PAPER

A Multiobjective Approach for Side-Channel Based Hardware Trojan Detection Using Power Traces

Priyadharshini MOHANRAJ[†a)], *Student Member and* **Saravanan PARAMASIVAM**[†b)], *Nonmember*

SUMMARY The detection of hardware trojans has been extensively studied in the past. In this article, we propose a side-channel analysis technique that uses a wrapper-based feature selection technique for hardware trojan detection. The whale optimization algorithm is modified to carefully extract the best feature subset. The aim of the proposed technique is multiobjective: improve the accuracy and minimize the number of features. The power consumption traces measured from AES-128 trojan circuits are used as features in this experiment. The stabilizing property of the feature selection method helps to bring a mutual trade-off between the precision and recall parameters thereby minimizing the number of false negatives. The proposed hardware trojan detection scheme produces a maximum of 10.3% improvement in accuracy and reduction up to a single feature by employing the modified whale optimization technique. Thus the evaluation results conducted on various trust-hub cryptographic benchmark circuits prove to be efficient from the existing state-of-art methods.

key words: *feature selection, hardware trojans, whale optimization algorithm, binary conversion, power traces, machine learning*

1. Introduction

Research in the area of Hardware Trojan (HT) detection is prevailing for more than a decade now. Due to the increasing complexity of design and time-to-market constraints, high reuse prevails in Integrated Circuit (IC) industry. In the different phases of the IC design cycle, design, and fabrication are the most vulnerable stages of trojan insertion [1]. Due to the involvement of third-party Intellectual Property (IP) blocks, design tools, standard libraries, and foundries for fabrication, the two stages: design and fabrication are considered untrusted. Henceforth the need for HT detection is essential and never-ending. A brief outline of the semiconductor IC supply chain is depicted in Fig. 1.

An HT is any addition or modification to a circuit or system with malicious intention. An HT has malicious goals such as controlling information, leaking sensitive information like a secret key, reducing circuit reliability, and so on. Thus the system will start malfunctioning before its lifetime expires. An untrusted IC is one that fails to deliver the required functionality or one which has an HT inside the system. HT detection is a way to establish trust in integrated circuits. The design of an HT is stealthy such that they are dormant until a rare event triggers it. Even though the post-fabrication testing and verification happen in the IC design cycle, the rarely triggering nature of the HTs fails to be detected. The HT detection approaches are classified as destructive and non-destructive. The destructive techniques are expensive and time-consuming. Also, just by testing a few samples judgment about the entire manufactured lot cannot be done. The methodologies [2], [3] however aim to detect any gate alterations in the circuitry, they are not a practical approach.

The side-channel analysis belongs to the non-destructive category where the effect of HT is observed in the physical parameters like transient current, power, path delay, or electromagnetic radiation. These measures are compared with the reference expected values to observe the effect of extra circuitry. A challenge to encounter in side-channel analysis is the effect of process variations and noise which masks the trojan effect when the trojan is not triggered. In literature, an attempt to magnify the side-channel impact of a trojan is presented in [4]. This technique assures to detect the small sized trojans which may be smaller than the infected sensor. To isolate the effect of a trojan circuit from process noise [5], the relationship between dynamic current and operating frequency was considered. A vector test generation was proposed to improve the detection rate. Also, integration with logic testing was done to detect small-sized trojans. Another concern in side channel techniques is the reference values. The trust-hub benchmarks [6] have been treated as a source for study by many researchers which are considered in this proposed work.

Several side channel activity based HT detection approaches have been explored in the past. Path delay was used as a side channel parameter to generate the fingerprints of an IC family [7]. Further Principle Component Analysis (PCA) was used to find factors that project the major trends of the original dataset in a lower dimensionality. Power simulation analysis based detection was considered to develop a set of fingerprints from several ICs in a batch and the remaining ICs were verified using statistical tests compared with the fingerprints [8]. [9] is the first approach to employ a Genetic Algorithm (GA) to select optimal frequencies of ring oscillators (ROs) from the ring oscillator network (RON). Many Machine Learning (ML) based HT detection schemes [10]–[17] have been explored in the literature where gate-level netlist features were utilized. Area and power related features [18] were identified and extracted from the gate-level netlist. The gradient boost model was used for HT classifica-

Manuscript received May 5, 2023.
Manuscript revised July 14, 2023.
Manuscript publicized August 23, 2023.

[†]The authors are with the Dept. of Electronics and Communication Engineering, PSG College of Technology, Coimbatore, India.
a) E-mail: 1907rl05@psgtech.ac.in
b) E-mail: dps.ece@psgtech.ac.in

DOI: 10.1587/transfun.2023EAP1050

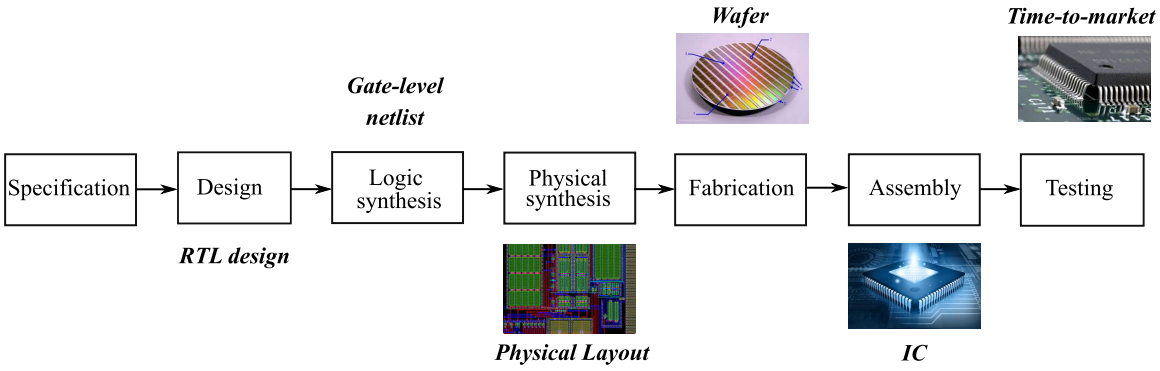


Fig. 1 Semiconductor IC supply chain.

tion and tested using the trust-hub benchmark circuits. Also, techniques using testability based features for HT detection have been studied [19]–[22].

