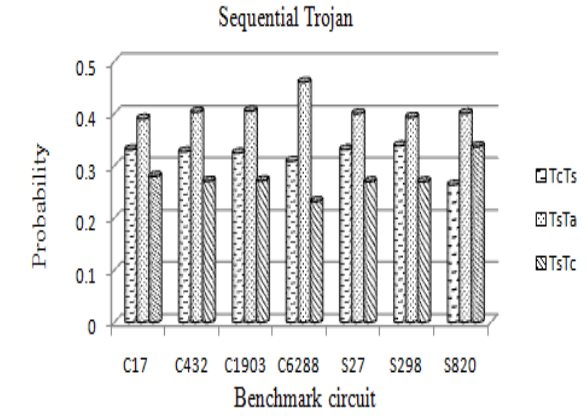
trojan type and high score value trojan modules are less likely inserted by the attackers as these high score trojan types are more effectively protected by the defender. Hence it is observed that the combinational type which has high score value are less choosen by the attacker.

The Fig. 6 shows the probability of defender detecting the hardware trojan in the game theory. The different trojan types are inserted in the design one at a time and the probability for the defender is shown in the Fig. 6 (a). It is observed that the probability of the defender is high for the corresponding trojan types, which infers that the game theoritical model optimally detects the trojan type based on the score value. Fig. 6 (b) shows the probability value for Trust-HUB circuits and it is observed that the defender has maximum probability for the corresponding trojan type inserted in the trusthub circuit. The optimal results are obtained by this game theory model and it proves that the features extracted from the deep learning algorithm is an optimal set for detecting the hardware Trojan.

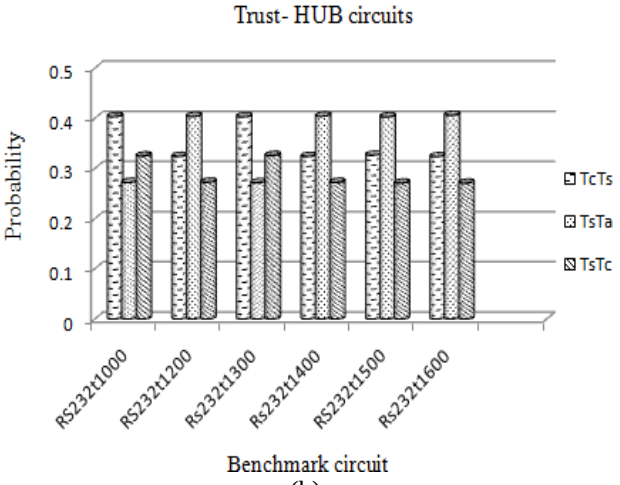
The Expected payoff for the defender is shown in Fig. 7 in which the fine value assigned to the game model yields a negative payoff for the attacker and a that the expected payoff for the defender is high

Table 7. Comparison of performance measures between existing method and proposed method

Trust-HUB circuits	True positive rate(TPR)				True negative rate (TNR)				Accuracy (%)			
	[13]	[15]	[16]	Ours	[13]	[15]	[16]	Ours	[13]	[15]	[16]	Ours
RS232-T1000	100	100	50	93.3	24.00	98.2	96.43	82.9	32.58	98.4	94.85	95.23
RS232- T1100	78.00	69.4	45.45	100	25.00	96.8	96.43	91.57	30.36	93.8	94.50	96.08
RS232- T1200	91.00	100	46.15	98.00	55.00	95.8	97.14	89.17	58.79	96.3	94.56	95.12
RS232-T1300	86.00	100	57.14	96.5	65.00	99.7	96.04	72.08	66.93	99.7	95.06	95.05
RS232- T1400	100	100	41.67	98.00	15.00	97.1	96.40	85.93	27.07	97.5	94.14	96.82
RS232- T1500	82.00	97.4	45.45	94.05	47.00	97.5	96.45	92.76	51.24	97.5	94.54	97.63
RS232- T1600	100	96.4	44.44	97.00	28.00	98.3	96.11	88.24	34.23	98.1	94.52	95.63



(a)



(b)

Figure. 6 Mixed strategies nash equilibrium for defender: (a) probability of defender for single trojan type and (b) trust-HUB circuits

positive payoff value for the defender. It is inferred compared to that of attacker's payoff, which clearly guarantees that the defender chooses the best stratergy against attacker for detecting the Trojan type. Thus proposed methodology is validated by the

