IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 August 2022, accepted 20 September 2022, date of publication 26 September 2022, date of current version 3 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3209705

RESEARCH ARTICLE

Data Augmented Hardware Trojan Detection Using Label Spreading Algorithm Based Transductive Learning for Edge Computing-Assisted IoT Devices

VAISHNAVI SANKAR, (Student Member, IEEE), NIRMALA DEVI. M, (Member, IEEE), AND JAYAKUMAR. M, (Member, IEEE)
Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore 641112, India
Corresponding author: Vaishnavi Sankar (s_vaishnavi@cb.students.amrita.edu)

ABSTRACT IoT devices handle a large amount of information including sensitive information pertaining to the deployed application. Such a scenario, makes IoT devices susceptible to various attacks. In addition to securing IoT devices, it is equally important to secure communication among devices and with the outside world. RS232 is a common communication protocol used in IoT and embedded devices. Hence ensuring, Trojan detection in RS232 plays a major role in providing secured communication among edge assisted IoT devices. The inclusion of malicious circuits known as hardware Trojans can occur at any stage of the IC design and manufacturing. Existing pre-silicon detection schemes with static features is limited by the number of features that are learned by the detection scheme. In contrast, machine learning allows enhanced Trojan space exploration. Existing machine learning-based Trojan detection consists primarily of supervised algorithms that rely on high-quality labeled datasets for efficient Trojan detection. Unsupervised methods, on the other hand, underperform due to limited training data and severe imbalance within the available data. To handle such a situation, a semi-supervised hardware Trojan detection has been proposed. In this work, permutation importance guided principal component analysis, correlation aware data augmentation, and hyper-parameter optimization using genetic algorithm aid in optimal dataset and model generation. Pseudo label generation using semi-supervised schemes is utilized to handle partially labeled datasets. For the Trust-HUB benchmarks, the proposed methodology achieves an average of 88.48% true positive rate and 95.77% true negative rate which, clearly indicates the effectiveness and feasibility of semi-supervised hardware Trojan detection.

INDEX TERMS Semi-supervised algorithm, hardware Trojan detection, correlation-aware data augmentation, hyper-parameter optimization, genetic algorithm, permutation importance, XGBoost.

I. INTRODUCTION

The rapid advancement in microelectronic technologies has led to the exploration of cloud computing, big data, artificial intelligence, embedded systems, 5G communication and internet of things (IoT). IoT extends from smart city to smart healthcare including many mission critical systems. IoT framework consists of sensors, actuators and embedded electronic devices that receive, store and transmit data. As per forecast, the number of connected smart devices will reach 75 billion by 2025 [1]. When the number of connected devices grow, there exists a multi-fold increase in the data to be handled. In such a scenario, quality of service (QoS) gets affected due to high network traffic and delay in time-sensitive applications. Edge computing (EC)-assisted IoT devices address the problem of degraded QoS by sharing data processing and enabling self-storage, which reduces the load on the cloud servers [2]. As shown in Fig.1, EC-assisted

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Seo Kim.

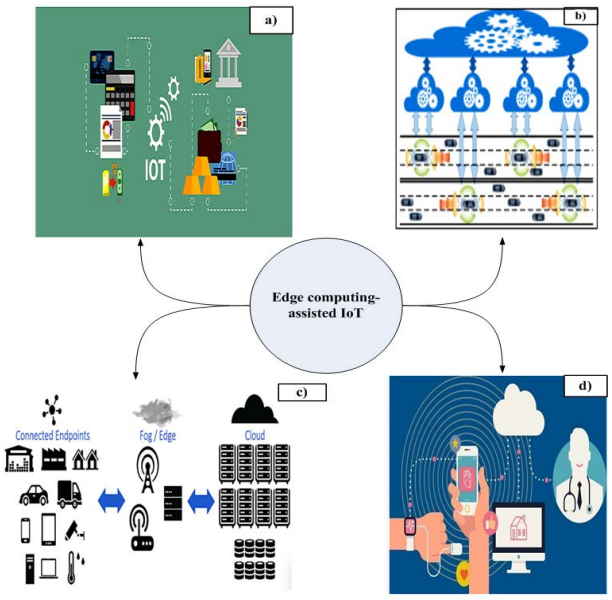

FIGURE 1. Applications of Edge Computing-assisted IoT a) Banking. b) Autonomous driving. c)5G networks. d) Health monitoring.

IoT systems manage a large amount of data pertaining to essential and sensitive applications. The situation makes the IoT devices susceptible to a wide variety of attacks at the software and hardware levels. Due to the necessity of ensuring information security, extensive research has focused on software security issues, neglecting the security hazards in the underlying hardware [3], [4]. Unfortunately, the hardware is still untrustworthy, like the software. The chip's hazards, which lead to cyberspace security threats, should not be overlooked. Among various hardware attacks, hardware Trojans (HT) have emerged as a critical threat [5]. Due to the stealthy nature of HT, it evades the functional testing/verification process intelligently.

High-profit drive, increased competition, and constrained time to market force the IC supply chain to be spread globally [6]. An adversary can insert a HT at any stage of Integrated Circuit (IC) supply chain. Involvement of untrusted parties such as third-party intellectual properties (3PIP) designer [7], computer aided design (CAD) tools [8], fabrication [9], testing [10] and distribution [11] facilitate malicious attacks in all stages. HT attacks span a variety of application platforms such as ML-accelerators [12], IoT devices [13], FPGAs [14], ASICs [15], cryptography cores [16] and CPUs [17]. Successful inclusion and activation of an HT can aid the adversary in accessing confidential information, thereby causing serious concerns.

Existing hardware Trojan detection (HTD) methods can be categorized into static [18] and dynamic detection [19] schemes. Static detection schemes use functional or structural parameters to perform detection, whereas dynamic detection methods apply stimuli for detection. Traditional HTD methods use a limited set of features, can handle a small group of Trojans, and lacks scalability and reusability [20]. An intelligent attacker can redesign the hardware Trojans to surpass traditional detection methods upon gaining knowledge of the utilized features. Machine learning (ML) algorithms can handle a wide variety of Trojans, thereby tackling the aforementioned issue.

Machine learning algorithms extract useful information or patterns from the input data for Trojan identification facilitating the development of reusable and scalable models for HTD. Among the existing machine learning based detection schemes, most methods apply supervised learning, but it is not always possible to have golden reference circuits, considering the real-time scenario. On the other hand, unsupervised strategies use functional features, targeting Trojans with low controllability and transition probability pertaining to their stealthy nature. Such methods can be evaded by redesigning Trojans to satisfy the conditions of a normal circuit [21]. Moreover, the methods that depend on structural features underperform in true positive rate (TPR) due to the limited Trojan space exploration in the training phase.

To be precise, existing machine learning-based Trojan detection approaches suffer from the following limitations. Requirement of a labeled dataset for supervised algorithms, limited learning of the Trojan space in the unsupervised case, and the model's inability to deal with design-specific bias, data imbalance, and/or requirement of light-weight machine learning models. To overcome these limitations, the proposed work uses semi-supervised algorithms for hardware Trojan detection to deal with a partially labeled dataset. Moreover, a dynamic method that can adapt to the new Trojan designs is the need of the hour. The proposed semi-supervised approach use transductive learning, leveraging structural information from graph-based algorithms to perform label predictions effectively on the unseen Trojan data. Furthermore, the method incorporates correlation-aware data augmentation schemes to address the problem of data imbalance. In addition, the method employs a permutation importance-based principal component analysis (PI-PCA) algorithm for feature selection. In addition, the XGBoost model's hyper-parameters are optimized using a genetic algorithm for improved Trojan detection. The following are the technical contributions:

- Execution of semi-supervised algorithms to apply label propagation and label spreading to handle the partially labelled dataset. Pseudo-label generation using transductive learning is adopted to handle unlabeled data
- Incorporation of circuit-based features along with net-based features to reduce the search space and aid the machine learning model to make better predictions. Combined feature set aids in handling design-specific bias.
- Adoption of correlation-aware data augmentation scheme to ensure that the data created is coherent with the original data distribution. The synthetic data samples enhance the label predictions, which in turn improves the detection accuracy

- Permutation importance-based principal component analysis to obtain optimal set of uncorrelated contributive features that enhances the prediction capability of the XGBoost algorithm
- Hyper-parameter optimization of XGBoost algorithm using genetic algorithm to tutor the model for better understanding of the underlying data

The rest of the paper is organized as follows, section II summarizes the existing hardware Trojan detection schemes, section III explains the governing aspects of problem formulation, section IV elaborates the proposed methodology and section V provides experimental results, analysis and inferences. Section VI concludes the work after elaborating the merits, limitations, and suggestions for further exploration.

