Machine learning Based Hardware Trojans Detection in Integrated Circuits: A Systematic Review

Ritu Sharma and Prashant Ranjan

Ritu Sharma is with the University of Engineering & Management (UEM), Jaipur, India (phone: 9462836744; e-mail: reetusharma310@gmail.com). Prashant Ranjan is with the Department of Electronics & Communication Engineering, University of Engineering & Management (UEM), Jaipur, India. (e-mail: prashant.ranjan@uem.edu.in).

Abstract: A purposefully inserted additional circuit known as the Hardware Trojan (HT) is implanted inside original integrated circuits during the designing or manufacturing stages. It has the potential to manipulate circuit performance or acquire underlying information. Due to machine learning's (ML) exceptional results across a range of learning domains, the academic and business community are now looking at how Hardware Trojan (HT) attacks can be strengthened by employing conventional methods. Only a few survey studies have thoroughly evaluated the achievements and covered the unresolved issues in this subject. The literature for methods of defining HT concerns centered on machine learning is being reviewed in this research. Specifically, we first classify all known HT attacks and later analyze the evolution of the latest machine learning models in five separate areas of HT detection: reverse engineering, side-channel analysis, and golden model-free analysis, circuit feature analysis and classification approaches. Based on the review, we analyze the lessons learned and obstacles that have emerged from prior investigations. HT Defense Studies discusses the pros and cons of Supervised and unsupervised ML. Finally, a comparison of machine learning-based and non-machine learning-based HT detection approaches is shown and current challenges with future work are also suggested.

Keywords**:** Hardware Trojan detection, Golden model, Integrated circuits, Machine learning, Supervised ML, Unsupervised ML, Reverse Engineering, Side channel analysis.

1. Introduction

Over the past few decades, Trojan intrusions on Integrated Circuits have considerably increased. Various engineering designs, including those for automobiles, communications, shipping, military applications, transmission networks, and home appliances, are controlled and operated by ICs. Any additional circuitry connected to the primary device to interfere with its functioning is referred to as a "Hardware Trojan." Any integrated circuit's manufacturing process involves interactions with several outside parties and businesses at various stages, leaving the circuitry open to malicious attacks. Figure1 depicts the evolution phase of a modern integrated circuit along with its credibility at different stages [1]. The vulnerability of different stages is marked with different colors, i.e. red means highly vulnerable for HT attacks; green means least vulnerable and yellow denotes the vulnerability for any attack between the most vulnerable and least vulnerable levels.

The hardware's integrity and dependability have been compromised due to these extra circuits. Therefore, it is getting harder to research every defense strategy that could be used against these attacks. There are essentially five types of Trojan detection techniques. The five topics are side-channel analysis, circuit feature analysis, golden model free analysis, classification approaches and reverse engineering.

In a perfect world, pre-silicon inspection and post-silicon verification would be able to identify any unwanted changes performed to an IC. A golden model of the complete IC is necessary for pre-silicon inspection or simulation, though. This could not necessarily be the case, particularly for IP-based layouts where IPs may originate from outside sources. During the post-silicon stage, the circuit can be validated whether it's via damaging de-packaging and IC reverse engineering [2] or by contrasting its functionality or circuit features with a "golden" variant of the IC [3]–[5].

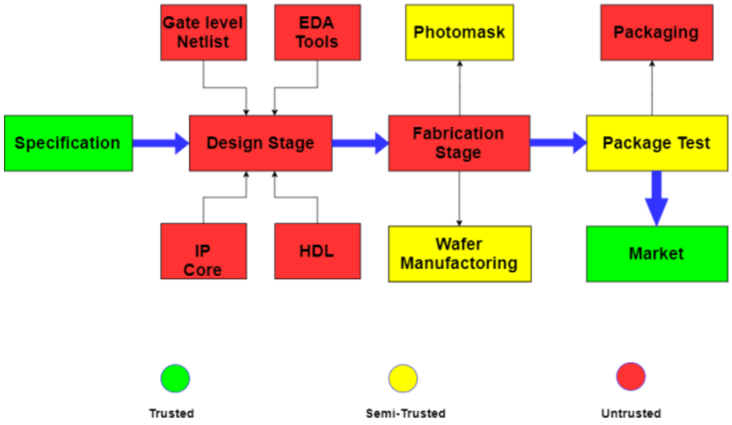

Figure 1 Potential stages of HT attack in a modern IC life cycle

Although recent work doesn't support destructive methods for HT detection in ICs and traditional logic testing method at post manufacturing stage is also not suitable to completely detect HTs.

Yin et al. [6] discussed the kind of threats of hardware Trojan at different layers of the chip, as shown in figure 2. The whole framework was formed according to the type of leakage information and type of damage at these layers. Device layer threat, system layer threat, data layer threat, and application layer threat are the four sections of the threat framework. Device layer threat is the case when the damage caused by the HTs is on the device. These malicious circuits physically damage the device by altering the chip parameters, resulting in hidden possibilities for HT information attacks.

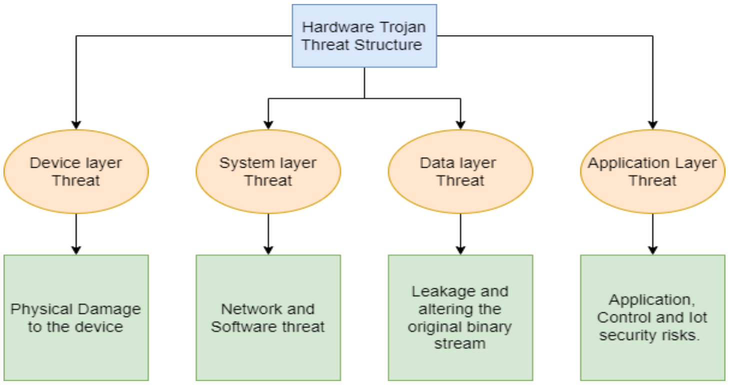
Hardware Trojan Threat Structure
Device layer Threat
System layer Threat
Data layer Threat
Application Layer Threat
Physical Damage to the device
Network and Software threat
Leakage and altering the original binary stream
Application, Control and Iot security risks.

Figure 2 HT threat structure in the four layers of an IC

System layer threats can be stated as system and network interference. Such threats are malicious code that directly attacks the normal software of the system and secretly receives all the essential data of the user. A data layer threat is an attack on the data or private information. The HTs alter the binary stream data of the original chip and leads in leaking and altering the private and vital data of the user. The application layer threat focuses on application, control, and IOT security risks. Such threats occur while interacting with the end users.