Some power consumption based HT detection approaches [23]–[35] were studied in the literature. In [23] a hardware trojan detection method in which the statistical parameters of power were calculated and used as feature vectors. The features were subject to PCA and further classified using Linear Discriminant Analysis (LDA). A machine learning approach based on isolation forest was proposed by [24] for HT detection employing the power traces as side channel information. The work [25] is a support vector machine (SVM) based detection approach to detect HTs using the power dissipated in a circuit. A Gaussian kernel function is chosen and cross-validated to select the best C penalty and σ kernel function parameter. For a better analysis of the SVM model, more trigger circuits and conditions are to be analyzed. This helps to better generalize the technique. An HT detection approach that employs a hierarchical temporal memory to detect the anomalies in the circuit under test is [26]. This technique makes sure the trojan is triggered and power consumption samples are obtained for both trojan and trojan-free conditions. The work on power traces based HT detection [27] employing Softmax Regression has considered the 128-bit AES circuit referring to the trust-hub benchmarks. A runtime HT detection approach that involved the power profile of MC8051 without a golden model in [28]. The controller was implemented in hardware description language to extract the power and the trained model was embedded in the controller for online HT detection employing the k-nearest neighbors (k-NN), naive Bayesian classification (NBC), decision tree (DT), and deep learning (DL) techniques. The work [29] based on cluster analysis of the power profile of AES was developed to prove the shortcomings of Euclidean distance compared to Mahalanobis distance for HT detection. Research work in power related artificial neural network (ANN) based HT detection works have also been explored. The approaches [30]–[32] are power consumption based HT detection methods that utilized the back propagation neural network (BPNN) for HT detection. [33] is an HT detection method where the extreme learning machine (ELM) technique was used to detect trojans in a self-designed sample circuit. A mini AES-8 circuit is considered in this work [34] for HT detection. PCA is used in combination with the particle swarm optimization algorithm to arrive at the optimal set of features. The golden-free HT detection technique [26] is able to detect trojans with a trigger mechanism alone and the unsupervised model produces more false negatives, so accuracy cannot be improved above a certain value. Also, the technique [35] was able to reduce the false negative rate only when the size of the trojan was increased. The technique detects foundry inserted trojans but fails to detect trojans inserted by a rogue employee in the in-house design team or third-party vendor. Thus, the HT detection technique with reference model is more capable of addressing always-on, rarely triggered types of trojan in the design and fabrication phases with better accuracy and minimum false negatives.

In this proposed work, the aim is to detect HT with a minimal number of traces. Rather than increasing the number of traces, improving the technique and model is predominant. To bridge the gap between the number of traces and the ML technique, the improved feature selection technique plays a major role. The main contributions of this work are summarized as follows:

- A side-channel analysis based hardware trojan detection method is proposed to efficiently detect HTs without any additional hardware overhead.
- The Whale Optimization Algorithm is modified and employed for feature selection from the power traces which is a first of its kind in the existing approaches for trojan detection.
- The binary conversion of the continuous optimized solution is done and the best feature subset is formulated.
- The k-nearest neighbor classifier is used as an evaluator with the optimization algorithm.
- The various experimental results prove that the proposed HT detection scheme helps to build an optimum model with a reduced number of features for the standard trust-hub benchmark circuits.

The rest of this paper is organized as follows, Sect. 2 presents a detailed explanation of the power traces, ML classifier, and the modified feature selection technique adopted, Sect. 3

discusses the experiment and performance analysis, Sect. 4 is an experiment on ISCAS'89 benchmark circuits and Sect. 5 concludes the work.

2. Proposed Methodology

2.1 Threat Model

The threat model assumes that the attacker can insert the hardware trojan either at the design phase or in the fabrication phase of the supply chain. The payload of the inserted trojans can leak secret information in a cryptographic circuit or can modify the output in a non-cryptographic circuit. Furthermore, considerations for trojan detection using power side-channel information are also incorporated into the study environment. This includes designing the IC to have appropriate power measurement points and ensuring that the power consumption can be accurately monitored during its operation. When malicious hardware is inserted into an IC, the additional computation due to the added elements is reflected in the power consumption. The proposed work requires the collection of power traces in the presence and the absence of hardware trojan for training the machine learning model. In order to obtain the power traces in the absence of hardware trojan, IC under test should be fabricated in a trusted foundry. On the other side, the power traces in the presence of hardware trojan can be obtained by using trojan-infected third-party IPs as well as malicious own IPs in the in-house design team. It is assumed that the evaluator who prepares the reference trojan-free and trojan-infected power traces has a deep knowledge of the design, access to power measurement setup, and expertise to implement the detection technique.

2.2 Sources of Power Dissipation

The power dissipated by a chip is composed of static and dynamic power. The ideal power consumption of a golden IC is:

$$P_{ideal} = P_{stat} + P_{dyn} \tag{1}$$

In the real scenario, power consumption is always affected by a noise like process, supply voltage, temperature (P_{PVT}), and measurement noise (P_M). So the actual power consumption of an IC will be:

$$P_{real} = P_{ideal} + P_{PVT} + P_M \tag{2}$$

When an HT is inserted into the circuit the extra power consumption will add up to the circuit. Hence the power consumption can be expressed as:

$$P_{Hreal} = P_{real} + P_{HT} \tag{3}$$

where P_{Hreal} is the actual power dissipated once an HT is inserted in the circuit. The concept of side-channel analysis is employed in this work. Side-channel analysis (SCA) is a non-destructive, non-invasive approach to HT detection by observing the effect of trojan on physical parameters like transient current, power, path delay, and electromagnetic radiation. The main advantage of SCA is that even if a trojan does not cause any malfunction, the presence of extra circuitry will be reflected in these parameters. Also, the calculation of power compared to other parameters involves a practically feasible setup. The brief outline of the proposed feature selection-based trojan detection method is drafted in Fig. 2.

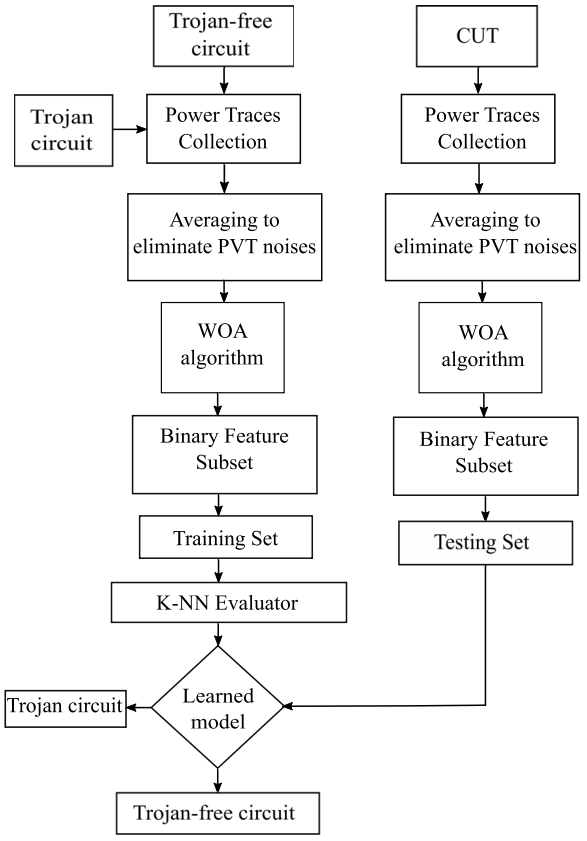
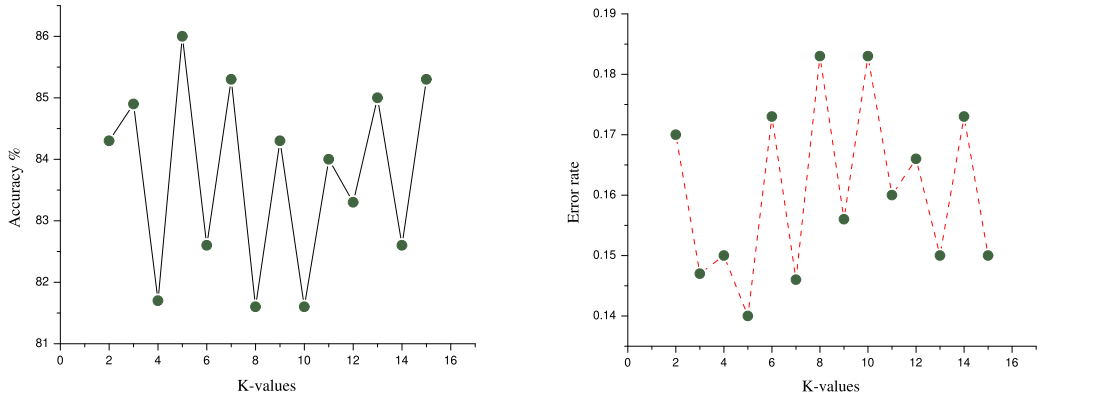