II. RELATED WORK

Existing hardware Trojan detection (HTD) methods, primarily focusing on the detection at the gate level netlist (GLN) are elaborated in this section. HTD schemes can be classified as pre-silicon and post-silicon detection [19] depending on the scheme applied prior to or after fabrication. Gate-level netlist detection [20], register transfer level (RTL) feature detection [22] and layout level detection [23] constitute HTD at pre-silicon stage. On the other hand, post-silicon detection consists of logical testing [24], [25] and side-channel analysis [26]. Among the wide variety of schemes available in the literature, the exploitation of machine learning algorithms has drawn much attention due to its inherent potential in handling a wide variety of Trojans.

C. H. Kok *et al.* utilized testability measures to train supervised machine learning-based classifiers such as weighted k-nearest neighbour(k-nn), fine gaussian support vector machine, and bagged trees [27]. It is a computationally intensive method that produces more false positives. Testability based HTD approaches was further extended to incorporate structural features [28] or fault modelling techniques [29] to handle the aforementioned limitations. Another reference-free HT detection scheme utilizing testability measures was developed in [30]. Further, information theory-based HT detection approach investigating the relation between transition probability and the information available on a net for unsupervised Trojan detection using density-based clustering algorithm was attempted in [31]. Transition probability and testability measures were further explored in [32] and [33].Limited representative training data resulted in low TPR. Liu *et al.* [34], [35] adopted structural features and testability measure-based features for enhanced Trojan detection. The method is computationally intensive, and its time complexity grows with circuit size. A class weighting scheme and feature selection scheme for XGBoost to tackle the problem of data imbalance and correlation among features was proposed [36]. Hasegawa *et al.* [37], proposed five structural features for HT net identification and employed support vector machine for classification. It used class weighting to handle the data imbalance problem that produced large false positives and false negatives. In the next scheme [38], 51 feature-based HTD had been attempted using a random forest algorithm, which reduced false positives in comparison with [37]. The method adopted f-measure for feature selection to find 11 optimal features from 51 structural features. Mere duplication of minority data using SMOTE caused the generation of false positives. The work was further extended with multi-layer neural network in [39]. Class weighting-based cross-entropy loss function was adopted to handle data imbalance issue. The method produced an average of 83% TPR but underperformed on normal net detection. Dong *et al.* [40] proposed additional structural features over the standard 51 features proposed in [37]. It used feature importance function to choose 49 optimal features, but class imbalance problem had not been dealt with. An effort to combine structural features based HTD with circuit partitioning schemes for Trojan localization, was attempted in [41] and [42]. An unsupervised HTD approach termed PL-HTD, where principal component analysis generates an optimal feature set for unsupervised classification using a local outlier factor algorithm had been attempted [43]. The method produced large false positives due to the poor generalization capability of the model. The triggering properties of Trojan circuits are outlined in [44] and [45] along with feature analysis technique based on a flip-flop level information flow graph. Few-shot learning-based hardware Trojan detection was attempted in [46]. It aims to generate a similarity function based HTD, but the results were not comparable with reported results. An effort to combine static and dynamic features had been attempted in [47] and [48]. Though it had produced 95% average TPR in Trojan detection, the method had not been generalized on varying Trojan circuits.

Among the existing machine learning-based detection schemes, the majority of the methods fall in the supervised category, which is not the case considering the real-time scenario. In addition, there is no unified method of labeling the nets, leading to discrepancies in result interpretation. Unsupervised strategies, in general, adopt testability measure-based features targeting Trojans that have low controllability and low observability [30]. Such methods can be circumvented by redesigning the Trojans to satisfy the conditions of a normal circuit, as mentioned in [21]. Furthermore, strategies that adopt structural features underperform in true positive rate (TPR) due to the limited Trojan space learned in the training phase, causing poor generalization capability. The performance of supervised algorithms relies on the availability of high-quality labeled data. Manual labeling of data for the complete circuit becomes tedious and time-consuming. The problem is further aggravated by the increase in the complexity of circuits. On the other hand, unsupervised algorithms require vast amounts of data to infer patterns revealing Trojan characteristics accurately. Hence a mechanism that overcomes the limitation of both methods becomes essential, considering the diversified threat conditions.

III. PROBLEM FORMULATION

The proposed work caters to the problem of hardware Trojan detection in the pre-silicon stage using a gate-level netlist of the circuit under test (CUT). It adopts an efficient semi-supervised machine learning algorithm that handles the data imbalance problem, feature selection and optimal model generation. The scheme adopts permutation importance-based principal component analysis to remove redundant features that generate large offsets leading to degraded model performance. Further, correlation-aware data augmentation scheme filters out uncorrelated synthetic data produced by adaptive synthetic generation algorithm to generate data that is coherent with original distribution. Furthermore, hyper-parameter optimization using genetic algorithm ensures that the underlying XGBoost model is optimally tuned for effective hardware Trojan detection.

A. PSEUDO LABEL GENERATION FOR HARDWARE TROJAN DETECTION

The development of HT detection algorithms and counterfeiting with new attacks go hand in hand, whereas the availability of labeled data is confined to a limited set of Trojans. This leads to poor generalization on unknown circuits with any new Trojans for supervised HTD schemes. On the other hand, due to the small number of Trojan samples available during the training phase, unsupervised machine learning algorithms face difficulty creating an effective decision boundary. Thus, it becomes important to use the valuable information present in the labeled data to work with unlabeled data.

Such a scenario calls for a semi-supervised algorithm that can work with information available in the labelled dataset to handle unlabeled data. Label propagation and label spreading algorithms that adopt transductive learning for predictions of the partially labeled dataset are explored in the proposed work. The obtained pseudo-labels are combined with labeled data to execute supervised XGBoost algorithm-based Trojan detection.

1) LABEL PROPAGATION

Dataset is split into labeled and unlabeled data and is converted into a weighted connected graph based on Euclidian distance [49]. Label information is propagated through nodes by performing *random walks to absorbing states* in the graph. These data points are manually labeled as 0 or 1, pertaining to the information available in Trust-HUB [50]. The maximum frequency of neighboring states determines the label assigned to the unknown data.

2) LABEL SPREADING

Label spreading [51] algorithm incorporates a method known as spreading activation networks. Points in the dataset are connected in a graph-based on their relative distances in the input space. The algorithm propagates label information upon considering the contribution of the initial labels. The structure in the input space is captured to pass the information through the graph that aid label assignment. It is performed using a weight matrix which is normalized symmetrically. The algorithm dynamically assigns labels depending on the regularization term α, which specifies the percentage of contribution considered from the initial set of labels. This adaptive nature makes it suitable to handle unknown Trojans.

B. CORRELATION-AWARE DATA AUGMENTATION

The small Trojan footprint causes a high degree of imbalance between normal nets and Trojan nets [19]. Correlation-aware data augmentation balances the data by generating synthetic samples coherently with the original data distribution. For synthetic data generation, the proposed scheme uses the adaptive synthetic sampling (ADASYN) [52] algorithm, which considers the density of the data to generate the synthetic samples of minority data. It means ADASYN produces more data samples for harder-to-learn data points. The proposed method captures linear and nonlinear relationships among data using correlation parameters such as *Pearson's correlation coefficient* [53] and *Spearman correlation coefficient* [54], respectively. Pearson correlation coefficient (r) effectively captures the linear relationships between two continuous variables x and y. Its value ranges from -1 to 1. It is calculated using (1).

$$r = \frac{\sum (x_i - \bar{x})\,(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \tag{1}$$

where x_i and y_i are corresponding x and y axis values of the i^{th} sample point and $\bar{x}$ and $\bar{y}$ are the mean values of continuous variables x and y. *Spearman correlation* captures the monotonic relationship among the continuous data. It is calculated on the ranked values of the variables. It is formulated as (2).

$$\rho = 1 - \frac{6 \sum d_i^2}{n\left(n^2 - 1\right)} \tag{2}$$

where d_i is the difference in the ranks of the observation and n is the number of observations. The coherence of the generated data with the original data is verified by analyzing the correlation parameters. Correlation values in the range of 0.7 to 0.9 facilitates the model to maximize the Trojan detection.