This paper's goal is to provide an insight of existing machine learning based techniques for HT detection and comparison in terms of precision, accuracy rate, TPR (True positive rate), TNR (True negative rate), SR (Success rate), FPR (False positive rate) and FNR (False negative rate) in order to choose the optimal ML model for the corresponding HT detection approach. Thorough surveys on the designs, ML model classifications, and measures for HT concerns are presented. Here is a summary of this article's significant contributions. This article carefully examines the most recent developments in the use of ML technologies for prevention and detection of HTs. We examine certain key elements of the use of ML algorithms in HT protection fields as well as possible issues with the ultra-modern and advanced techniques are discussed. The study finishes with a general evaluation, mentions of research gaps, and recommendations for improved HT detection, categorization, and prevention.

The rest of this research is divided into the following sections. In part 2, we give an overview of machine learning and its models as well as several kernel types that have been applied to HT detection so far. The thorough discussion of several ML-based HT detection techniques is covered in Section 3. The final results of the bibliographic references, along with the difficulties and opportunities facing the HT detection sector, are presented in section 4 through tables. The article is eventually concluded in section 5.

2. Overview of Machine Learning and Models

This section deals with the terminology, concept and models of machine learning that have been used to detect HT attacks.

2.1 Machine Learning Terms & Definitions

Data science field known as machine learning (ML) use statistically significant approaches to enhance effectiveness based on prior experience to uncover novel trends in large amounts of data. The use of self-improving algorithms is a crucial part of machine learning. Like other creatures, humans too gain knowledge from their previous experiences and flaws. Self-improving algorithms also form decisions like humans only. These previous experiences are considered as "training data' to perform any task. In machine learning, data is made up of instances that can be termed as variables or attributes. These variables can have nominal, ordinal, binary numbers or numeric values. The previous experiences train the learning algorithms and ML models and their performance for the tasks increases with experience over time. The following are the phases involved in applying ML algorithms in general: preprocessing phase, learning phase, evaluation phase and prediction phase [7]. The preprocessing procedure selects the suitable features first, and the data having these attributes is subsequently extracted out from raw data. These characteristics are eventually employed to distinguish between the various intended outcome values. To create learning models from the provided given dataset, the appropriate learning algorithms are found and executed throughout the learning phase. Following that, techniques like outcome assessment, cross-validation, and hyper-parameter enhancement are used to produce final models.

In the evaluation process, the completed models are tested against the testing dataset to gauge their effectiveness. To determine the anticipated values of the required outcomes for the recent input data, the finalized models are applied in the prediction phase. After following through these phases at the end, the model is set to predict future output values for the new data input. Figure 3 depicts the flowchart of ML.

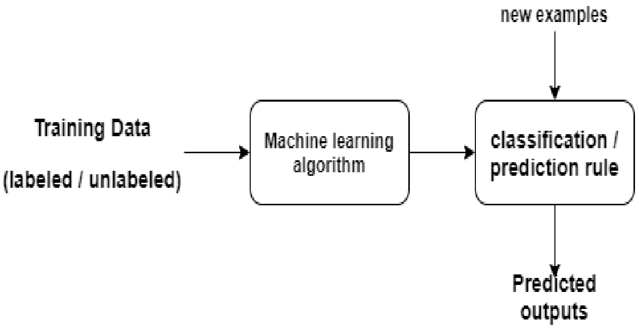

Figure 4 A typical ML workflow [3]

2.2 Machine Learning Algorithms

ML algorithms are basically divided into two different styles or classes of algorithms based on the utilization of previous experiences or knowledge over the input dataset. In supervised ML algorithms, each specimen in the training set is assigned a tag that serves as a marker for the category to which it belongs. The goal is to develop a generalized rule that accurately predicts the proper tag for each sample and can be applied to data outside the training set. In unsupervised ML algorithm there is no labeling of training set, instead, there is no difference between training and test set of data. Moreover, the algorithm itself modifies its processing as per the input data with the intention of uncovering some unseen patterns. Dimensionality Reduction (DR) approach, which is generally done before the learning process, is commonly used in both supervised and unsupervised learning to produce a more concise lower-dimensional representation.

2.3 ML Models

This segment presents an overview of different supervised and unsupervised ML models, dimensionality-reduction methodologies, and feature selection models used in hardware Trojan detection.

2.3.1 Supervised Machine learning

Support Vector machine (SVM): It is a supervised machine learning algorithm whose idea is to find an optimal solution by maximizing the gap between training data points of various target classes [8]. Drawing classification boundary lines are more accessible using SVM algorithms. When the data is not linearly separable, the SVM uses Kernel-trick, which transforms the data to high-dimension from low dimension where they can be separated [9]. The below fig.4 presented how SVM selects a hyperplane with maximum margin and separates two classes of data sets perfectly; hence, to achieve both characteristics, hyperplane A is best to choose.

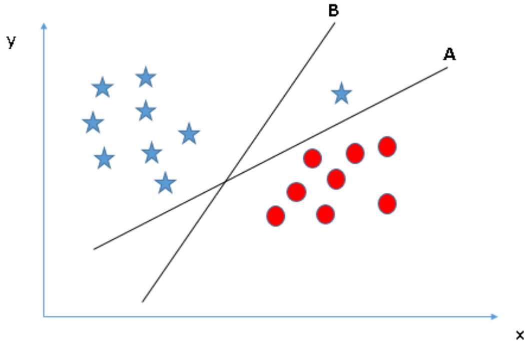

Figure 3 SVM hyperplane separating data points in a 2D plane

Artificial Neural network (ANN): ANNs are commonly known as Neural Network. They are basically work according to brain functionality like solving complex functions, audio and visual recognitions, decision making and pattern generation [10]. ANNs works basically by evaluating data in several layers. And process it by visualizing the objects through layers of neurons, inspired by humans. In most cases, ANN is depicted as a network of interconnected neurons that interact with one another. Each link has a numeric weight based on previous experience that can be changed. In a range of classification applications, such as HT detection, ANNs have proven to be useful.

Bayesian Models [BM]: are supervised learning algorithms that can be used to solve regression problems or classification. These are based on misjudgment losses and probability statistics [11].

One class SVM: Supervised model versions employed for determining if new training data belongs to a specific class. When data from the attacked systems is unknown before, they are commonly employed in hardware security [12]. The optimization goal in one-class SVM is to reduce the volume around a target, a hypersphere that encloses only one class's training data.