Fig. 2 Proposed feature selection based trojan detection method.

2.3 Data Acquisition

In this experiment, 750 power traces (each trojan-free and trojan) with 10000 sampling points were collected to form the dataset matrix. Various preprocessing steps were followed in the past like removing high frequencies [24] and averaging to improve trojan detection. As discussed in the previous section, the noise due to process variations suppresses the effect of HT. So averaging is a very basic and familiar technique adopted in eliminating PVT noises. Even though averaging reduces the noise levels to an extent, certain pre-processing techniques are used extensively in the literature to work on high-dimensional data in many areas. The two approaches for dimensionality reduction are feature extraction and feature selection. Feature extraction transforms the features into entirely new values based on the combinations of the original values. The newly extracted features are not easily



(a) (b)

Fig. 3 Variation of (a) accuracy and (b) error rate with k-value.

interpretable, unlike the original feature values.

2.4 Feature Selection

Feature Selection is a pre-processing step that does not transform into new features but creates a subset of the raw features which provides better interpretability. Feature selection aims to eliminate the irrelevant features by retaining the ones with minimum redundancy and maximum relevance (mRMR) in the dataset. It also reduces the complexity and time considerably which results in better training and testing thereby creating better optimal models. Feature selection also helps to avoid overfitting and this enhances the performance. Feature selection methods are mainly classified into: Filter and Wrapper methods based on how they interact with the classifier. Filter methods are generic and pick the intrinsic property of the features instead of going by the performance. They measure the relevance of features by their correlation with the dependent variables. However, the selection of features and the classification process happen independently so the filter methods are not very efficient. Also, they follow the local search in a small search space which is overcome by the wrapper method which collaborates both local and global optimum. Wrapper methods follow a greedy search approach by evaluating all possible subsets of the features. These methods are based on specific machine learning algorithms as evaluators that work hand in hand with the feature selection process.

2.5 Data Classification

k-NN is the most straightforward classification algorithm. k in k-NN is the number of nearest neighbors. k-NN uses the majority vote from k neighboring points to determine the class of unlabeled data. In general, k-NN is very effective in categorizing the data points that are nearer to the unlabeled point and assigns it to the smallest expected misclassification cost. k is the hyperparameter that needs to be tuned at the time of model prediction. There is no fixed value of k, it varies according to the dataset. The power traces matrix of 1500 traces (both trojan-free and trojan-infected) with 10000 sampling points is considered. The number of training and testing traces are 1050 and 450 respectively. The learned model is created using the k-NN classifier and the testing accuracy and error rate for the predicted values are calculated. In this work, the k value is subjected to a range of values from 2 to 15 to examine the accuracy and error rate values. The value $k = 1$ is not considered since any new object will be simply classified to the class of the single nearest neighbor. Figure 3(a) & (b) shows the accuracy and error rate values for different k values. The value $k = 5$ is taken as the optimum value since it achieves better accuracy with a minimum error rate value. Hence $k = 5$ is chosen as the nearest neighbor for classification pre and post-feature selection evaluation.

Table 1 Analysis of accuracy values for feature selection techniques.

Technique	Accuracy %
BA	92.5
CS	92.7
SSA	92.3
JA	93.4
HHO	94.3
WOA	**96.3**

When the number of features increases then the number of traces or data also needs to be increased. This increase in dimension sometimes may lead to a condition known as overfitting or the curse of dimensionality. To encounter this problem the need for feature selection arises. The swarm-based optimization algorithms namely Bat (BA) [36], Cuckoo Search (CS) [37], Jaya (JA) [38], Harris Hawk Optimization (HHO) [39], and Salp Swarm Algorithm (SSA) [40] in the literature are used in various applications for attribute selection of high dimensional data. The AES-128 trust-hub benchmark circuit is considered here. The T100 trojan trigger is an always-on condition where the detection process would be quite simple and harder to differentiate the efficiency between the algorithms. Hence the T1000 trigger condition is chosen and evaluated with the different algorithms. Table 1 gives an analysis of the accuracy values achieved by using the mentioned algorithms. By studying some of the feature selection methods for these specific

power consumption traces, we have streamlined the Whale Optimization Algorithm (WOA) [41] owing to better performance. Hence WOA is effectively used for feature selection in this proposed work.

2.6 Whale Optimization Algorithm

The WOA is a meta-heuristics optimization algorithm inspired by the scavenging behavior of humpback whales. The humpback whales are involved in the process of encircling the prey, bubble-net feeding, and searching for the prey.

2.6.1 Encircling the Prey

Humpback whales can first detect the location of their prey and encircle them. The humpback whales can premeditate the current best location in the search space since the optimum solution is not known prior. It assumes the target prey to be the best solution or is close to the optimum. After the best search position of the whale is defined, the other whales will update their position toward the best search. This situation is mathematically modeled as

$$\overrightarrow{D} = |\overrightarrow{C}.\overrightarrow{X}_{best}(t) - \overrightarrow{X}(t)| \tag{4}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X}_{best}(t) - \overrightarrow{A}.\overrightarrow{D} \tag{5}$$

where t indicates the current iteration, $\overrightarrow{X}_{best}(t)$ is the position of the best solution, $\overrightarrow{X}(t)$ is the current position of the whale, $\overrightarrow{A}$ and $\overrightarrow{C}$ are coefficient vectors and $(.)$ represents dot product multiplication. $\overrightarrow{X}_{best}(t)$ is updated in each iteration once a better solution is encountered. The vectors $\overrightarrow{A}$ and $\overrightarrow{C}$ can be calculated as:

$$\overrightarrow{A} = 2\overrightarrow{a}.\overrightarrow{r} - \overrightarrow{a} \tag{6}$$

$$\overrightarrow{C} = 2.\overrightarrow{r} \tag{7}$$

$$\overrightarrow{a} = 2 - t * (2/T) \tag{8}$$

where $\overrightarrow{r}$ is a random vector in the interval $[0, 1]$, t is the current iteration, T the maximum number of iterations and $\overrightarrow{a}$ is linearly decremented from 2 to 0 during the iteration in both the exploration and exploitation phases.