C. PERMUTATION IMPORTANCE-BASED PRINCIPAL COMPONENT ANALYSIS FOR FEATURE SELECTION USING BARTLETT's TEST OF SPHERICITY

Presence of correlated features can cause offsets that lead to degradation in model performance and hence has to be removed. The degree of correlation among features is analyzed using Bartlett's test of sphericity as given in (3)

$$\chi^2 = -\left(n - 1 - \frac{2p + 5}{6}\right) \times \ln |R| \tag{3}$$

where n is the number of observations, p is the number of variables, and R is the correlation matrix. The chi square test is

then performed on $(p^2-p)/2$ degrees of freedom. Highly correlated features are removed using principal component analysis [55]. All the data samples are projected to eigenvalues that exhibit maximum variance amongst each other. Such a process yields features that minimize the offsets and enhances the model performance. It does not consider the impact a feature has on the model's predictive capability. For measuring the predictive capability of the model, permutation importance [30] is adopted. It calculates model dependency on the features separately. The features $fsV = \{f_1, f_2, ..f_n\}$ is the original feature set from which, random permutation is performed to form the permuted dataset. The feature importance is calculated as the difference between original and permuted accuracy value which is stored as $Iv = \{Iv_1, I\ v_2, ..I\ v_n\}$. Threshold for feature selection is set by m given by (4). The process of permutation of features and model performance evaluation are iterated until no further enhancement in accuracy is observed.

$$m = \frac{\sum_{i=1}^{n} Iv_n}{n} \tag{4}$$

D. HYPER-PARAMETER OPTIMIZATION USING GENETIC ALGORITHM

Appropriate hyper-parameter selection aids in maximising performance of the underlying ML model thereby, reducing generated errors. Meta-heuristic algorithms are proven to be effective in finding global optimal solution from complex search spaces. Various methods such as particle swarm optimization(PSO), simulated annealing (SA) and ant colony algorithm (ACA) can be used to find the optimal choice of hyper-parameters [56]. When compared to these, genetic algorithm (GA) [57] can find the global optimal solution that is independent of the initial conditions for complex problems. Hence GA is chosen to optimize hyper-parameters and is applied to XGBoost algorithm for Trojan detection. It produces good classification results with its ability to handle large-scale data. Seven of the most influential parameters for the XGBoost algorithm are chosen to be optimized. The parameters are *learning_rate, n_estimators, max_depth, min_child_weight, gamma, sub_sample, colsample_bytree*. *learning_rate* is the step size the model takes for each iteration of residual error correction. A value too low can lead to slow convergence, and a value too high can lead to non-attainment of the global optimum. *n_estimators* define the number of boosted trees present in the ensemble. *max_depth* indicates how deep the tree is with respect to the root node. A lower value leads to underfitting, and a higher value leads to overfitting. *gamma* is the regularisation parameter. *sub_sample* and *colsample_bytree* give the fraction of data and fraction of columns to be randomly sampled for tree generation. A lower value leads to underfitting, and a higher value causes overfitting. Hence obtaining optimal hyper-parameters can lead to the generation of a model that effectively tackles problems such as slow convergence, non-attainment of global optimum, overfitting, and underfitting. Each of the seven hyper-parameters is real vector encoded and concatenated to form a chromosome. Each chromosome represents a hyper-parameter configuration of the XGBoost model. The initial population is assigned a random float value adhering to the predefined ranges of parameter values. F-measure is chosen as the fitness criterion to address the inherent data imbalance problem. The model tries to find hyper-parameters that maximize the selected fitness function.

IV. METHODOLOGY FOR SEMI-SUPERVISED PI-PCA BASED HTD

The proposed work uses semi-supervised algorithm for hardware Trojan detection to deal with the partially labeled dataset. The major steps, include feature extraction, correlation-aware data augmentation, and PI-PCA based feature selection. Genetic algorithm-based hyper-parameter optimization further enhances Trojan detection.

A. THREAT MODEL

The work targets the identification of rarely activated Trojans present in gate level netlist. The chosen HTs can be classified into degrade of *performance (DoP), change of functionality (CoF)* and *denial of service (DoS)*. The work proposes pre-silicon static detection scheme exploiting semi-supervised learning. The proposed scheme has been validated on Trust-HUB circuits with combinational and sequential Trojans.

B. PROPOSED METHODOLOGY

The proposed methodology is illustrated in Fig.2. As the first step, the design is converted into netlist using Synopsys DC [58]. Circuit and net related, 78 features are extracted from the netlist. Permutation importance-based principal component analysis algorithm is performed on the extracted features. It produces an optimal set of uncorrelated and contributive features that maximize the predictive performance of the underlying model. XGBoost model tackles the problem of overfitting due to limited data, by applying regularization. It produces faster convergence by analyzing the feature distribution. Data imbalance in the produced dataset is handled using a correlation-aware data augmentation scheme. It produces synthetic data that is coherent with the original data by satisfying the correlation constraints on the ADASYN algorithm. The scheme removes uncorrelated samples and ensure the coherence of synthetic samples with the original data.

A pseudo label generation algorithm is adopted to make label predictions on the partially labeled dataset. The available labeled data and the generated pseudo labels are combined to form the final training data set. During training, hyper-parameter optimization is performed. The performance of the model is evaluated using test data by adopting the leave-one-out cross-validation method. The adopted testing process makes each circuit considered for testing is unknown to the trained XGBoost model.

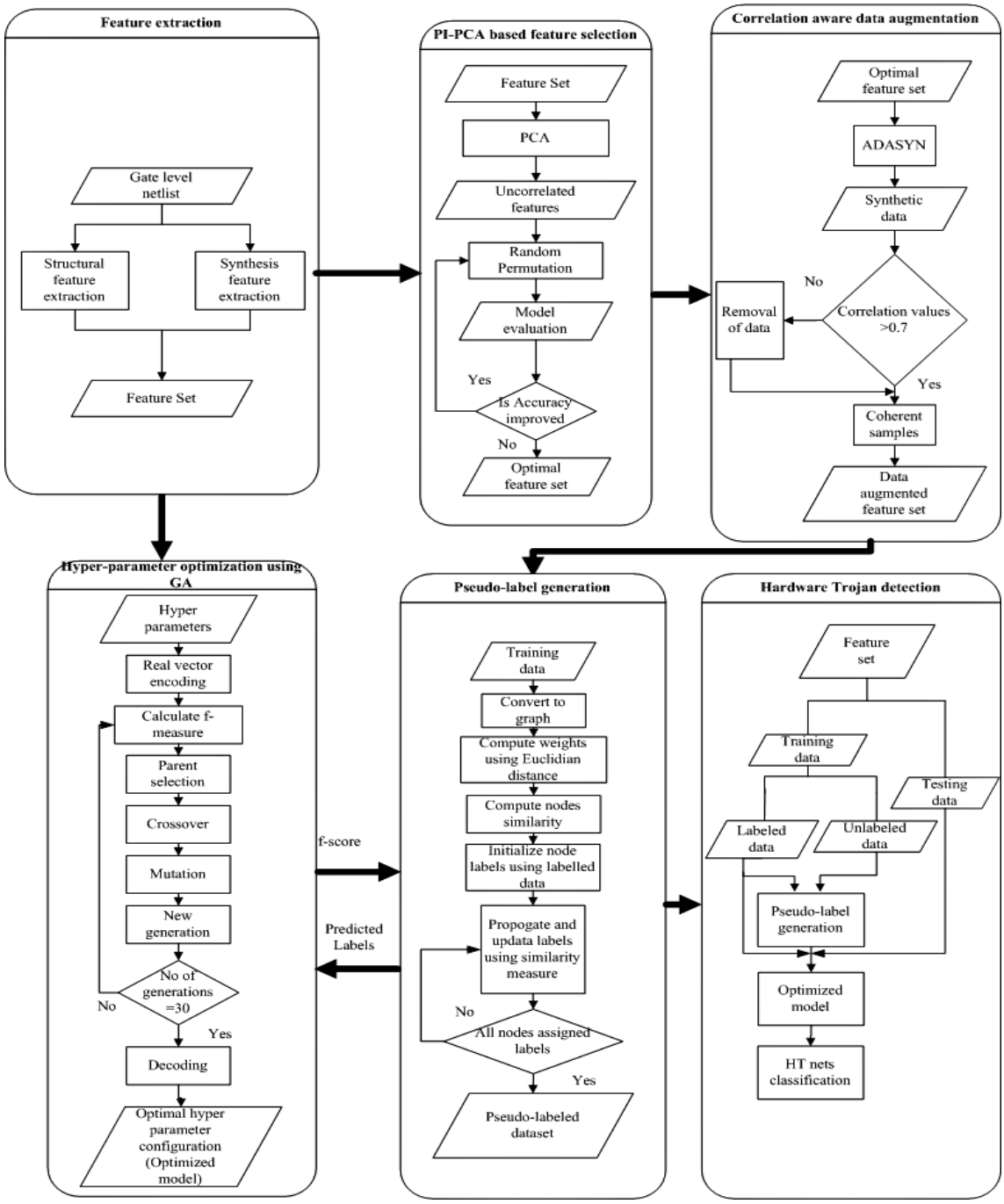

FIGURE 2. Major processes involved in the proposed methodology.