Linear Regression: One independent variable is used in linear regression to interpret the result of the dependent variable [13].

Decision Tree: It is a tree kind of learning model, which has its root node, several end or leaf nodes and intermediate nodes [14]. Internal nodes lead to attributes while the leaf node leads to final decision. Throughout the testing phase, every node decision is taken according to splitting algorithm and then data points traverse to the end nodes for the final decision. Decision trees tend to over-fit when a significant number of attributes are used.

K-Nearest neighbors: Measure the distance between distinct eigenvalues to classify the training dataset [15]. The fundamental notion is that if the maximum of the k most identical samples in the hyperplane refers to a specific category, then so does the sample, where k is generally a positive number under 20. Here identical sample refers to those data points which are in nearest neighbor in feature space. All objects that have been accurately categorized are considered selected neighbors in K-NN.

2.3.2 Un supervised Machine learning

Clustering Algorithms: The clustering algorithm calculates the distance between groupings and separates data points into several groups based on how close they are to one another [16]. It is different than classification. The unobtainable attributes of golden designs/ICs are one of the main reasons for choosing CAs in the field of HT detection, as CAs are unaffected to this condition.

K-means Clustering: It is a modified version of CAs which splits the data points further in groups [17]. Its objective is to reduce the intra-group interspace connecting the data sets in a similar group while gradually increasing the distance between groups between the datasets of distinct groups. A collection of n sample points is sub-divided into k number clusters with the intention of maximizing the similarity index with in a cluster as shown in figure 4.

Ordering points to identify the clustering structure: These are conventional CAs depending on density. They can identify higher-density points by estimating the density of the related nodes and various clusters are created by gradually connecting all the high-density points to form one block. By using this algorithm dataset of distinct shapes and sizes can be obtained.

Partitioning around Medoids (PAM): PAM is one of the clustering approach which is comparable to K-means clustering, except that in this real data points of dataset is considered as initial centroids which are the cluster's medoids, rather than the cluster's mean. In a nutshell, the PAM clustering algorithm outperforms K-means clustering in terms of noise resistance [8].

2.3.3 Dimensionality Reduction and Feature selection

Principal Component Analysis (PCA): It is a well-liked dimensionality reduction approach that maps n-dimensional characteristics to a k-dimensional space [18]. The mapped k-dimensional attributes are orthogonal which are also termed as principal components. Here features size of the k component is much smaller than the original n component size.

Genetic Algorithm (GA): [16] is a type of heuristic algorithm that is widely used. A GA can be used to locate a minimal subset of features from a collection of data that delivers the best classification accuracy in classification tasks.

2.4 Kernel Functions in ML

Kernels are a collection of distinct forms of pattern analysis algorithms. Using a linear classifier, they are employed to address a non-linear problem. SVM (Support Vector Machines), which are utilized in regression and classification concerns, utilize Kernel Methods. The SVM employs a technique known as the "Kernel Trick," in which the data is processed and an adequate boundary for the various outputs is determined. Following are the Kernel functions employed in ML.

- Kernel is frequently used in Machine Learning, referring to the kernel trick, which is a way of solving a non-linear issue with a linear classifier.
- The kernel function is used to translate the initial non-linear observations into a higher-dimensional space where they can be divided.
- The Kernel Trick enables us to work in the initial feature region without determining the data's coordinates in a higher dimensional space.

Various kinds of kernels being used SVM are shown below (Support Vector Machine).

a) Liner Kernel

If we have two vectors named A1 and B1, the linear kernel is defined by their dot product:

$$K\ (A1, B1) = A1.B1$$

b) Polynomial Kernel

The following equation describes a polynomial kernel:

$$K\ (A1, B1) = (A1.B1 + 1)d$$

Where, d is the polynomial's degree, while A1 and B1 are two vectors.

c) Gaussian Kernel

A radial basis function kernel is something like this.

The equation is as follows:

$$k\ (X_1, X_2) = \exp -\gamma\ \|X_1 - \ X_2\|^2$$

The provided sigma significantly impacts the Gaussian kernel's performance and should not be exaggerated or underrated; instead, this should be meticulously tailored to the task.

d) Laplacian Kernel

A Laplacian kernel is much less likely to alter and is entirely equal to the formerly stated exponential function kernel. The Laplacian kernel equation is as follows:

$$k\ (x, y) = exp\left(-\ \frac{\|x - y\|)}{\sigma}\right)$$

e) Hyperbolic or the Sigmoid Kernel

This kernel is utilized in the machine learning field of neural networks. The bipolar sigmoid function is used to activate the sigmoid kernel. The hyperbolic kernel function has the following equation:

$$k\ (x, y) = \ tanh\ \ \llbracket \alpha x^T y + c \rrbracket$$

a) Anova radial basis kernel

Like the Gaussian and Laplacian kernels, this kernel is designed to exhibit well enough in multidimensional regression situations. This is also referred to as a radial basis kernel.

The Anova kernel's equation is:

$$k(x, y) = \sum_{k=1}^{n} \exp(-\sigma\ (x^k - y^k)^2)^d$$

There are many other kinds of kernel methods, and here the most common ones are described. The kernel function that is employed is solely determined by the type of problem. Below given table 1 demonstrated the pros and cons of all the ML algorithms. Supervised learning is more effective when dealing with cases with fewer features and can produce superior classification results. A conclusive result can be supplied for each input, which is very useful for HT detection. But for reference, a golden/design IC is required which is difficult to achieve and this is not suitable for learning on big datasets. These deficiencies can be overcome by using unsupervised ML algorithm. On the other hand, un-supervised learning is susceptible to chaos and is prone to local optimal. Particularly, it is difficult to forecast the outcomes of each training step. Dimensionality reduction technique can help to minimize data dimensions and related attributes.

Table 1: Advantages & Disadvantages of different kind of ML algorithms for HT detection.

Machine Learning Algorithm	Advantages	Disadvantages
Supervised Machine Learning	a) Suitable for fewer feature cases. b) Have proper output for input. c) Accurate classification outcomes. d) Learning model is insensitive towards noise.	a) Golden designs for reference are needed. b) Simple to under fitting and over fitting c) Not proper to train big dataset. d) Multi-classification problems are not able to be addressed.
Unsupervised Machine Learning	a) No need of reference IC. b) Unpredicted output. c) Efficient to train large datasets. d) The model is basic, straightforward to implement, and the performance is independent of parameter selection.	a) Simple to fall into local best solutions. b) Noise affects the learning model. c) Classification outputs aren't very good. d) It's difficult to choose a cluster number. e) Affected by starting cluster center value.
Feature Selection & Dimensionality Reduction	a) Choose the most useful features, reduce data dimensionality, and eliminate redundant elements. b) Reduce attribute space while improving HT detection accuracy and preventing over-fitting.	a) It is a time consuming process. b) As superfluous data, HT features might be lost. c) Threshold is determined manually. d) Multiple attempts and modifications are required.