2.6.2 Bubble-Net Attacking Method- (Exploitation Phase)

Two strategies are modeled to attack the prey using the bubble-net method.

The shrinking encircling strategy behavior is achieved by reducing the value of $\overrightarrow{a}$ in Eq. (8). This in turn decreases the fluctuation range of $\overrightarrow{A}$. By varying $\overrightarrow{A}$ value randomly in the interval $[-1, 1]$ the new position of the whale can be defined anywhere between the original position and the best candidate position. The spiral updating position equation is modeled by the position of whale and prey to imitate the helix-shaped movement as

$$\overrightarrow{X}(t+1) = \overrightarrow{D}.e^{bl}.cos(2\pi l) + \overrightarrow{X}_{best}(t) \tag{9}$$

$$\overrightarrow{D} = |\overrightarrow{X}_{best}(t) - \overrightarrow{X}(t)| \tag{10}$$

where $\overrightarrow{D}$ is the distance between whale and prey during the i^{th} iteration(the best solution so far), b is a constant defining the shape of the spiral set as 1, l is a random number in the interval $[-1, 1]$.

2.6.3 Search for the Prey- (Exploration Phase)

The search for the prey randomly with the same strategy based on the variation of $\overrightarrow{A}$ is the exploration phase. The humpback whales perform a random search based on the position of each other. This search considers values of $|A| > 1$ instead of restricting to the reference agent and performs a global search. The mathematics is modeled as follows:

$$\overrightarrow{D} = |\overrightarrow{C}.\overrightarrow{X}_{rand}(t) - \overrightarrow{X}| \tag{11}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X}_{rand}(t) - \overrightarrow{A}.\overrightarrow{D} \tag{12}$$

where $\overrightarrow{X}_{rand}(t)$ is the random position vector of the whale from the current population. At each iteration in the WOA algorithm, the position of the whale is updated either by a random search or by the best solution obtained. The random search is performed when $|\overrightarrow{A}| > 1$(exploration) and the best solution is chosen when $|\overrightarrow{A}| < 1$(exploitation) for updating the position of the whale.

2.7 Modified Binary WOA Algorithm for Hardware Trojan Detection

The pseudocode of the modified Binary Whale Optimization Algorithm (BWOA) is drafted in Algorithm 1. A reference matrix (X_{ij}) with random values (x_{ij}) is initialized with the number of rows equal to N (the number of solutions) and the number of columns equal to the total number of features in the power traces dataset (PT_{rj}). Based on the threshold value of 0.5 random values exceeding 0.5 is set to 1 and those values below 0.5 to 0. This binary conversion is performed to selectively choose the best features. Those features are run through the k-NN classifier to ensure the relevance between the selected features and calculate the error rate. The initial fitness value (FF_{inf}) is set to an infinitely large value. The fitness value for each iteration from 1 to N is calculated and the solution corresponding to the least fitness is moved to $\overrightarrow{X}_{best}(t)$. The best set of features is identified by calculating the fitness value for each solution from 1 to N. Then, the algorithm finds the best position vector $\overrightarrow{X}_{best}(t)$ in T iterations. The proposed method targets to arrive at the least fitness value. Here the purpose of feature selection is multiobjective. It enhances the accuracy and also minimizes the number of features. This fitness function is formulated for maximum accuracy (minimum error rate) and also for a minimal number of selected features. These two objectives are combined in the fitness function:

$$FF = \alpha Error + \beta \frac{|S|}{|F|} \tag{13}$$

Algorithm 1 Pseudocode of the modified WOA algorithm for hardware trojan detection

Inputs: Number of solutions N, maximum number of iterations T, Power traces matrix PT_{rj}(Traces × Features), k value in k-NN

Initialize a reference matrix X_{ij}(i=1,2,...N) with random values $(min < x_{ij} < max)$
Binary conversion of continuous values based on threshold $(min + max)/2$
for (i = 1 to N) **do**
    Select features in PT_{rj} where $x_{ij} = 1$
    Perform k-NN classification of the modified power traces matrix PT_{mod} with selected features
    Evaluate the Fitness value and Error rate for individual solution using Eqns.(13)and(14)
    Set FF_{inf} to a infinitely large value
    if $(FF_i < FF_{inf})$ **then**
        $\overrightarrow{X}_{best}(t) = \overrightarrow{X}_i(t)$
        $FF_{inf} = FF_i$
    end if
end for
while $(t < T)$ **do**
    Define a using Eq.(8)
  // Shrinking encircling mechanism phase //
    Update the coefficients A and C using Eqns.(6)and(7) respectively
    Initialize p a random number in the interval [0,1]
    if $(p < 0.5)$ **then**
        if $(|\overrightarrow{A}| < 1)$ **then** [Encircling prey phase]
            Update the continuous position values using Eq.(5)
        else if $(|\overrightarrow{A}| \geq 1)$ **then** [Searching prey phase]
            Select a random whale $\overrightarrow{X}_{rand}(t)$
            Update the continuous position values using Eq.(12)
        end if
    else if $(p \geq 0.5)$ **then**[Spiral updating phase]
        Update the continuous position values using Eq.(9)
    end if
    Perform binary conversion of $\overrightarrow{X}_{best}(t)$ solution
    Repeat steps 3 to 12
end while
  // Generate best feature subset //

Outputs: Accuracy of validated model, the indexes, and number of selected features

$$Error = 1 - Accuracy \tag{14}$$

The model plot of the fitness function for AES-T1000 benchmark circuit evaluated using the binaryWOA (BWOA) feature selection technique is depicted in Fig. 4. The error is computed by the classification algorithm. α is the control parameter, $|S|$ denotes the length of the reduced feature subset, and $|F|$ denotes the maximum or total number of features. α and β are the weights for accuracy and feature reduction respectively, where $\beta = (1 - \alpha)$. Based on literature [40], [42] various experiments conducted on datasets imply that accuracy rates increase with an increase in α. The value of α is set to 0.99 and β will be 0.01 since the focus is on improving accuracy and minimizing the number of features. In this study, the k-NN algorithm works as an evaluator or wrapper with the feature selection algorithm with Euclidean distance and $k = 5$.

The swarm-based WOA [41] is well known for its advantages over evolution-based algorithms. The swarm-based algorithms are way ahead in preserving the search space information over subsequent iterations instead the evolutionary algorithms discard them when a new population is formed. The WOA is advantageous in its good exploitation capability. The balance between the exploration and exploitation phases helps to avoid the local optima and approach the global optimum. In this way, the WOA best suits our problem to find the best position and thereby extract the best features. The WOA algorithm performs the search for the best position value $\overrightarrow{X}_{best}(t)$ for 1 to T iterations. The binary conversion for $\overrightarrow{X}_{best}(t)$ is performed and steps 3 to 12 in Algorithm 1 are repeated to arrive at the best fitness value. The number of solutions and the number of iterations for this feature selection is chosen as $N = 10$ and $T = 100$ [40], [42]. Some experiments conducted in literature with these N and T values on more sensitive datasets have managed to show better results. Also, it is seen from Fig. 4 that the algorithm converges above 60 iterations. Considering all the other benchmark circuits and their convergence rate these values are chosen.