C. FEATURE EXTRACTION

For a particular net n, features such as level, connectivity, primary input, primary output, *fan_in_x*, *in_flipflop_x*, *out_flipflop_x*, *in_multiplexer_x* and *out_multiplexer_x* are calculated. Number of flipflops, multiplexers, and gates up to x level away from the targeted nets are extracted. Circuit-based features are synthesis features extracted from Synopsys DC that include the number of cells, ports, nets, combinational switching power, total switching power, and total power, black box, register, clock network leakage, and total power cell areas of combinational, etc. are defined in Table.1 The adopted feature set helps to tackle the problem of design-specific bias. Trojans can exhibit different characteristics with respect to the inserted design. For example, consider a combinational Trojan with eight trigger inputs inserted in the S38417 and RS232 circuits. It can be observed that although the Trojan is similar in structure, the Trojan in S38417 is harder to activate when compared to that of RS232. Hence it is important to consider both net-based and circuit-based features for effective Trojan identification. The proposed

TABLE 1. Partial list of features ($1 \leq x \leq 5$).

Feature Type	Feature	Feature definition
Net based	Connectivity	Number of gates connected to the net
	Level	The distance from primary input
	Primary input	Level the net is from the primary input
	Primary output	Level the net is from the primary output
	Fan_in_x	The distance from primary input
	In/out flipflop_x	The number of flipflops connected x level from input/output side of net
	In/out_multiplexer	The number of muxes connected x level from input/output side of net n
Circuit Type	Net switching power	The power dissipated when net or internal capacitance charges or discharges upon encountering a change of bit value
	Dynamic power	It is the cumulative sum of switching power and short circuit power. Short circuit power produced due to connection between ground and supply voltage at the instant gate switches.
	Total power	It is the cumulative sum of leakage power and dynamic power

experiment typically considers 78 features comprising 29 net-based and 49 circuit-related features.

D. PI-PCA ALGORITHM FOR FEATURE SELECTION

In order to remove offsets created by correlated and less contributive features, permutation importance-based principal component analysis is executed. Firstly, principal component analysis is performed on the feature set to select features with maximum variance. In addition, we adopt a scheme using permutation importance for feature selection, which is indicative of the generalization capability of the developed model. The impact of each feature on model accuracy is considered after random permutation. The difference between the model performances using the original feature (*Nacc*) set and generated feature set (*Nnewacc*) is taken as the feature importance of the selected features. The average of the feature importance is used as the threshold parameter *m* for feature selection. The process of permutation and feature importance calculation is repeated until no further improvement in model performance is observed such that $Nnewacc \leq Nacc$. The final dataset contains uncorrelated but contributive features to attain enhanced detection accuracy.

E. CORREALATION-AWARE DATA AUGMENTATION

The small Trojan size leads to a severe imbalance in the generated dataset. This, in turn causes the model to develop a bias towards the majority class, which is the normal nets. Hence, to handle the developed bias, synthetic data generation is executed using ADASYN. The density distribution of the data is considered to generate more data points that are harder to detect. Trojan data remain hidden within the normal data points and such a data generation scheme makes the model prone to errors. Hence the data that produces positive correlation values, satisfying the predefined range of correlation values are retained. This further enhances the ability of the model to understand patterns reflecting Trojan characteristics.

F. HYPER-PARAMETER OPTIMIZATION USING GENETIC ALGORITHM

The most influential seven parameters are considered for optimization. All hyper-parameters are real vectors representing a gene that are concatenated to form a chromosome.

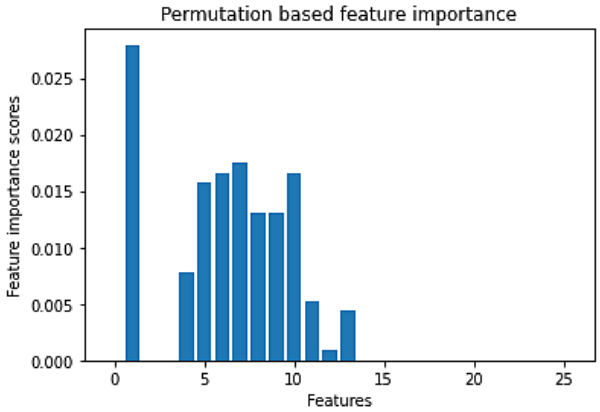

FIGURE 3. Permutation scores of feature set for RS232-T1500 circuit.

It is assigned a random value after which, parent chromosomes are randomly selected for child chromosome generation. Child chromosomes are produced through *crossover* and *mutation*. In the process of *crossover*, a random part of the parent's chromosomes forms the new chromosome. In the process of *mutation*, the values assigned to the gene are changed to a new random value. F-measure is chosen as the fitness criterion to address the data imbalance problem. Chromosomes with the highest fitness values are chosen as parent chromosomes in the succeeding generations, and the process continues. The procedure returns the chromosome with the highest f-measure score upon reaching the user-defined convergence criteria. In the proposed work, max number of generations which is 30 is set as the criterion. The corresponding chromosome gives the optimal hyper-parameter configuration of the XGBoost algorithm. It effectively addresses the problem of overfitting due to the limited training data through regularization. In addition, the XGBoost algorithm considers feature distribution for faster convergence. The efficacy of the proposed algorithm is validated on the Trust-HUB benchmark circuits.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A standard communication protocol used in embedded and IoT devices is universal asynchronous transmitter-receiver (UART) communication. RS232 circuits being the

TABLE 2. Benchmark circuits from trust-HUB.

Circuit Name	Trojan Functionality	Effect of payload
RS232-T1000	Changes certain bits of the transmitted message	CoF
RS232-T1100	Changes certain bits of the transmitted message	CoF
RS232-T1200	Prevents notification for message transmission	DoS
RS232-T1300	Prevents module from receiving and transmitting data	DoS,DoP
RS232-T1400	Prevents module from receiving and transmitting data	DoS
RS232-T1500	Prevent further messages to be received and alters the transmitted message	DoP,CoF
RS232-T1600	Completely stops the module operation	DoS