3. A Review of Machine Learning Methods for Hardware Trojan Detection

3.1 Reverse Engineering

In the first method of detection that is reverse engineering [RE] determines an IC's internal structure, its links, nets, and so many other things to identify how it was created and how it was performed. This process includes decapsulation, delayering, imaging, annotation and schematic creation. RE directly inspects the internal architecture of an IC and then extracts the images of each layer by delayering. Later, these images are compared with the golden/reference chip. Several HT methods have been proposed to detect HT attack using RE method. A very high-resolution scanning electron microscope [SEM] is required to take high resolution images of ever layer which are to be compared with the golden chip. The SEM images of each layer are visually compared with the graphic design file of an IC. Since it is an irreversible and destructive process of HT detection, it might need so many weeks or months to work on a complex structure of an IC hence leads to test limited samples of IC only.

But still this destructive detection process is utilized to get the characteristics of a golden batch of chips [1]. Bao et al. proposed a post-silicon reverse engineering HT detection method for identifying HT-free ICs using ML [19], [20] as shown in figure 5. He used one-class SVM ML models and k-means technique to develop a classifier which automatically differentiates the expected structure of an IC with the suspicious one. From the golden layout N chips are classified according to their grid size, noise margin parameter values. Images of each layer of all N chips are obtained undergoing initial three steps of RE. These chip images are then divided into non interacting grids. For every grid corresponding to each layer features are retrieved. Then, using a subset of the chips, the classifier is trained and derives a decision boundary for every single layer. Based on the v-SVM decision boundaries of each layer, grids in each layer are differentiated and marked as Trojan-free (TF) or Trojan-inserted (TI) after training. Lastly, on the basis of grid categorization each chip is labeled. This whole ML process converts the 5-step complex process into 3 steps only but still golden/reference chip was needed.

Li et al. presented using behavioral pattern mining to reverse the unknown ICs into a tracking model of input and output and behavioral pattern matching to find the Trojan architecture [21]. So this destructive irreversible time consuming process of detecting HT is only valid for simpler circuits and suffers to represent full functional design.

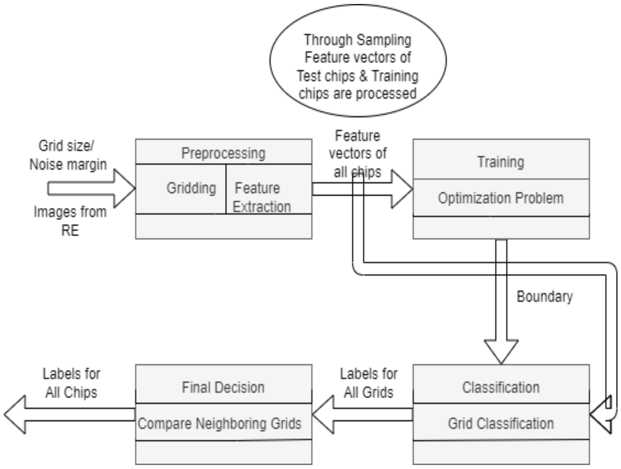

Figure 5 Block diagram of ML based detection method [3]

3.2 Circuit feature analysis

inserted as redundant module in an IC design. They remain in dormant state until some triggering event does not occur. During formal testing state these stealthy malicious HT circuits remain undetected[22], [23]. As a result, circuit characteristics extracted from gate-level netlists, including structural or functional aspects, could be computed and are checked to see whether any of the gate or net is doubtful, where net feature and switching activity are two quantitative metrics often used for Trojan detection. Kasegawa et al. from gate-level netlist collected feature values of some infected net from every net and used a classifier like SVM or ANNs to learn them. Then, a collection of variables from an unidentified gate-level netlist can be categorized using the learnt classifier. Such technique can improve the true positive rate (TPR) of identifying an infected layout; nevertheless, average accuracy and true negative rate (TNR) are also lacking [24].

To find consecutive HTs in 3PIP cores, Zhou et al. introduced a structural feature matching technique [25]. The method initially extracts the LTS induced by HTs into a pointed graph by examining the structural characteristics of less-toggled signals (LTS) in the gate-level circuits of IP cores as shown in figure 6 and uses dynamic CAs for structural feature matching after that.

Hasegawa et al [26] proposed a new approach for classifying HT and Trojan free (TF) ICs using netlist characteristics and machine learning. A group of HT infected and Trojan free nets, as well as a group of new of border nets, were used to train the ML model. The experiments' findings showed how this technology correctly identifies the majority of HT nets. A continuous learning-based methodology was also proposed by Bhunia et al. to check the 3PIPs core integrity of unreliable hardware. [27]. In contrast to traditional learning models that rely primarily on structural information, they combined structural and functional data to generate a resilient training dataset and employed an optimal probabilistic voting array blended with numerous learning methods of improving the HT detection efficiency.

3.3 Side-Channel Analysis

Side-channel analysis (SCA) is a noninvasive post-silicon HT detection technique. SCA compares the circuit

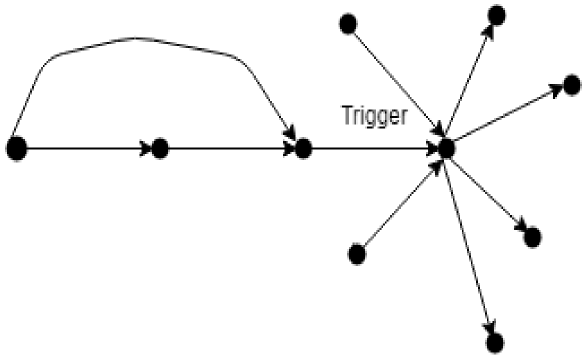

Figure 6 LTS pointed graph due to HT [20]

parameters like power, voltage, electromagnetic emission profile, temperature, and path delays to differentiate a Trojan contained one with the reference IC. The parametric fluctuations in these circuit parameters information provided by additional circuits are extensively used in this analysis. Because side-channel variables can be impacted by noise and process variations (PVs), the signal to noise ratio (SNR) as well as the Trojan to circuit ratio (TCR) have a significant impact on the efficiency of HT identification via side-channel analysis [28]. ML is anticipated to address the shortcomings and optimize the SNR when paired with side-channel analysis [19], [29].