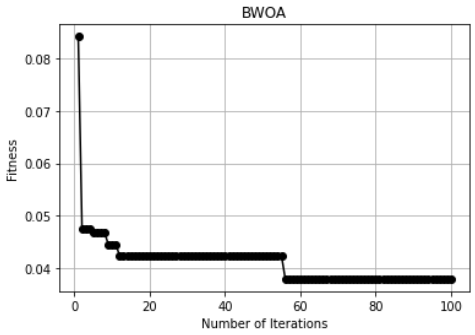


Fig. 4 Plot of fitness function for AES-T1000 circuit.

3. Experiment and Performance Results

3.1 Experimental Setup

In this proposed work, the Advanced Encryption Standard (AES-128) trust-hub benchmark circuit is evaluated. The power traces acquisition experimental setup is depicted in Fig. 5. The SAKURA-G FPGA board is used in this experiment for implementation and verification. The bitstream file is transferred to the FPGA and the power traces are acquired using a MSOX3104T oscilloscope at 1 GHz frequency and sampling frequency of 5G samples/sec. Figure 6 is a sample power trace of the AES-T1100 benchmark circuit.

3.2 Performance Analysis

To analyze the efficacy of our proposed approach, the different types of trigger conditions are inserted in the AES-128 golden circuit and tested independently. The effectiveness of the proposed model is tested initially by performing a classification with k-NN classifier prior to feature selection. Except T100 others are internally triggered Trojans. T100 belongs to the *always-on* type activation mechanism. Table 2

Table 2 Classification with k-nearest neighbor classifier.

HT Type	Trigger Activation Type	Accuracy %	Recall %	Precision %	F1-Score	AUC
T100	Always on	99	98.1	100	0.99	0.99
T1000	Trig., Internally by a predefined plain text	86	73.9	99.1	0.85	0.87
T1100	Trig., Internally by a sequence of predefined plain text	84.3	71.7	100	0.84	0.86
T1200	Trig., Internally based on number of encryptions	86.7	76.5	99.2	0.86	0.88
T700	Trig., Internally by a predefined input value	86	83.4	89.1	0.86	0.86
T800	Trig., Internally by a sequence of predefined input values	93	96.2	90.9	0.94	0.93
T900	Trig., Internally based on number of encryptions	79	71.3	86.2	0.78	0.79

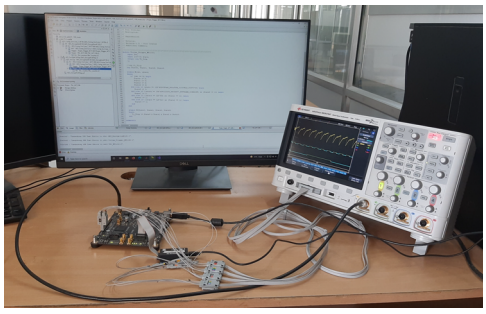


Fig. 5 Data acquisition experimental platform.

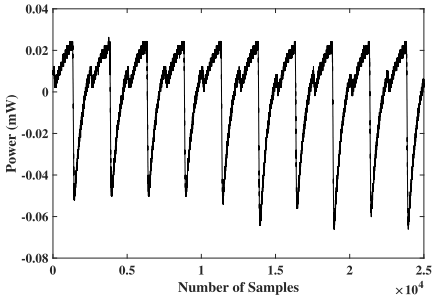


Fig. 6 Sample power trace of AES-T1100.

gives information about various evaluation metrics and trigger conditions when different types of trojans are inserted and classified using k-NN. The payload is to leak the key information through a secret covert channel which remains the same for all the circuits.

The classification metrics play a major role to evaluate the predictions of the model. Accuracy is one of the simplest yet universal metrics to measure the strength of the model. It is the number of correct predictions to the total number of predictions made.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{15}$$

where true positive (TP) is the number of HT samples predicted as HT correctly, false positive (FP) is the number of HT-free samples predicted as HT samples wrongly, false negative (FN) is the number of HT samples predicted as HT-free samples wrongly and true negative (TN) is the number of HT-free samples predicted correctly as HT-free itself. The recall is a measure of our model correctly identifying the true positives. It also refers to how accurately our model can identify the problem of concern. Precision is the ratio between the true positives and all the positives. Precision is a measure concerned only with the relevant data points.

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

Often there may be situations where, accuracy might be very high but precision or recall might be low which must be avoided. Ideally, the aim is to detect the maximum number of trojans. Improving recall will indirectly decrease precision. So the focus is to have a trade-off between recall and precision. In such a situation it gets meaningful to work with a single metric the F-measure also referred to as the F1-score or f-score. F-score is the harmonic mean of precision and recall.

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{18}$$

F1-score approaching one refers to a better value and nearing 0 refers to a worse value. The Area under the ROC curve (AUC) is another performance metric suitable for binary classification problems. The receiver operating characteristic (ROC) curve is a plot between the false positive rate (fpr) and the true positive rate (tpr). TPR refers to how good the model is at predicting the true class when the outcome is actually positive. FPR is the opposite which refers to how often an outcome is predicted as true when the actual outcome is false. The AUC is an approximate integral under the ROC curve:

$$AUC = \int_0^1 ROC(x)dx \tag{19}$$

The range of AUC varies from 0 to 1 and the value nearing one is focused on a better model. In this proposed work, various power consumption related trust-hub benchmark circuits have been analyzed. The power consumption traces can be measured under three occurrences: disabled HT, enabled HT, and triggered HT. The disabled HT would be a trojan-free AES-128 circuit. The enabled HT condition is when there are some extra malicious circuitry but not sure if the trojan would be triggered or not. The triggered HT condition is when the trojan is triggered by a particular input or physical condition. In the practical scenario, mostly the trojan is only triggered by some rare activation condition. In literature, some of the power consumption related works [23], [24], [27], [29], [30], [32]–[34] have failed to

Table 3 Improved performance metrics of the BWOA feature selection method.

HT Type	Accuracy %	Recall %	Precision %	F1-score	AUC	Reduced feature subset
T100	100	100	100	1	1	1
T1000	96.3	92.8	99.8	0.96	0.96	207
T1100	94.4	89.4	99.3	0.94	0.94	175
T1200	95.6	92.9	98.1	0.96	0.96	488
T700	93.1	94.1	92.3	0.93	0.93	457
T800	98.4	98.2	98.6	0.98	0.98	665
T900	85.7	82.6	88	0.85	0.86	577