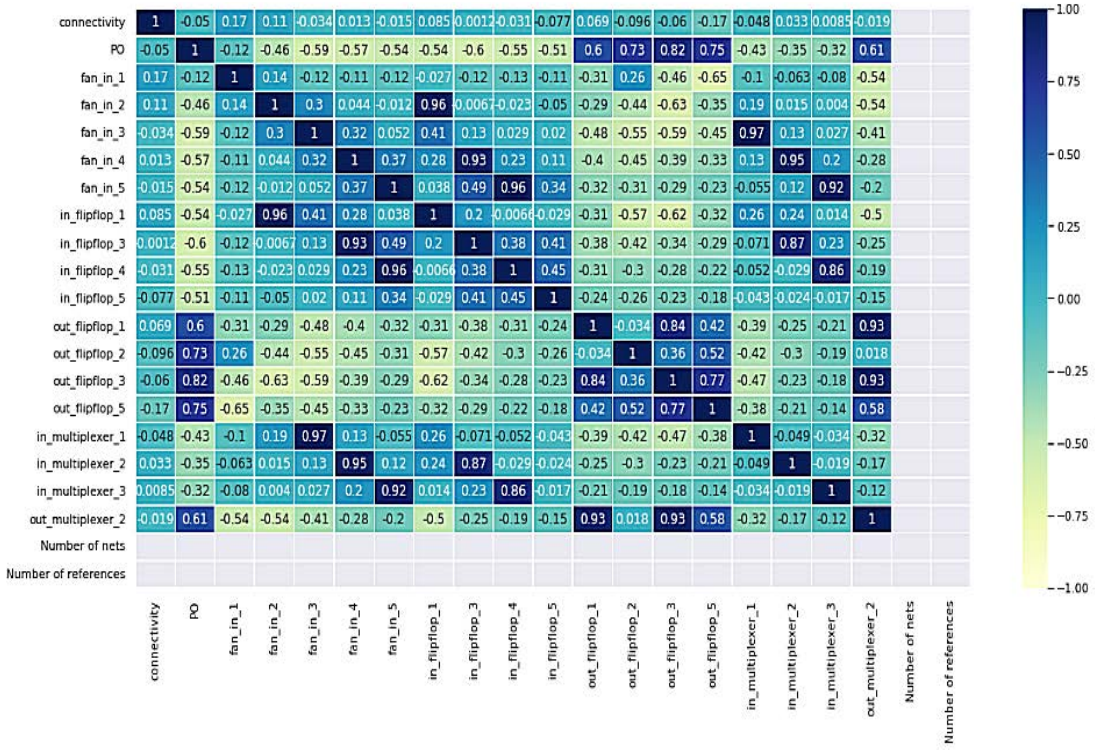

FIGURE 4. Correlogram for the optimized feature set for RS232-T1500 circuit.

underlying hardware, has to be devoid of Hardware Trojans. Hence validating, Trojan detection on RS232 circuit ensures secured communication among edge computing-assisted IoT devices. The circuit under test (CUT) includes a rarely triggered Trojan that covers three popular and challenging payload effects ranging from denial of service(DoS) to change of functionality(CoF) and degradation of performance(DoP).

These CUTs are chosen from the standard set of Trust-HUB benchmark circuits to provide a fair comparison and analysis of the obtained results. Details of the test circuits and inserted Trojans are provided in Table.2. The selected circuits are synthesized by Synopsys Design Compiler(DC) with Semiconductor Manufacturing International Corporation cell library for 90-nm silicon-on-insulator process. The framework of feature extraction, PI-PCA algorithm, correlation-aware data augmentation, hyper-parameter optimization using genetic algorithm, and pseudo label generation algorithm are developed in Python. XGBoost algorithm is utilized for model development using scikit library [59] and executed on an Intel system with Win10 server, running at 1.2GHz with 8GB RAM.

A. DATA PRE-PROCESSING FOR ENHANCED TROJAN DETECTION

Data pre-processing stage consists of permutation importance-based principal component analysis (PI-PCA) for feature selection and correlation-aware data augmentation. Redundant and less contributive features are removed using the PI-PCA algorithm. PCA algorithm selects 21 prominent features that are uncorrelated and exhibit maximum variance from the initial set of 78 features. Since, PCA considers only the global information without looking into local information that can be discriminative for the model predictions. To tackle such a scenerio, permutation importance guided PCA algorithm is developed. It ensures the retention of the most influential seven features from the pruned set of 21 features, as depicted in Fig.3. The correlation plot of the pruned set of features is depicted in Fig.4. Thus, the proposed algorithm

TABLE 3. Correlation coefficients of uncorrelated samples.

Data Sample	Pearson correlation coefficient	Spearman correlation coefficient
Sample 124	0.099	0.4617
Sample 125	0.099	0.460
Sample 136	0.099	0.460
Sample 149	0.022	0.162
Sample 151	-0.014	0.298

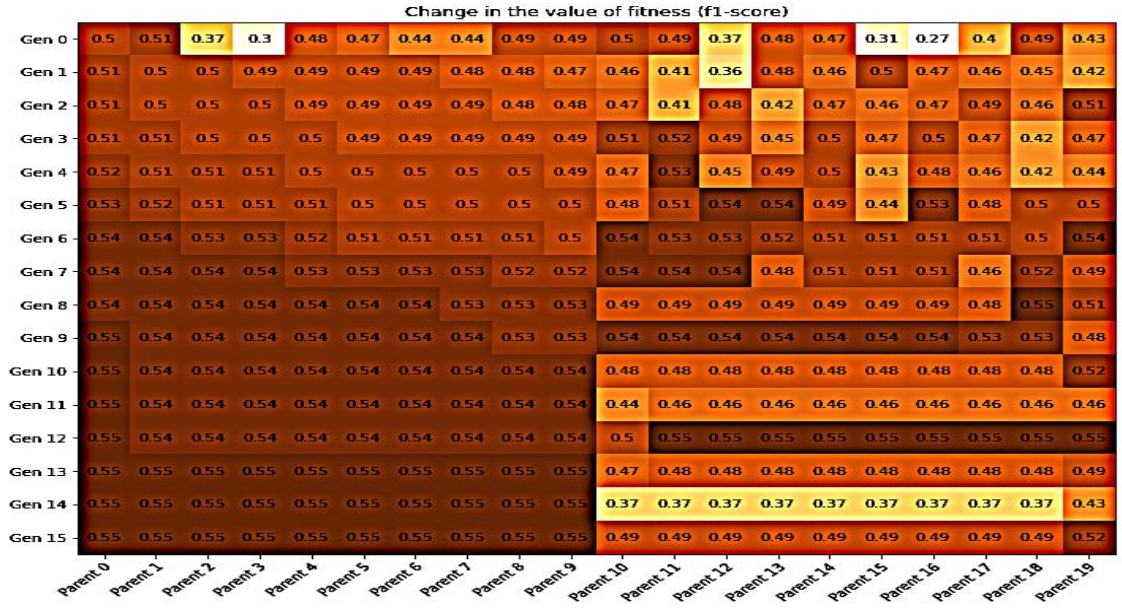

FIGURE 5. Hyper-parameter optimization using genetic algorithm on RS232-T1500.

aids in exploiting the global information captured using the PCA algorithm and local information captured using the PI scheme for attaining optimal feature set.

To select features with maximum contribution, a threshold of 0.01 is set in this experiment. Optimal feature selection significantly reduces the model complexity and leads to lightweight machine learning model. In addition, the large offsets caused by redundant features are also removed.

Upon experimentation, it is observed that the choice of hyper-parameters impacts the detection capability of the model, as depicted in Fig.5. Global search space adopted by genetic algorithm prevents overfitting, underfitting, convergence to local optimum. It further aids in attaining optimal model configuration for enhanced HT detection. Hyper-parameter optimization is performed prior to correlation-aware data augmentation so to handle the imbalanced test data. It is observed that for an imbalanced dataset upon training, the model produces an f-measure spanning a range from 27% to 55%. Despite the influence of severe data imbalance, the model achieves an f-measure of 55% by adopting the appropriate choice of hyper-parameters, as depicted in Fig.5. Despite feature selection, the model attains a recall of 27%, reflecting the impact of bias incurred due to imbalanced dataset. The effect of generating a balanced dataset set is analyzed using receiver operating

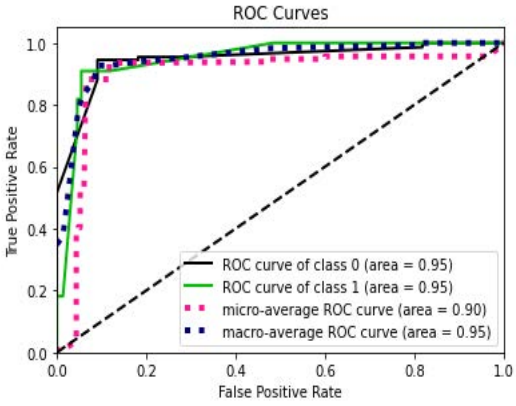

FIGURE 6. ROC curve of RS232-T1500 for optimal feature set.

characteristics (ROC) and precision-recall curves (PR). The capability of the model in performing accurate Trojan detection is reflected in the increased area under the curve(AUC) score. Fig.6 depicts the impact of data imbalance on model performance and is quantified using the AUC score of the Trojan class. Small Trojan footprint to evade standard verification schemes, causes a severe data imbalance in the generated dataset. To tackle this problem, ADASYN is used to create synthetic data. Analysis of the 210 generated synthetic

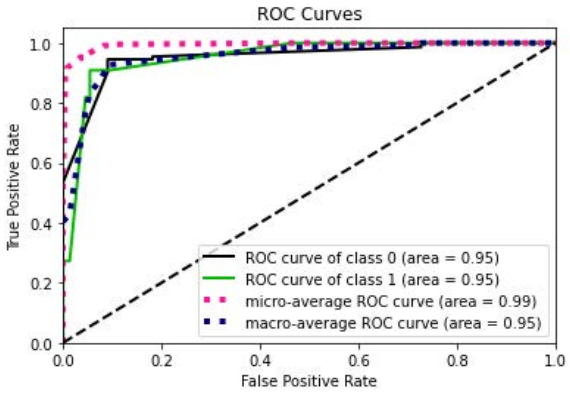

FIGURE 7. ROC curve of RS232-T1500 for correlation aware data augmented dataset.