Tang et al. [30] introduced a thermal image-based HT detection and placement technique that can detect HTs in fewer than 20 gates. The temperature shift of the Trojan in the duplicated thermal scan is used to calculate the activity factor. The targeted IC in the overlapped thermal picture is compared to the golden / reference chip created by the modelling software.

Wang et al. published an HT detection approach that uses an Extreme learning machine (ELM) algorithm to sample and categorize current features to determine whether the ICs are affected of Trojans [31]. Plenty of the above mentioned side-channel analysis classifications methods rely on the availability of Trojan-free manufactured ICs, also known as "golden/reference ICs" for training. In actuality, though, realizing such a presumption is challenging. Scholars have tried to employ Clustering algorithms (CA) to side-channel analysis for this reason because they don't require prior knowledge and can significantly limit the dependency on golden layouts. [32]. Side-channel analysis is a non-invasive technique. It is able to adequately detect and obtain excellent detection accuracy rate, particularly with large Trojans.

3.4 Golden model Free Analysis

For HT identification, several strategies, such as SCA, require the existence of reference standards as a starting point. It is not a smart strategy to rely on a reference model because most of the organizations do not give such reference circuits, and creating them is an expensive and time-consuming procedure. Golden model sceptic solutions for preventing HT assaults have evolved to circumvent these constraints.

Very first study of GMF technique was suggested by Jap et al [33]. Side-channel leakage metrics and a one-class SVM classifier were used to create this novel approach. The technique used supervised and unsupervised ML modules to detect HTs either with and without reference / golden model circuits. The testing findings revealed that the recommended technique could detect HTs with excellent accuracy even when there was a lot of noise. Depending on observability and controllability assessments paired with an unsupervised ML clustering algorithm, Salmani et al [34] suggested a unique procedure for detecting HTs in a gate-level netlist. The results demonstrated that this method distinguished Trojan free gates from the infected ones with minimum complexity and without employing reference circuits.

In an another study authors [35] provided a unique automated method for detecting HTs at fabrication phase in which features were extracted from the transient power supply of each simulated ICs utilizing enhanced two class SVM classification. The result depicted a higher accuracy rate in identifying Trojan infected known and unknown ICs.

3.5 Classification Approaches

Yier Jin et al [36] were the first to suggest the use of machine learning approaches for the recognition and characterization of HTs. Depending on on-chip measurement collection and a one-class classifier, the authors developed a universal framework for HT detection. The work was carried out at post deployment phase in a wireless cryptographic IC. The authors demonstrated that the technique could correctly distinguish safe and unsafe regions, as well as conduct post-deployment trust assessments.

Iwase et al [37] proposed a frequency domain feature-based detection method based on an SVM model. They considered applying discrete Fourier transform in which data from power consumption waveform was converted into frequency domain from time domain.

A unique strategy focused on side-channel data while considering an ELM (Extreme Learning Machine) model was reported in another work [31]. The classifier was developed on variable power consumption metrics, and benchmarks revealed that the approach could discriminate contaminated and conventional ICs with a high efficacy, but with noise vulnerability. Qamarina et al. [38] designed a methodology based on ML classification and features identifying the most important characteristics of HTs. The classification of HTs was compared using three machine learning methods: DT, KNN, and SVM. KNN and DT models were shown to be able to predict HTs with an accuracy of above 83 percent.

In one another study [18] , a method for detecting HTs in wireless cryptographic ICs was introduced, in which a one-class classifier was used for data analysis of transmission power. The evaluation findings revealed that this approach could efficiently detect TF circuits from the infected one without having any prior knowledge of the attack details. Lodhi et al. [15] conducted a comparison of three ML modals during the testing process and developed a self-learning framework based on time - varying signals like the propagation delay. The researchers found that only one model, when particularly compared to the others, could accurately identify HTs. Table 2 demonstrates a brief comparison of favorable and unfavorable impact of ML based strategies on all five HT detection methodologies.

Table 2: Comparison of Favorable and Unfavorable ML Approaches for HT Detection.

HT Detection Methodologies	Favorable ML	Unfavorable ML
Reverse Engineering	• There are no reference chips accessible. • Locate intrusive circuits automatically. • Condense the usual 5-stage reverse engineering process into 3 stages. • Avoid manually creating and inserting gate-level netlists.	• Will be using golden designs as a guide. • Parameter influences classification model outcomes. • Clustering methods are noise-sensitive. • Primarily applicable when taking tests.
Circuit Feature Analysis	• Will automatically recognize and categorize HT-net attributes. • Boost TPR and effectiveness. • Take note of the key HT net features. • Make attribute spaces (also known as feature vectors) smaller.	• Demand the use of golden designs as precedent. • Incapable of detecting implicit Trojan. • TNR and accuracy require more development. • The circuit design and chosen numerical measurements affect processing time.
Side channel Analysis	• Reduce the effects of PVs and noises. • Effectively able to minimize data dimensionality and retrieve pertinent features • Align the golden ICs in portion. • Increase TPR, accuracy, etc.	• HT impacts could be eliminated as noise or pointless features. • The choice of pertinent characteristics, ML models, variables, etc., is crucial for efficacy. • Extend the time frame.
Golden Modal Free Analysis	• Doesn't require the golden chips for HT detection. • Can achieve high accuracy rate even under significant noise. • Run time detection can also be achieved at high accuracy rate.	• Efficient for less complex circuits. • Accuracy can be improved further. • This method is particularly suitable at the fabrication stage only not at designing stage.
Classification Approaches	• Doesn't require the golden chips for HT detection. • No prior knowledge of attack is required. • Can be used during testing phase. • Effective for post deployment trust assessments.	• Sensitive to noise. • Not all employed ML models detect HTs with high precision.

3.6 Final Result of Hardware Trojan Detection Approaches

In this section HT detection approaches are summarized and being compared according to their usability as shown in figure 6 below. From the figure it can be seen that RE is the least employed method for HT detection. It represents 13.6% of all the considered techniques in this category. The ML models used in RE were one-class SVM, K means etc and were prepared by extracted features from IC images or grips (Table 3). In the same manner CFA represented 18.2 % of all the total methods (Figure 6). Two-class SVM, Multilayer ANNs, Dynamic CAs etc ML algorithms were recommended in CFA (Table 3), which extracted data set from LTS and net related features.