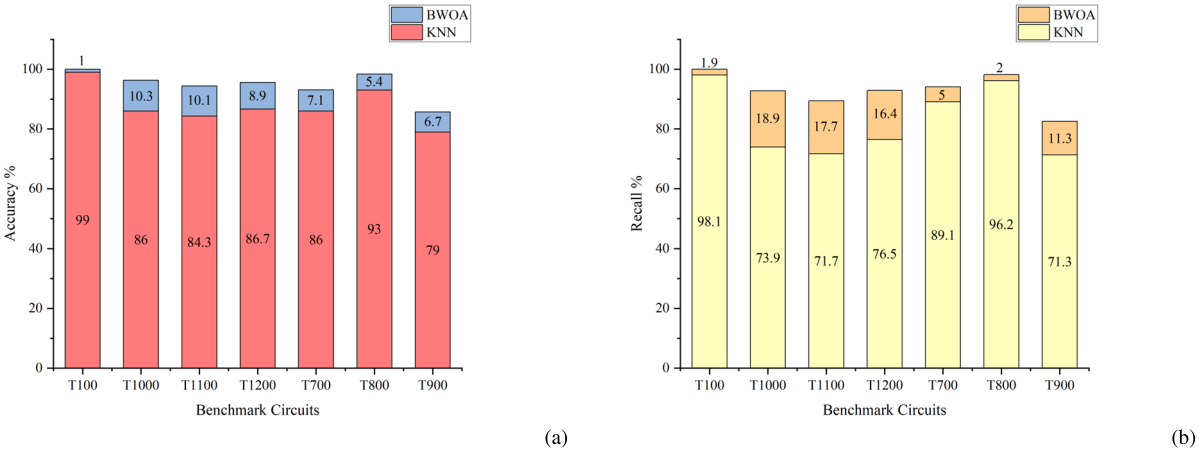


Fig. 7 Improvement in (a) accuracy and (b) recall by the BWOA technique.

mention this triggering schema. The information on which scenario the power traces are acquired is essential to evaluate the model's efficiency in a better way. However, in this proposed work, it is made sure the trojan is triggered by a particular trigger condition by creating a practical environment to attain an HT detection measure.

From Tables 2 and 3 it is evident that there is a steady improvement in the evaluation metrics. By taking a close observation it may be noted that there is a small fall in the precision values in T1100 and T1200 circuits after applying the feature selection technique. But the feature selection method (BWOA) plays a major role to improve the accuracy and also builds a trade-off between the recall and precision parameters, which in turn decreases the number of false negatives. This balancing property of the feature selection technique shows a maximum of 10.3% improvement in accuracy and 18.9% improvement in the recall. Figure 7(a) & (b) depicts the importance of the BWOA feature selection technique by showing the increase in accuracy and recall values by comparing it with the k-NN classifier. The ROC curve for the various benchmark circuits is shown in Fig. 8. It is evident that the area approaches the ideal measure 1 for all the benchmark circuits considered in this work. The feature selection technique plays a major role in improving the evaluation metrics and greatly contributes towards minimizing the number of features. A 10-fold validation is performed for all the benchmark circuits and the average of the parameters are considered. The power consumption traces are subjected to a maximum reduction of redundant features by the BWOA method up to a minimum of a single feature for the T100

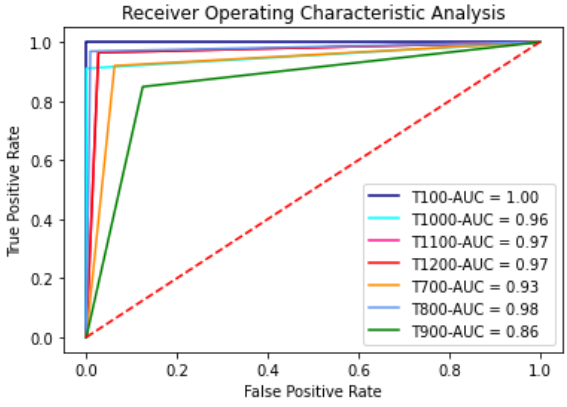


Fig. 8 ROC curve for the benchmarks.

benchmark circuit. The least number of features obtained from the 10000 feature vectors while performing the 10-fold validation in the proposed method is evident from Table 3 for each benchmark circuit.

Table 4 gives a brief of the various existing power profile-related HT detection methods. In [23] the trojan trigger inserted is not evident and only a single measurement is used for testing which will not be a robust way to classify the trojan in the AES circuit. The PCA technique used in the works [23], [24], [27], [29], [34] does not ensure the best feature selection from the dataset but instead aims at dimensionality reduction. The principal components being the linear combination of the feature vectors are less interpreted than the original feature vectors. Also, it tends to lose the critical characteristics of the HT. In [24] increas-

Table 4 Various power analysis-based trojan detection methods.

Article	Learning Model	Original Dimension	Preprocessing Technique	Reduced Feature Dimension	Circuit	Implementation Platform	Evalution Metrics
[23]	LDA	256	PCA	-	AES-128	Spartan-6 (xc6slv45)	Acc=100%
[24]	Isolation Forest	20k×10k	PCA	2	AES-T100	Sakura-G	Acc=94%
[25]	SVM	2k×5k	-	5000	AES-T100	Sakura-G	Acc=93.7%
[27]	Softmax Regression	1837 power traces	PCA	-	AES-128	Sakura-G	Acc=97%
[28]	k-NN,DT, NBC,DL	-	-	-	MC8051	Spartan-6 (xc6slv45)	Acc=99.02%
[29]	Mahalanobis distance	10000	PCA	2050	AES	Cyclone IV	Acc=100%
[30]	BPNN	1000×1330	-	1330	AES-T100	Sakura-G	Acc=100%
[31]	BPNN	1200 sample points	-	24	AES	KC750	-
[32]	BPNN	12000 sample points	Wavelet Transform	2000	AES	Kintex 7	Sen=99.2%
[33]	ELM	1000×2000	-	2000	-	Cyclone IV	Acc=100%
[34]	NN	8000	PCA+PSO	17	mini AES-8	-	Acc=99.1%
Proposed	**k-NN**	**1500×10k**	**BWOA**	**1**	**AES-T100**	**Sakura-G**	**Acc=100%**

Acc is Accuracy
Sen is Sensitivity

ing the number of iTrees and power traces for accuracy will rather increase the complexity and processing time of the whole model. The isolation forest [24] and SVM [25] based HT detection methods have employed 10000 and 1000 traces separately for each group. However, in this proposed method, the number of traces is 750 for each group and the BWOA feature selection technique has made a vast minimization in the number of features to generate a better accuracy.

In [26] the model is trained using both the benchmark circuits' trojan and trojan-free power traces; except for a single circuit used for testing which follows anomaly detection. Since data on the number of sample points (features) and traces considered for classification are not mentioned, the complexity of the technique could not be compared. In [27] the ML model's efficiency needs to experiment with the different trojan-inserted benchmark triggers which are not evident in their work. The ROC curve is a plot used to measure the efficiency of classification problems and will not in any way represent the ability to detect HT. The work [28] is an HT detection technique for a micro-controller and [29] uses Mahalanobis distance for the detection technique. The circuit and detection base varies in these works [28], [29] compared to the proposed method. The works [30]–[34] methods are neural network-based detection techniques. Even though the AES-T100 trigger is common with the proposed work, a BPNN technique is utilized in [30] and the proposed work uses the k-NN classifier. A classification algorithm combined with a feature selection technique should be fast enough to determine results. k-NN is advantageous in producing results faster at a low cost in minimum processing time [43]. The HTs are self-designed in which three trojans are triggered in a sequence with different functionality [31] and secret key leakage type of trojan [32] are implemented in the AES circuit. The technique [33] uses a self-designed circuit and three types of HTs to perform the power analysis. The work [34] is one similar technique to the proposed work which has involved the PSO algorithm for feature selection but, only a sub-module of the AES circuit is considered whose complexity is less.