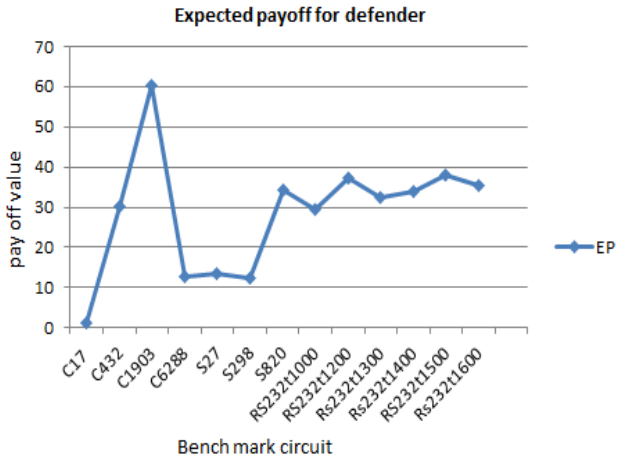


Figure. 7 Expected pay off for defender

mathematical game theory approach for detecting the presence of malicious anamoly in the digital circuit.

5. Comparision with existing techniques

The existing machine learning technique are classified into supervised and unsupervised learning. The supervised learning algorithm like SVM based classifier [11], Multilayer neural network [13] method and the unsupervised learning algorithm like boundary net structure [15], PL-HTD [16] are compared with deep learning scheme. The proposed deep learning technique are more focused, since it overcomes the drawback of the machine learning algorithm and it also improves the performance metric. The proposed method is compared with the related work for the Trust–HUB circuit and the performance metric evaluated are summarized in Table 7. It is observed that, comparing with MNN the deep learning algorithm performs better with an increase in average TPR by 5.69 % and average accuracy by 52.27 %. As for PL- HTD technique, the average value of TPR and accuracy of our method increases by 49.50%, and 1.27% respectively. It is deduced that the higher TPR leads to better detection of Trojan nets and the rate of misjudgement of

genuine nets to be reduced, which highlights the feature extraction in the extraction phase.

It is also inferred that deep learning based Trojan detection performs better in TPR metric compared to the existing machine learning techniques, which indicated the nets affected by Trojan are perfectly classified. Although the resulting TNR was to some extend behind the boundary based and PL-HTD methods, but the normal nets of the proposed method focus on feature extraction for classifying the Trojan net without compromising the classification of genuine nets subsequently reduces the manual computation. Compared with boundary based our proposed method average TPR increases by 1.95 % but the average accuracy is only 1.45 % smaller but the overall accuracy of our scheme is 96.25% for ISCAS and Trust –HUB circuit without compromising the reliability of the design. To summarize, it is analyzed that among the classifiers (MNN, PL-HTD, Boundary based), the proposed deep learning technique performs the best in terms of the classification rate with the optimal feature set.

6. Conclusion

In this work a deep learning-based hardware Trojan detection and classification technique is proposed which develops algorithms for extracting the circuit parameters. The structural reports of the gate level netlist are generated in this work to extract Trojan features and make the detection process a static. The simulation is demonstrated on ISCAS'85, ISCAS'89 and Trust-HUB circuits which shows that the proposed technique achieves an average accuracy of 96.25% and average True positive rate of 96.69% at minimum processing time. The deep learning-based detection technique is also validated using game theoretical approach which provides maximum probability for the defender by ensuring the probability of detecting the Trojan in the benchmark circuit. The major challenges of the algorithms developed are addressing a specific threat model attacks and providing the solutions. It has to be extended and trained for different attack or threat models. In future work would be incorporating the delay parameters as a static timing analysis in the proposed technique for detection process and also modelling the denoising auto encoder architecture to minimize the prediction error on a supervised learning scheme.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, methodology, software, validation, writing, original draft preparation, Priyatharishini Murugesan; review and editing, supervision, project administration, Nirmala Devi Manickam.

Acknowledgments

Currently the article is not receiving any funds under any organization.

References

[1] S. Bhasin and F. Regazzoni, "A survey on hardware Trojan detection techniques", In: *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2021-2024, 2015.

[2] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection", *IEEE Design and Test of Computers*, Vol. 27, No. 1, pp. 10–25, 2010.