SCA and CA were the two most frequently (27.3%) employed methods in this category (Figure 6). In SCA side channel leakages like power, delays, EM traces etc were extracted from benchmark circuits to form datasets and the ML models used were mainly SVM, ANN, DT, KNN, ELM, CAs. GMFA is also one of two least frequent (13.6 %) method used in this category (Figure 6) of HT detection. Mainly EM traces and power profiles were the extracted features which were trained by SVM, Clustering, K-NN ML algorithms. Lastly CA being one of the two most frequent method represented in this category holds 27.3 % of share. Transmission power, frequency domain of power consumption, on chip classifiers, power profile etc were the key attributes extracted from benchmarks circuits datasets were then trained using SVM, K-NN, DT, NBC ML models (Table 3). So far different kind of benchmark circuits have been utilized for extracting data sets in these sub categories of Hardware Trojan detection. Mainly Trust-Hub, ISCAS'89, Custom, ITC' 99, Microcontroller, Circuit Simulator, and FPGA are considered for extracting different attributes to prepare datasets which are then trained by using different ML models or algorithms. As per the studies [39] CA utilized Trust-Hub, microcontroller, circuit simulator, ITC’99 and Custom benchmark circuits mostly to extract dynamic power, transmission power, time signature, dominant attributes of HT etc. On the other hand, GMFA used FPGA along with Trust-Hub and microcontroller benchmark circuits for feature extraction. RE used mainly ISCAS’89 and ITC’99 benchmark circuits for extraction of high resolution IC images. SCA utilized Trust-Hub, ISCAS’89, microcontroller, FPGA benchmark circuits to extract transient supply current, power, leakage signals, path delays as feature datasets. CFA on the other hand extracts less toggled signals, gate level net lists features by using Trust-Hub benchmark circuit by employing different ML algorithms.

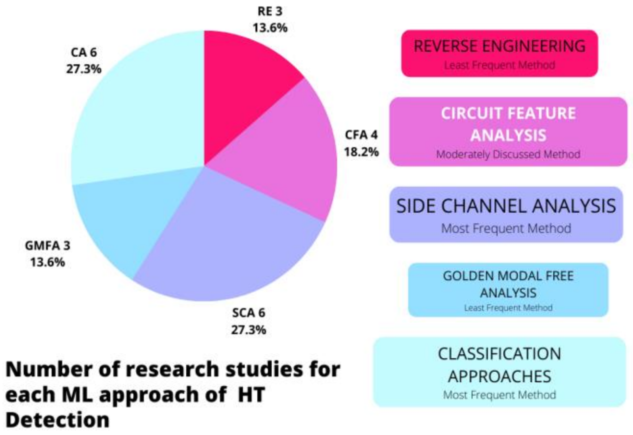

Figure 7 Number of research studies for each ML approach of HT Detection.

Studies have also presented the type of ML models being utilized by each Hardware Trojan detection method. Like SVM was used by all the five HT detection methods mostly for training datasets. On the other hand, CA used all different ML algorithms like ANN, EL, DT, BM, IBM for training the extracted feature data sets.

4. Discussion and Future Work

In the IEEE Xplore digital collection, articles on ML-based strategies used for HT protection are shown in figure 7. As shown below, in the past decade more than 80 research papers have been published on machine learning approaches in HT detection. Most of the study is done in recent years which show the success of ML based HT detection techniques in hardware security. In this article approximately 20 research papers have been included which has discussed and summarized numerous ML models and algorithms for HT defense. HT detection techniques can be classified as pre-silicon and post-silicon according to the process of production of ICs. Here in this article three post-silicon ML based HT techniques have been discussed. Several ML models have been discussed based on which the research papers are discussed and summarized here. In destructive RE, K-means or one-class SVM can categorize or combine IC images. The gate-level netlists of IC designs can be reversed to an input-output track model via pattern mining. In circuit feature analysis detection technique Trojan-related features derived from the net of ICs can be classified using a two-class SVM or multiple ANN. And the Voting Ensemble of RF can categorize functional and structural aspects of ICs, which improves HT detection accuracy. Side-channel analysis HT detection technique included the study which stated that electrical current characteristics of ICs can be classified using ELM. Power consumption, profiling aspects and traces of ICs can be classified or clustered using KNN, DT, DL, PCA, CAs, and CA ensemble. But these process are time consuming and do need a plenty of resources and finances.

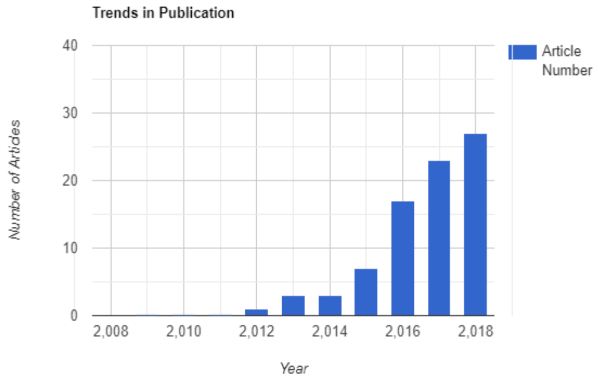

Figure 8 Trends in publishing with ML in HT detection.

Previously done research work is being compared and demonstrated in the below described table 3 while considering different learning models and test data sets. ML, on the other hand, can bypass some of the aforementioned flaws. Golden designs, on the other hand, are required to provide the training dataset for the functional features and structural features of gate-level netlists. And these methods are only useful for detecting explicit Trojans and not for detecting implicit Trojans. ML based circuit feature analysis HT detection approaches are not effective for larger nets circuit. ML based side-channel analysis approach is efficient in decreasing the effect of PVs and noise. It can extract key information and decrease data dimensionality effectively, offset the golden ICs in part and improves TPR and detection accuracy. Trojans, on the other hand, have insignificant effects on circuits; therefore, they may be deleted as irrelevant features or clutter in this approach, lowering the HT detection accuracy.