Table 5 Comparison with existing power-related techniques.

Article	Dataset Dimension		Accuracy %
	Training	Testing	
[24]	400×2	100×2	AES-T100=94
			T1000=94
			T1100=93.9
			T1200=93.8
[25]	1200×5k	400×5k	AES-T100=93.7
[30]	800×1330	200×1330	AES-T100=100
Proposed	1050×1	450×1	AES-T100=**100**
		450×207	T1000=**96.3**
		450×175	T1100=**94.4**
		450×488	T1200=**95.6**

Even though there are several works in the literature with power profile-related HT detection techniques the works [24], [25] and [30] are works that have employed similar benchmark trojan trigger circuits. The work [24] employs an unsupervised technique but this is the only existing technique related to power analysis with the similar T100, T1000, T1100, and T1200 triggers. The works [25] (SVM) and [30] (BPNN) have considered the T100 trigger condition in common with the proposed technique. The techniques with common circuit and trigger conditions with the proposed method are listed in Table 5. It is to be noted that the proposed method has outperformed the existing [24], [25] ML techniques in accuracy by considering similar trigger conditions. The proposed technique has also achieved 100% accuracy comparable with the BPNN technique [30] but with single feature reduction. The proposed technique is efficient in scaling and independent with respect to the circuit and power values, thus the testing dataset dimension varies for each trigger condition as seen in Table 5.

Table 6 Classification metrics of ISCAS'89 circuits with k-NN and BWOA.

Circuit	k-NN			BWOA			
	Accuracy %	Recall %	Precision %	Accuracy %	Recall %	Precision %	Reduced feature subset
s298	89.3	89.8	88.5	90.7	93.3	88.6	197
s526	72.5	57.5	80.4	77.5	64.7	88.7	285

4. Experiment on ISCAS'89 Circuits

The proposed technique is further investigated using the ISCAS'89 benchmark circuits to test the effectiveness of the feature selection technique. The trojan is inserted in the circuit, where the trigger is a rare instance and the payload will produce a change in the output data. The payload part involves some circuitry that includes a reasonable overhead of 5% of the total number of gates in the original circuit. Here 500 power traces (each trojan-free and trojan-infected) with 5000 sampling points are collected to form the classification dataset. The BWOA technique is further applied to improve classification efficiency. The number of solutions and the number of iterations for this specific power traces dataset is chosen as N=10 and T=100. The algorithm was found to converge earlier before 50 iterations. Hence the number of iterations is fixed as T=50.

Table 6 lists the evaluation metrics of the s298 and s526 benchmark circuits with k-NN classifier and the proposed BWOA feature selection technique. The percentage of accuracy depends on the effect of the payload and the number of traces acquired. As observed in the AES circuits, there is a steady improvement in the recall parameter which reduces the number of false negatives. A maximum of 7.2% and 8.3% increase is observed in the recall and precision values of the s526 circuit. However, by slightly increasing the number of power traces acquired the classification accuracy can be still improved.

5. Conclusion

In this paper, we propose a novel technique for HT detection employing the modified BWOA for feature selection. The continuous optimization solutions are subjected to binary conversion to apply for feature selection. The feature selection method plays a vital role in improving the accuracy and recall measures and also reduces the number of features considerably. The proposed technique also brings a trade-off between recall and precision values which in turn decreases the number of false negatives. A maximum of 10.3% and 18.9% improvement in accuracy and recall measures respectively is observed by applying the proposed method. Also, a noticeable reduction in the number of features up to a single feature with a minimal number of traces is achieved.

The proposed technique is unique in combining the modified optimization technique with ML and making its best use for HT detection. This technique has varied applications and some of the areas include defense and healthcare. The future scope lies in analyzing this technique for different vulnerable non-cryptographic circuits. Also, we aim to investigate how the proposed method changes to different parameter variations of the feature selection algorithm.

References

[1] R.S. Chakraborty, S. Narasimhan, and S. Bhunia,"Hardware Trojan: Threats and emerging solutions," IEEE International High Level Design Validation and Test Workshop, pp.166–171, 2009.
[2] F. Courbon, P. Loubet-Moundi, J.J.A. Fournier, and A. Tria, "A high efficiency hardware trojan detection technique based on fast SEM imaging," IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), pp.788–793, 2015.
[3] T. Sugawara, D. Suzuki, R. Fujii, S. Tawa, R. Hori, M. Shiozaki, and T. Fujino, "Reversing stealthy dopant-level circuits," International Workshop on Cryptographic Hardware and Embedded Systems, pp.112–126, 2014.
[4] E. Jedari and R. Rashidzadeh, "A hardware Trojan detection method for IoT sensors using side-channel activity magnifier," IEEE Internet Things J., vol.9, no.6, pp.4507–4517, 2021.
[5] S. Narasimhan, D. Du, R.S. Chakraborty, S. Paul, F.G. Wolff, C.A. Papachristou, K. Roy, and S. Bhunia, "Hardware Trojan detection by multiple-parameter side-channel analysis," IEEE Trans. Comput., vol.62, no.11, pp.2183–2195, 2012.
[6] H. Salmani, M. Tehranipoor, and R. Karri, "On design vulnerability analysis and trust benchmark development," IEEE Int. Conference on Computer Design (ICCD), pp.471–474, 2013.
[7] Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," IEEE International Workshop on Hardware-Oriented Security and Trust, pp.51–57, 2008.
[8] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," IEEE Symposium on Security and Privacy (SP'07), pp.296–310, 2007.
[9] N. Karimian, F. Tehranipoor, M.T. Rahman, S. Kelly, and D. Forte, "Genetic algorithm for hardware Trojan detection with ring oscillator network (RON)," IEEE International Symposium on Technologies for Homeland Security (HST), pp.1–6, 2015.
[10] H.S. Choo, C.Y. Ooi, M. Inoue, N. Ismail, M. Moghbel, S.B. Dass, C.H. Kok, and F.A. Hussin, "Machine-learning-based multiple abstraction-level detection of hardware Trojan inserted at register-transfer level," Proc. IEEE 28th Asian Test Symposium (ATS), pp.98–980, 2019.
[11] M. Oya, Y. Shi, M. Yanagisawa, and N. Togawa, "A score-based classification method for identifying hardware-Trojans at gate-level netlists," Proc. Design Automation & Test in Europe Conference & Exhibition (DATE), pp.465–470, 2015.
[12] K. Hasegawa, M. Oya, M. Yanagisawa, and N. Togawa, "Hardware Trojans classification for gate-level netlists based on machine learning," Proc. IEEE Symposium on On-Line Testing and Robust System Design (IOLTS), pp.203–206, 2016.
[13] K. Hasegawa, Y. Shi, and N. Togawa, "Hardware Trojan detection utilizing machine learning approaches," Proc. 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications, pp.1891–1896, 2018.
[14] T. Han, Y. Wang, and P. Liu, "Hardware Trojans detection at register transfer level based on machine learning," Proc. IEEE International Symposium on Circuits and Systems (ISCAS), pp.1–5, 2019.
[15] C. Wang, J. Li, M. Yu, and J. Wang, "An intelligent classification