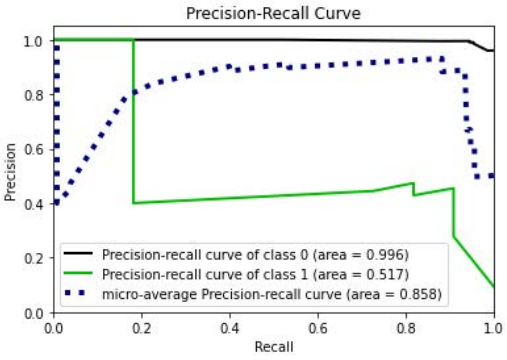

FIGURE 8. PR curve of RS232-T1500 for optimal feature set.

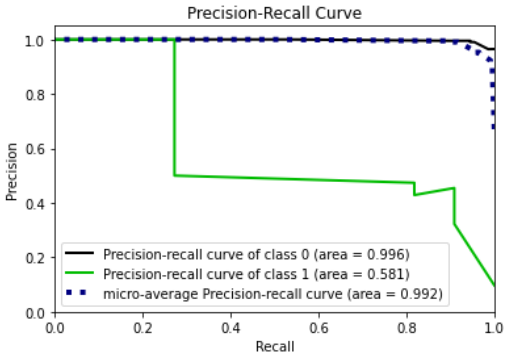

FIGURE 9. PR curve of RS232-T1500 for correlation-aware data augmented dataset.

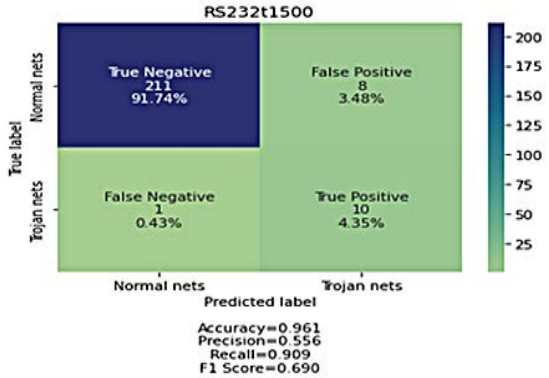

FIGURE 10. Confusion matrix for RS232-T1500.

data samples exhibits that 70 of them are highly uncorrelated with respect to the original data distribution. Correlation analysis is performed by adopting *pearson* and *spearman correlation* in order to verify the coherence of the generated data has with the original data. Correlation coefficients of a few uncorrelated data samples are shown in Table.3. To aid accurate detection, the uncorrelated data points are removed, which led to the improvement of the f-measure from 35.3% to 42.6%. Improved AUC score of 7% as depicted in Fig.6 and Fig.7 further confirms the scheme's efficacy in generating balanced data set. The effectiveness of an HTD scheme relies on the ability of the model to maximize Trojan detection which is achieved using the generated dataset. In addition, the accuracy of Trojan detection enhances by 0.09% as observed in Fig.8 and Fig.9. Thus the generated balanced data set aids in effective Trojan detection with minimal trade-off incurred for Trojan net and normal net detection.

B. EVALUATION METRICS FOR RESULT ANALYSIS

The results are analyzed using precision, recall, f-measure, accuracy, receiver operating characteristic curve, precision-recall curve, true positive rate, and true negative rate [19] and are depicted in Table.4. They are derived from the confusion matrix shown in Fig.10. For binary classification of positive and negative classes, the matrix is generated using parameters such as True negative (TN), true positive(TP), false negative(FN), and false positive (FP). For the application of hardware Trojan detection, Trojan nets are represented as positive class and normal nets as negative class. Fig.10 exhibits the confusion matrix generated using the aforementioned notation for the RS232-T1500 test circuit. In the field of Trojan detection, the efficacy of the model relies on its ability to improve Trojan recognition and reduce the normal net miss-classification rate. In effect, this translates to minimization of the generation of false positives and false negatives.

C. HARDWARE TROJAN DETECTION USING PARTIALLY LABELLED DATASET

Label propagation and label spreading algorithm have been applied to the pre-processed data to generate pseudo labels. The dynamic nature of the label generation process of label spreading algorithm makes it suitable for the application of Trojan detection. It is observed that the value of *alpha* which denotes the ratio of information inferred from the neighboring nodes and from the initial labels, impacts model performance. TNR value increases with decrease in the contribution of initial label information, and the highest TNR is reached by adopting an *alpha* of 0.8 to 0.9 on average. Labeled data and generated pseudo labels are combined to form the final dataset, which is then applied to the optimized

TABLE 4. Performance metrics for evaluation of trojan detection.

Performance metrics	Definition	Formula
Recall (True positive rate)	The rate at which the positive class samples are predicted as positive class samples.	$TPR = TP/(FN + TP)$
Precision (P)	The rate at which the negative class samples are predicted as negative class samples	$P = TP/(FP + TP)$
F-measure (F)	It is derived from the precision and recall rates. It gives the harmonic mean of recall and precision rates	$F = 2PR/(P + R)$
Accuracy (A)	It is the rate at which the data instances are correctly predicted with respect to the total predications made by the classifier	$A = (TP+TN)/(TP+TN+FP+FN)$
True negative rate (TNR)	It is the rate at which the normal nets are correctly predicted as normal nets	$TNR = TN/(TN + FP)$
Receiver operating characteristic(ROC)	The efficiency with which the model handles both true positive rates and false positive rates	-
Precision recall (PR) curve	It is the tradeoff between detection of both classes	-

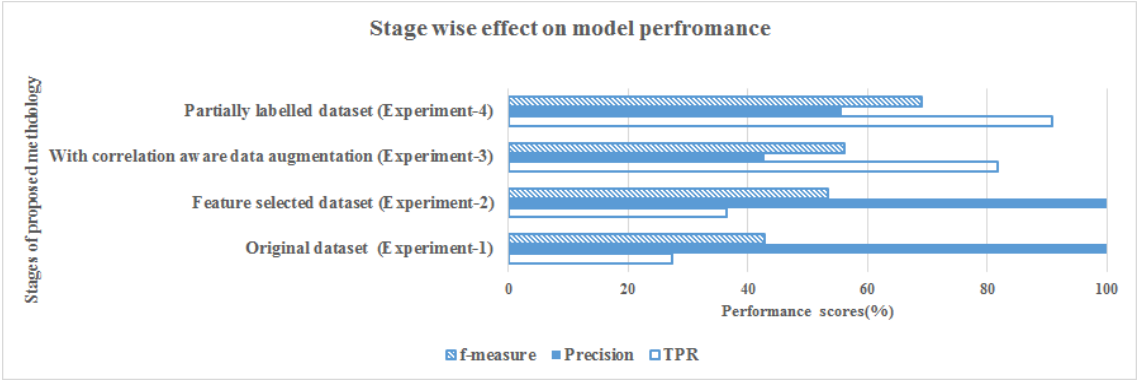

FIGURE 11. Impact of each process on Trojan detection for RS232-T1500.

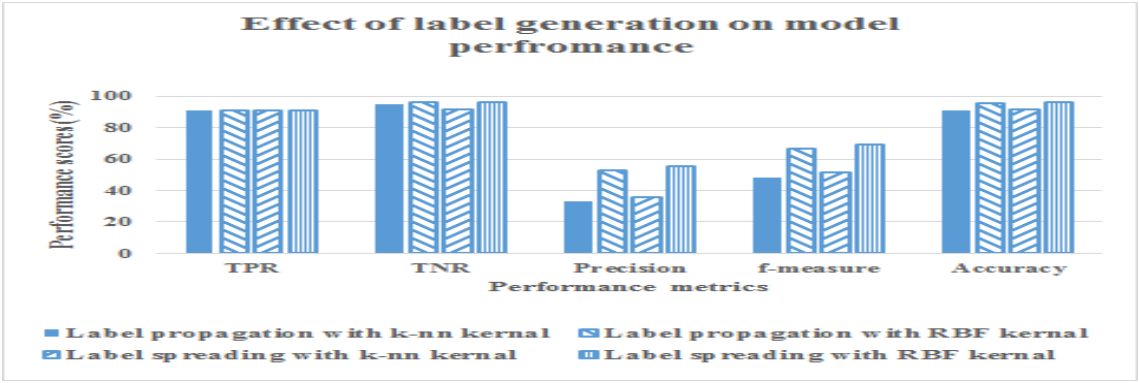

FIGURE 12. Impact of various label generation schemes on Trojan detection for RS232-T1500.