Table 3: Uses of ML models in HT detection

Article	HT Detection Method	Feature selection	Training data-set	Testing data-set	Learning model	Repeating of trials	Accuracy
J. Chen [19]	RE	IC Images /Grips	29	45	One-class SVM	10	Accuracy = 100%
C.X. Bao [20]	RE	IC Images /Grips	N/A	2000	K-means	500	Accuracy = 97.2% – 100%
K. Hasegawa [26]	CFA	Net-related Features	16	1	Two-class SVM	0 – 20	TNR = 22% –100% TPR = 0% – 100%,
F.K. Lodhi [15]	SCA	Propagation Delay	271	1329	K-NN, DT, BC	500	Accuracy = 93.1% – 95.1% Precision = 79.7% – 95.9%
S. Bhasin [33]	GMFA	EM Traces	150	50	One-class SVM	50	Accuracy = 57.6% – 99.4%
S.Q. Li [25]	CFA	LTS	16	1	Dynamic CAs	NA	Accuracy = 100%
K. Hasegawa [24]	CFA	Net- related Features	16	1	Multi - layer ANN	3	TPR = 74.3 % – 88.9 % TNR = 53.3 % – 70.1 %
Y Makris [18]	CA	Transmission Power	30	90	One-class SVM	30	FPR = FNR = 0%
T. Iwase [37]	CA	Frequency domain of Power Consumption	12	1	Two-class SVM	2918	Accuracy = 100%
G Hospodar [40]	SCA	Power Traces	5K	2K	SVM, KNN, DT	2	SR = 49.9 % – 99.3%
Yier Jin [36]	CA	On Chip classifiers	1K	4K	NA	2	TPR > 97%, TNR > 99% FPR = 0.1 % – 0.2 %, FNR = 0 % – 2.8 %
G Hospodar [40]	SCA	Power Traces	3.5K	1.5 K	bit(4)-SVM (RBF)	3–6	SR = 74.7 % –52.7 %
Q. Cui [41]	SCA	Power Consumption	360K	120K	PCA, Markov	NA	Accuracy = 92% – 100%
F. K. Lodhi [14]	CA	Power Profile	1260	6300	K-NN, DT, NBC	NA	Accuracy = 86.4% – 99%, Precision = 68.4% – 100%

5. Conclusion

Moreover, the efficiency and performance of any ML-based HT detection method are decided by the kind of learning ML models used, their corresponding parameters and related features. In the future, it is anticipated to be an incredibly effective method to identify HTs on ICs. One of the feasible ways to accomplish it is by utilizing and mixing a variety of characteristics mostly from a number of standard / benchmark circuits to create models which can predict HTs both in the pre-silicon and post-silicon stages. Even though machine learning is a potential method for hardware integrity, the system is definitely at serious danger if the detecting accuracy and true positive rate are below 100%. Therefore, we advise that only identified threats be acknowledged when machine learning algorithms are used in safety applications; moreover, occurrences which are labelled as normal should indeed be addressed as questionable because they may be mistakenly categorized as normal when they are actually attacks. The involvement of deep learning algorithms can also enhance the effectiveness and precision of HT detection. More robust ML models are needed to develop, which can be applied effectively to real hardware security applications.

References

1. Z. Huang, Q. Wang, Y. Chen, and X. Jiang, “A Survey on Machine Learning against Hardware Trojan Attacks: Recent Advances and Challenges,” *IEEE Access*, vol. 8, pp. 10796–10826, 2020.

2. M. S. Anderson, C. J. G. North, and K. K. Yiu, “Towards Countering the Rise of the Silicon Trojan,” 2008.

3. Y. Jin and Y. Makris, “Hardware Trojan detection using path delay fingerprint,” *2008 IEEE Int. Work. Hardware-Oriented Secur. Trust. HOST*, pp. 51–57, 2008.

4. M. Banga and M. S. Hsiao, “A novel sustained vector technique for the detection of hardware trojans,” *Proc. 22nd Int. Conf. VLSI Des. - Held Jointly with 7th Int. Conf. Embed. Syst.*, pp. 327–332, 2009.

5. M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey, “Hardware trojan horse detection using gate-level characterization,” *Proc. - Des. Autom. Conf.*, pp. 688–693, 2009.

6. L. Yin, B. Fang, Y. Guo, Z. Sun, and Z. Tian,

"Hierarchically defining Internet of Things security: From CIA to CACA," *Int. J. Distrib. Sens. Networks*, vol. 16, no. 1, 2020.

7. R. Sharma and P. Ranjan, "A Review: Machine Learning Based Hardware Trojan Detection," *IEMECON 2021 - 10th Int. Conf. Internet Everything, Microw. Eng. Commun. Networks*, pp. 1–4, 2021.

8. R. Elnaggar and K. Chakrabarty, "Machine Learning for Hardware Security: Opportunities and Risks," *J. Electron. Test. Theory Appl.*, vol. 34, no. 2, pp. 183–201, 2018.

9. N. H. Farhat, "Photonit neural networks and learning mathines the role of electron-trapping materials," *IEEE Expert. Syst. their Appl.*, vol. 7, no. 5, pp. 63–72, 1992.

10. W. S. Mcculloch and W. Pitts, "A logical calculus nervous activity," *Bull. Math. Biol.*, vol. 52, no. l, pp. 99–115, 1990.

11. X. Chen, L. Wang, Y. Wang, Y. Liu, and H. Yang, "A General Framework for Hardware Trojan Detection in Digital Circuits by Statistical Learning Algorithms," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 36, no. 10, pp. 1633–1646, 2017.

12. M. GhasemiGol, R. Monsefi, and H. S. Yazdi, "Ellipse support vector data description," *Commun. Comput. Inf. Sci.*, vol. 43 CCIS, pp. 257–268, 2009.

13. O. Theobald, "Machine Learning For Absolute Beginners, 2nd Edition-Oliver Theobald(2017)," p. 302, 1385.

14. 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," *Proc. 2017 Des. Autom. Test Eur. DATE 2017*, pp. 294–297, 2017.

15. F. K. Lodhi, I. Abbasi, F. Khalid, O. Hasan, F. Awwad, and S. R. Hasan, "A self-learning framework to detect the intruded integrated circuits," *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2016-July, pp. 1702–1705, 2016.

16. N. Karimian, F. Tehranipoor, T. Rahman, S. Kelly, and D. Forte, "with Ring Oscillator Network (RON)," 2015.

17. R. S. Society, R. S. Society, and A. Statistics, "Algorithm AS 136 A K-Means Clustering Algorithm," vol. 28, no. 1, pp. 100–108, 2012.

18. Y. Liu, S. Member, Y. Jin, and A. Nosratinia, "Silicon Demonstration of Hardware Trojan Design and Detection in Wireless Cryptographic ICs," pp. 1–14, 2016.

19. C. Bao, "On Application of One-class SVM to Reverse Engineering-Based Hardware Trojan Detection," 2014.

20. C. Bao, D. Forte, and A. Srivastava, "On Reverse Engineering-Based Hardware Trojan Detection," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 35, no. 1, pp. 49–57, 2016.