method for Trojan detection based on side-channel analysis," IEICE Electron. Express, vol.10, no.17, p.20130602, 2013.
[16] K. Hasegawa, M. Yanagisawa, and N. Togawa, "Trojan-net feature extraction and its application to hardware-trojan detection for gate-level netlists using random forest," IEICE Trans. Fundamentals, vol.E100-A, no.12, pp.2857–2868, Dec. 2017.
[17] K. Hasegawa, M. Yanagisawa, and N. Togawa, "A hardware-Trojan classification method using machine learning at gate-level netlists based on Trojan features," IEICE Trans. Fundamentals, vol.E100-A, no.7, pp.1427–1438, July 2017.
[18] K.G. Liakos, G.K. Georgakilas, and F.C. Plessas, "Hardware Trojan classification at gate-level netlists based on area and power machine learning analysis," IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp.412–417, 2021.
[19] M. Priyadharshini and P. Saravanan, "An efficient hardware Trojan detection approach adopting testability based features," IEEE International Test Conference India, pp.1–5, 2020.
[20] X. Xie, Y. Sun, H. Chen, and Y. Ding, "Hardware Trojans classification based on controllability and observability in gate-level netlist," IEICE Electron. Express, vol.14, no.18, p.20170682, 2017.
[21] C.H. Kok, C.Y. Ooi, M. Moghbel, N. Ismail, H.S. Choo, and M. Inoue, "Classification of Trojan nets based on SCOAP values using supervised learning," Proc. IEEE International Symposium on Circuits and Systems (ISCAS), pp.1–5, 2019.
[22] C.H. Kok, C.Y. Ooi, M. Inoue, M. Moghbel, S.B. Dass, H.S. Choo, N. Ismail, and F.A. Hussin, "Net classification based on testability and netlist structural features for hardware Trojan detection," Proc. IEEE 28th Asian Test Symposium (ATS), pp.105–1055, IEEE, 2019.
[23] R. Shende and D.D. Ambawade, "A side channel based power analysis technique for hardware trojan detection using statistical learning approach," IEEE Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN), pp.1–4, 2016.
[24] T. Hu, L. Wu, X. Zhang, and Z. Liao, "Hardware Trojan detection combines with machine learning: An isolation forest-based detection method," IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE), pp.96–103, 2020.
[25] T. Hu, L. Wu, X. Zhang, Y. Yin, and Y. Yang, "Hardware trojan detection combine with machine learning: an svm-based detection approach," IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID), pp.202–206, 2019.
[26] S. Faezi, R. Yasaei, A. Barua, and M.A. Al Faruque, "Brain-inspired golden chip free hardware trojan detection," IEEE Trans. Inf. Forensics Security, vol.16, pp.2697–2708, 2021.
[27] V.P. Hoang, "Hardware Trojan detection based on side-channel analysis using power traces and machine learning," International Conference on Research in Intelligent and Computing, pp.53–56, 2021.
[28] F.K. Lodhi, S.R. Hasan, O. Hasan, and F. Awwadl, "Power profiling of microcontroller's instruction set for runtime hardware Trojans detection without golden circuit models," Design, Automation and Test in Europe Conference and Exhibition (DATE), pp.294–297, 2017.
[29] Q. Cui, K. Sun, S. Wang, L. Zhang, and D. Li, "Hardware trojan detection based on cluster analysis of mahalanobis distance," International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), vol.1, pp.234–238, 2016.
[30] L. Xu, J. Li, L. Dai, and N. Yu, "Hardware Trojans detection based on BP neural network," IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA), pp.149–150, 2020.
[31] J. Li, L. Ni, J. Chen, and E. Zhou, "A novel hardware Trojan detection based on BP neural network," International Conference on Computer and Communications (ICCC), pp.2790–2794, 2016.
[32] L. Ni, J. Li, S. Lin, and D. Xin, "A method of noise optimization for hardware Trojans detection based on BP neural network," IEEE International Conference on Computer and Communications (ICCC), pp.2800–2804, 2016.
[33] S. Wang, X. Dong, K. Sun, Q. Cui, D. Li, and C. He, "Hardware Trojan detection based on ELM neural network," International Conference on Computer Communication and the Internet (ICCCI), pp.400–403, 2016.
[34] C.X. Wang, S.Y. Zhao, X.S. Wang, M. Luo, and M. Yang, "A neural network trojan detection method based on particle swarm optimization," International Conference on Solid-State and Integrated Circuit Technology (ICSICT), pp.1–3, 2018.
[35] S. Yang, T. Hoque, P. Chakraborty, and S. Bhunia, "Golden-free hardware trojan detection using self-referencing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.30, no.3, pp.325–338, 2022.
[36] X.S. Yang, "Bat algorithm for multi-objective optimisation," International Journal of Bio-Inspired Computation, vol.3, no.5, pp.267–274, 2011.
[37] X.-S. Yang, and S. Deb, "Engineering optimisation by cuckoo search," International Journal of Mathematical Modelling and Numerical Optimisation, vol.1, no.4, pp.330–343, 2010.
[38] R.V. Rao and A. Saroj, "A self-adaptive multi-population based Jaya algorithm for engineering optimization," Swarm and Evolutionary Computation, vol.37, pp.1–26, 2017.
[39] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," Future Generation Computer Systems, vol.97, pp.849–872, 2019.
[40] H. Faris, M.M. Mafarja, A.A. Heidari, I. Aljarah, A.M. Al-Zoubi, S. Mirjalili, and H. Fujita, "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems," Knowledge-Based Systems, vol.154, pp.43–67, 2018.
[41] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Advances in Engineering Software, vol.95, pp.51–67, 2016.
[42] M. Mafarja, I. Aljarah, and A.A. Heidari, "Binary dragonfly optimization for feature selection using time-varying transfer functions," Knowledge-Based Systems, vol.161, pp.185–204, 2018.
[43] R.S. Arslan and A.H. Yurttakal, "K-nearest neighbour classifier usage for permission based malware detection in android," ICONTECH INTERNATIONAL JOURNAL, vol.4, no.2, pp.15–27, 2020.

Priyadharshini Mohanraj received a B.Tech in Electronics and Communication Engineering in 2010 from Amrita University, Coimbatore, India, and M.E. degree in VLSI Design in 2019 from Government College of Technology, Coimbatore, India. She has around 2.5 years of industrial experience. She is now a research scholar at PSG College of Technology. Her research interests include hardware security and VLSI design.

Saravanan Paramasivam received a Ph.D. degree in hardware security from Anna University, Chennai, India, in 2015. He is currently an Associate Professor with the Department of Electronics and Communication Engineering, PSG College of Technology, Coimbatore, India. His research interests include hardware security, quantum computing, and multi-scale modeling of nanoelectronic devices. He has around five years of industrial experience. He is a member of the IETE, ISSS, IEEE, and VLSI Society of India.