XGBoost algorithm for Trojan detection on the chosen test circuits. Fig.11 indicates the impact each stage of operation has on model performance. Each stage of operation is reflected in the nomenclature of the resultant dataset. Fig.11 indicates the performance metrics attained by the model post feature selection, correlation-aware data augmentation, and pseudo-label generation, respectively. It can be observed that despite feature selection stage, before data augmentation in Experiment.2, the model achieves high precision rate at the cost of recall rate, reflecting the impact of data imbalance. Upon correlation-aware data augmentation indicated by Experiment.3, the model attains an improved precision, recall, and f-measure as indicated in Fig.11. The exploitation of structural information and the available prior information by the graph-based transductive approaches in Experiment.4, results in optimal model performance and is indicated by the improved f-measure. Semi-supervised algorithms are realized using the scikit library. Upon experimenting with the available kernels such as radial basis function (RBF) kernel and Knn kernel, the former obtained optimal Trojan detection results as illustrated in Fig.12. The dynamic nature of label prediction adopted by the label spreading algorithm makes

TABLE 5. TPR and TNR comparison of PCA based local outlier factor algorithm(unsupervised) and proposed work(PW).

Circuit Name	TPR(%) [43]	TPR(%)(PW)	TNR (%) [43]	TNR(%) (PW)
RS232-T1000	50	90	96.43	95.8
RS232-T1100	45.45	90	96.43	95.4
RS232-T1200	46.15	100	97.14	95.2
RS232-T1300	57.14	66.7	96.04	94.7
RS232-T1400	41.67	90.9	96.4	96.7
RS232-T1500	45.45	90.9	96.45	96.3
RS232-T1600	44.44	90	96.11	96.3
Average	47.19	**88.49**	**96.43**	95.77

TABLE 6. Performance comparison of PCA based local outlier factor algorithm and proposed work (PW).

Circuit Name	f-measure(%) [43]	f-measure(%)(PW)	Precision(%) [43]	Precision(%)(PW)	Accuracy(%) [43]	Accuracy(%)(PW)
RS232-T1000	40	64.3	33.33	50	94.83	95.6
RS232-T1100	28.21	62.1	33.33	47.4	94.5	95.2
RS232-T1200	27.3	72.2	40	56.5	94.56	95.5
RS232-T1300	27.69	66.7	26.67	26.7	95.09	94.7
RS232-T1400	24.44	71.4	33.33	58.8	94.14	96.7
RS232-T1500	25.8	69	33.33	55.6	94.54	96.3
RS232-T1600	27.69	66.7	26.7	52.9	94.52	96.3
Average	28.73	**63.4**	32.38	**49.7**	94.59	**95.57**

TABLE 7. TPR and TNR comparison of multi-layer neural network(supervised) and proposed work(PW) in terms of TPR and TNR.

Circuit Name	TPR(%) [39]	TPR(%)(PW)	TNR(%) [39]	TNR(%)(PW)
RS232-T1000	100	90	24	95.8
RS232-T1100	78	90	25	95.4
RS232-T1200	91	100	55	95.2
RS232-T1300	86	66.7	65	94.7
RS232-T1400	100	90.9	15	96.7
RS232-T1500	82	90.9	47	96.3
RS232-T1600	100	90	28	96.3
Average	**91**	88.49	37	**95.77**

it more suitable for hardware Trojan detection. Furthermore, it can be observed that the model effectively uses the information retrieved from the generated dataset and the structural information obtained from the produced graph to achieve optimal detection.

D. PERFORMANCE COMPARISON WITH EXISTING WORKS

The efficacy of supervised HTD schemes relies on the high quality labeled dataset. Whereas, obtaining high-quality datasets with labels is tedious and time-consuming and there exist discrepancies in the process of data labeling. On the other hand, unsupervised algorithms require a large amount of unlabeled data to identify patterns reflecting Trojan characteristics effectively. Hence a semi-supervised approach that uses prior label information for the prediction of unlabeled data becomes the need of the hour, which is attempted in this work. The efficiency of a model in Trojan detection is analyzed by TPR and TNR scores. The work aimed at enhancing TPR with minimum possible degradation of TNR using partially labeled datasets. In comparison with an unsupervised approach attempted in [43], the model produces an improvement of 41.3%, 34.67%, 17.32%, and 0.981% in terms of TPR, f-measure, precision, and accuracy respectively as depicted in Table.5 and Table.6. The valuable prior information in the labeled data has been exploited in the proposed semi-supervised algorithm to enhance the TPR when compared to [43]. The improved TPR values can be attributed to the utilization of initial cluster information by the label spreading algorithm that reveals significant relationships among data samples within the dataset. It is observed from Table.7, that the adequate learning of Trojan characteristics has led to an appreciable performance in comparison with supervised learning [39]. Overall, an improvement of 58.87% is observed for TNR, 36.64% in terms of f-measure, 33.88% precision, and 52.32% in terms of accuracy, as depicted in Table.7 and Table.8. Table.9 compares the performance of the proposed work with existing supervised schemes such as [37], [38], unsupervised schemes [30], [31] and few-shot learning based schemes [46] in terms of TPR. The method outperforms [31], [37] and [30] by 4.03%, 16.62% and 11.9% in terms of TPR. The method achieves comparable performance in comparison with [38]. Supervised approaches largely rely on the availability of high qualtity labeled dataset for effective Trojan detection. This is further possible by the procurement of golden circuit of the base design and prior knowledge of the inserted Trojan structure as experimented in [38]. However, in reality, this is not the case. Moreover, with rapidly evolving Trojan designs, an approach to handle

TABLE 8. Performance comparison of multi-layer neural network and proposed work (PW).

Circuit Name	f-measure(%) [39]	f-measure (%)(PW)	Precision(%) [39]	Precision(%)(PW)	Accuracy(%) [33]	Accuracy(%)(PW)
RS232-T1000	25.08	64.3	14.34	50	32.58	95.6
RS232-T1100	2027	62.1	11.65	47.14	30.96	95.2
RS232-T1200	31.74	72.2	19.22	56.5	59.75	95.5
RS232-T1300	32.31	66.7	19.89	26.7	66.93	94.7
RS232-T1400	27.95	71.4	16.24	58.8	27.13	96.7
RS232-T1500	28.95	69	17.57	55.6	51.24	96.3
RS232-T1600	21.04	66.7	11.8	52.9	34.23	96.3
Average	26.76	**63.4**	15.82	**49.7**	43.25	**95.57**

TABLE 9. TPR comparison of proposed work with existing work.

Circuit Name	Supervised learning TPR(%) [37]	Supervised learning TPR(%) [38]	Few shot learning TPR(%) [46]	Unsupervised learning TPR(%) [31]	Unsupervised learning TPR(%) [30]	Semi-supervised learning TPR(%)(PW)
RS232-T1000	84.09	100	NA	62	53.33	90.9
RS232-T1100	80.95	50	NA	67	58.33	90
RS232-T1200	78.79	88	NA	89	80	100
RS232-T1300	87.1	100	NA	89	88.89	66.7
RS232-T1400	86.96	98	NA	61	83.33	90.9
RS232-T1500	93.33	95	NA	73	83.33	90.9
RS232-T1600	80	93	NA	62	88.89	90
Average	84.46	89.14	70.3	71.85	76.58	**88.48**

NA:Not available

unknown Trojan data needs to be addressed, which forms the basis of our work. The proposed methodology adopts a semi-supervised scheme that leverages a transductive learning approach and structural information from a graph-based algorithm to adeptly handle unknown Trojan data. Quantitatively, the proposed approach attains, on average 88.48% TPR and 95.77% TNR scores, thereby obtaining a better trade-off between TPR and TNR values with respect to existing approaches.