[3] H. Salmani, M. Tehranipoor and J. Plusquellic, "A novel technique for improving hardware Trojan detection and reducing Trojan activation time", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 20, No. 1, pp. 112–125, 2011.

[4] M. Banga and M. Hsiao, "A region based approach for the identification of hardware Trojans", *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 40–47, 2010.

[5] D. K. Karunagaran and N. Mohankumar, "Malicious Hardware Trojan Detection by Gate level minimization 90nm Technology", In: *Proc. of Fifth International Conf. on Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1-7, 2014.

[6] S. Dupuis, B. A. Papa Sidi, G. Di. Natale, M. L. Flottes, and B. Rouzeyre, "A novel hardware logic encryption technique for thwarting illegal overproduction and hardware Trojans", In: *2014 IEEE 20th International On-Line Testing Symposium (IOLTS)*, pp. 49-54, 2014.

[7] B. Liu and B. Wang, "Embedded reconfigurable logic for ASIC design obfuscation against supply chain attacks", In: *Proc. of Design, Automation & Test in Europe Conf. & Exhibition (DATE)*, pp. 1-6, 2014.

[8] M. Priyatharishini, M. Nirmala Devi, "A Compressive Sensing based optimal test pattern generation for Hardware Trojan Detection", *International Journal of Electrical and*

Computer Engineering, Vol. 9, No.5, pp. 4035-4043, 2019.

[9] J. Popat, U. Mehta, "Transition probabilistic approach for detection and diagnosis hardware Trojan in combinational circuits", In: *Proc. of IEEE Annual India Conf. (INDICON)*, pp. 1-6, 2016.

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

[11] K. Hasegawa, M. Oya, M. Yanagisawa, and N. Togawa, "Hardware Trojans Classification for Gate-level Net-lists based on Machine Learning", In: *IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pp. 203-206, 2016.

[12] K. Hasegawa, M. Yanagisawa, and N. Togawa, "Trojan-feature Extraction at Gate-level Net-lists and Its Application to Hardware-Trojan Detection Using Random Forest Classifier", In: *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-4 ,2017.

[13] K. Hasegawa, M. Yanagisawa, and N. Togawa, "Hardware Trojans Classification for Gate-level Net-lists Using Multi-Layer Neural Networks", In: *IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pp. 227-232 IEEE, 2017.

[14] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, A. M. Umar, O. U. Linus, H. Arshad, A. A. Kazaure, U. Gana, and M. U. Kiru, "Comprehensive review of artificial neural network applications to pattern recognition", *IEEE Access*, Vol. 4, No. 7, pp. 158820-158846, 2019.

[15] K. Hasegawa, M. Yanagisawa, and N. Togawa, "A hardware-Trojan classification method utilizing boundary net structures", In: *Proc. of IEEE International Conf. on Consumer Electronics (ICCE)*, pp. 1-4, 2018.

[16] C. Dong, Y. Liu, J. Chen, X. Liu, W. Guo, and Y. Chen, "An Unsupervised Detection Approach for Hardware Trojans", *IEEE Access*. 2020. (In press)

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT press, Cambridge, 2016.

[18] Introduction to deep learning, Available*:* http://introtodeeplearning.com.

[19] J. Graf, "Trust games: How game theory can guide the development of hardware Trojan detection methods", In: *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 91-96, 2016.

[20] C. A. Kamhoua, H. Zhao, M. Rodriguez, and K. A. Kwiat, "A Game-Theoretic Approach for Testing for Hardware Trojan", *IEEE Transactions on Multi-Scale Computing Systems*, Vol. 2, No.3, pp. 199-210, 2016.

[21] S, R. Hasan, C. A. Kamhoua, K. A. Kwiat, L. Njilla, "A Novel Framework to Introduce Hardware Trojan Monitors using Model Checking Based Counterexamples: Inspired by Game Theory", In: *IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 853-856, 2018.

[22] W. Saad, A. Sanjab, Y. Wang, C. A. Kamhoua, K. A. Kwiat, "Hardware Trojan detection game: A prospect-theoretic approach", *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 9, pp. 7697-7710, 2017.

[23] J. Graf, W. Batchelor, S. Harper, R. Marlow, E. Carlisle, P. Athanas, "A practical application of game theory to optimize selection of hardware Trojan detection strategies", *Journal of Hardware and Systems Security*, Vol. 4, No. 2, pp. 98-119, 2020.