21. U. C. Berkeley and Z. Wasson, "Reverse Engineering Circuits Using Behavioral Pattern Mining," pp. 83–88, 2012.

22. O. G. Netlist and H. Salmani, "COTD : Reference-Free Hardware Trojan Detection and Recovery Based on Controllability and," vol. 12, no. 2, pp. 338–351, 2017.

23. "Hardware Trojan Detection for Gate-level ICs Using Signal Correlation Based Clustering," pp. 471–476, 2015.

24. K. Hasegawa, M. Yanagisawa, and N. Togawa, "Hardware Trojans Classification for Gate-level Netlists Using Multi-layer Neural Networks," pp. 227–232, 2017.

25. E. R. Zhou, S. Q. Li, J. H. Chen, L. Ni, Z. X. Zhao, and J. Li, "A novel detection method for hardware trojan in third party IP cores," *Proc. - 2016 Int. Conf. Inf. Syst. Artif. Intell. ISAI 2016*, pp. 528–532, 2017.

26. K. Hasegawa, M. Yanagisawa, and N. Togawa, "A Hardware-Trojan Classification Method Utilizing Boundary Net Structures," pp. 1–4, 2018.

27. T. Hoque, J. Cruz, P. Chakraborty, and S. Bhunia, "Hardware IP Trust Validation : Learn (the Untrustworthy), and Verify," *2018 IEEE Int. Test Conf.*, pp. 1–10, 2018.

28. B. S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware Trojan Attacks : Threat Analysis and Countermeasures," 2014.

29. J. Brownlee, "Supervised and Unsupervised Machine Learning Algorithms," *Understand Mach. Learn. Algorithms*, pp. 1–9, 2016.

30. Y. Tang, L. Fang, and S. Li, "Activity Factor Based Hardware Trojan Detection and Localization," pp. 293–302, 2019.

31. S. Wang, X. Dong, K. Sun, Q. Cui, D. Li, and C. He, "Hardware Trojan Detection Based on ELM Neural Network," no. 7, pp. 400–403, 2016.

32. M. Xue, R. Bian, W. Liu, and J. Wang, "Defeating Untrustworthy Testing Parties: A Novel Hybrid Clustering Ensemble Based Golden Models-Free Hardware Trojan Detection Method," *IEEE Access*, vol. 7, pp. 5124–5140, 2019.

33. D. Jap, W. He, and S. Bhasin, “Supervised and unsupervised machine learning for side-channel based Trojan detection,” *Proc. Int. Conf. Appl. Syst. Archit. Process.*, vol. 2016-Novem, pp. 17–24, 2016.

34. H. Salmani, “COTD: Reference-Free Hardware Trojan Detection and Recovery Based on Controllability and Observability in Gate-Level Netlist,” *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 2, pp. 338–350, 2017.

35. M. Xue, J. Wang, and A. Hux, “An enhanced classification-based golden chips-free hardware Trojan detection technique,” *Proc. 2016 IEEE Asian Hardw. Oriented Secur. Trust Symp. AsianHOST 2016*, 2017.

36. Y. Jin, D. Maliuk, and Y. Makris, “Post-deployment trust evaluation in wireless cryptographic ICs,” *Proc. -Design, Autom. Test Eur. DATE*, pp. 965–970, 2012.

37. T. Iwase, Y. Nozaki, M. Yoshikawa, and T. Kumaki, “Detection technique for hardware Trojans using machine learning in frequency domain,” *2015 IEEE 4th Glob. Conf. Consum. Electron. GCCE 2015*, pp. 185–186, 2016.

38. N. Q. M. Noor, N. N. A. Sjarif, N. H. F. M. Azmi, S. M. Daud, and K. Kamardin, “Hardware Trojan Identification Using Machine Learning-based Classification,” *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 3-4 Special Issue, pp. 23–27, 2017.

39. K. G. Liakos, G. K. Georgakilas, S. Moustakidis, N. Sklavos, and F. C. Plessas, “Conventional and machine learning approaches as countermeasures against hardware trojan attacks,” *Microprocess. Microsyst.*, vol. 79, no. October, p. 103295, 2020.

40. G. Hospodar, B. Gierlichs, E. De Mulder, I. Verbauwhede, and J. Vandewalle, “Machine learning in side-channel analysis: A first study,” *J. Cryptogr. Eng.*, vol. 1, no. 4, pp. 293–302, 2011.

41. Q. Cui, K. Sun, S. Wang, L. Zhang, and D. Li, “Hardware Trojan detection based on cluster analysis of Mahalanobis distance,” *Proc. - 2016 8th Int. Conf. Intell. Human-Machine Syst. Cybern. IHMSC 2016*, vol. 1, no. 328201505, pp. 234–238, 2016.

42. G. Sumathi, L. Srivani, D. Thirugnana Murthy, K. Madhusoodanan, and S. A. V. Satya Murty, “A Review on HT Attacks in PLD and ASIC Designs with Potential Defence Solutions,” *IETE Tech. Rev.*, vol. 35, no. 1, pp. 64–77, 2018.

43. H. Bai, G. Liu, W. Liu, J. Zhai, L. Yang, and Y. Dai, “Identification of Network Application Behaviors Hiding in HTTP Tunnels,” *IETE Tech. Rev. (Institution Electron. Telecommun. Eng. India)*, vol. 38, no. 1, pp. 112–129, 2021.

Authors

Ritu Sharma is presently working as a Lecturer in the Department of Electronics & Communication Engineering, Technical Department of education, Rajasthan. She received the M.Tech degree in VLSI from Malaviya National Institute of Technology, Jaipur in 2016. She is currently pursuing the Ph.D. degree in University of Engineering and Management Jaipur, Rajasthan, India. Her research work includes Hardware Trojan detection with machine learning, network security.

Email: reetusharma310@gmail.com

Prashant Ranjan is presently working as an Associate Professor in the Department of Electronics & Communication Engineering, University of Engineering & Management Jaipur, Rajasthan, India. He received his M.Tech and Ph.D. degree from Motilal Nehru National Institute of Technology Allahabad, Uttar Pradesh, India. His present area of research includes the design and development of UWB filtering antennas, machine learning, vehicle- to- vehicle wireless technology, Non- Invasive RF Sensors, Agricultural & Medical Applications.

Email: prashant.ranjan@uem.edu.in

List of Tables

List of Figures