Overall, it can be observed that although supervised schemes produce better performance, the high-quality labeled dataset is hard to achieve considering the real-time scenario. In contrast, the proposed model surpasses the TPR achieved by few-shot learning [46] and unsupervised learning by its efficiency in utilizing prior information in the partially labeled dataset and the structural information from the generated graphs. Thus, the proposed scheme sheds light on the exploration of semi-supervised hardware Trojan detection. Experiment analysis confirmed that the proposed work that combined pseudo-label generation with correlation-aware data augmentation has significantly enhanced the model performance.

VI. CONCLUSION AND FUTURE WORK

Existing Trojan detection methods face limitations such as the requirement of labeled datasets for supervised algorithms, limited learning of the Trojan space, and the model's inability to deal with design-specific bias, data imbalance, and/or requirement of lightweight ML models. Such limitations are tackled in the proposed work using semi-supervised algorithms for hardware Trojan detection using partially labeled datasets. Permutation importance-guided principal component analysis has been adopted to capture both global and local information for efficient feature reduction. Correlation-aware data augmentation curates the ADASYN algorithm to generate data coherent with the underlying data distribution for optimal data balancing. In addition, genetic algorithm-based hyper-parameter optimization maximizes Trojan detection by attaining hyper-parameter configuration resulting in a global optimum. Furthermore, a graph-based semi-supervised scheme that utilizes transductive learning effectively uses prior information in the partially labeled dataset and the structural information from the generated graphs for enhanced detection performance. The efficiency and feasibility of the proposed work have been established upon comparison with existing supervised, unsupervised, and few-shot learning-based schemes of hardware Trojan detection. The proposed methodology achieves 88.48% average true positive rate and 95.57% average true negative rate for the Trust-HUB benchmark circuits. Specifically, RS232 benchmark test circuits are chosen to validate the proposal. Ensuring Trojan detection of the RS232 circuit plays a major role in providing secured communication among edge computing-assisted IoT devices. In the era of the connected world, the very volatile nature of edge computing to security threats faced by IoT devices compel this choice.

Experimentation and analysis on the test circuits indicate the effectiveness and feasibility of a semi-supervised approach for hardware Trojan detection. The computational complexity of graph creation for pseudo-label generation linearly increases with the circuit size and has to be optimized. The exploitation of explainable machine learning to avoid manual intervention for result analysis, extending to incorporate more variety of Trojan designs and optimized pseudo-label generation are the suggested future work.

REFERENCES

[1] B. Safaei, A. M. H. Monazzah, M. B. Bafroei, and A. Ejlali, "Reliability side-effects in Internet of Things application layer protocols," in *Proc. 2nd Int. Conf. Syst. Rel. Saf. (ICSRS)*, Dec. 2017, pp. 207–212.

[2] R. Vinayakumar, M. Alazab, S. Srinivasan, Q.-V. Pham, S. K. Padannayil, and K. Simran, "A visualized botnet detection system based deep learning for the Internet of Things networks of smart cities," *IEEE Trans. Ind. Appl.*, vol. 56, no. 4, pp. 4436–4456, Jul. 2020.

[3] T. A. Ahanger, A. Aljumah, and M. Atiquzzaman, "State-of-the-art survey of artificial intelligent techniques for IoT security," *Comput. Netw.*, vol. 206, Apr. 2022, Art. no. 108771.

[4] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[5] A. Dhavlle, R. Hassan, M. Mittapalli, and S. M. P. Dinakarrao, "Design of hardware trojans and its impact on CPS systems: A comprehensive survey," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.

[6] H. Li, Q. Liu, and J. Zhang, "A survey of hardware trojan threat and defense," *Integration*, vol. 55, pp. 426–437, Sep. 2016.

[7] K. Basu, S. M. Saeed, C. Pilato, M. Ashraf, M. T. Nabeel, K. Chakrabarty, and R. Karri, "CAD-base: An attack vector into the electronics supply chain," *ACM Trans. Design Autom. Electron. Syst.*, vol. 24, no. 4, pp. 1–30, Jul. 2019.

[8] C. Pilato, K. Basu, F. Regazzoni, and R. Karri, "Black-hat high-level synthesis: Myth or reality?" *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 4, pp. 913–926, Apr. 2019.

[9] S. Kaji, M. Kinugawa, D. Fujimoto, and Y.-I. Hayashi, "Data injection attack against electronic devices with locally weakened immunity using a hardware trojan," *IEEE Trans. Electromagn. Compat.*, vol. 61, no. 4, pp. 1115–1121, Aug. 2019.

[10] 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.

[11] 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, 2018.

[12] X. Hu, Y. Zhao, L. Deng, L. Liang, P. Zuo, J. Ye, Y. Lin, and Y. Xie, "Practical attacks on deep neural networks by memory trojaning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 6, pp. 1230–1243, Jun. 2021.

[13] S. Guo, J. Wang, Z. Chen, Y. Li, and Z. Lu, "Securing IoT space via hardware trojan detection," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11115–11122, Nov. 2020.

[14] C. Krieg, C. Wolf, and A. Jantsch, "Malicious LUT: A stealthy FPGA trojan injected and triggered by the design flow," in *Proc. 35th Int. Conf. Comput.-Aided Design*, Nov. 2016, pp. 1–8.

[15] S. Ghandali, T. Moos, A. Moradi, and C. Paar, "Side-channel hardware trojan for provably-secure SCA-protected implementations," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 6, pp. 1435–1448, Jun. 2020.

[16] A. Jain and U. Guin, "A novel tampering attack on AES cores with hardware trojans," in *Proc. IEEE Int. Test Conf. Asia (ITC-Asia)*, Sep. 2020, pp. 77–82.

[17] A. De, M. N. I. Khan, K. Nagarajan, and S. Ghosh, "HarTBleed: Using hardware trojans for data leakage exploits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 4, pp. 968–979, Apr. 2020.

[18] M. Priyatharishini and M. N. Devi, "A deep learning based malicious module identification using stacked sparse autoencoder network for VLSI circuit reliability," *Measurement*, vol. 194, May 2022, Art. no. 111055.

[19] R. S. Chakraborty, S. Pagliarini, J. Mathew, S. R. Rajendran, and M. N. Devi, "A flexible online checking technique to enhance hardware trojan horse detectability by reliability analysis," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 2, pp. 260–270, Apr. 2017.

[20] M. Qin, W. Hu, X. Wang, D. Mu, and B. Mao, "Theorem proof based gate level information flow tracking for hardware security verification," *Comput. Secur.*, vol. 85, pp. 225–239, Aug. 2019.

[21] N. Zhang, Z. Lv, Y. Zhang, H. Li, Y. Zhang, and W. Huang, "Novel design of hardware trojan: A generic approach for defeating testability based detection," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 162–173.

[22] H. S. Choo, C. Y. Ooi, M. Inoue, N. Ismail, M. Moghbel, and C. H. Kok, "Register-transfer-level features for machine-learning-based hardware trojan detection," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E103.A, no. 2, pp. 502–509, Feb. 2020.

[23] T. M. Supon, M. Seyedbarhagh, R. Rashidzadeh, and R. Muscedere, "A method to prevent hardware trojans limiting access to layout resources," *Microelectron. Rel.*, vol. 124, Sep. 2021, Art. no. 114212.

[24] L. Kampel, P. Kitsos, and D. E. Simos, "Locating hardware trojans using combinatorial testing for cryptographic circuits," *IEEE Access*, vol. 10, pp. 18787–18806, 2022.

[25] C. Nigh and A. Orailoglu, "AdaTrust: Combinational hardware trojan detection through adaptive test pattern construction," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 3, pp. 544–557, Feb. 2021.

[26] A. Vakil, A. Mirzaeian, H. Homayoun, N. Karimi, and A. Sasan, "AVATAR: NN-assisted variation aware timing analysis and reporting for hardware trojan detection," *IEEE Access*, vol. 9, pp. 92881–92900, 2021.

[27] 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," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.

[28] M. Priyadharshini and P. Saravanan, "An efficient hardware trojan detection approach adopting testability based features," in *Proc. IEEE Int. Test Conf. India*, Jul. 2020, pp. 1–5.

[29] P. Naskar, T. Dhar, and S. K. Roy, "Hardware trojan detection using improved testability measures," in *Proc. Int. Symp. Devices, Circuits Syst. (ISDCS)*, Mar. 2020, pp. 1–6.

[30] M. Tebyanian, A. Mokhtarpour, and A. Shafieinejad, "SC-COTD: Hardware trojan detection based on sequential/combinational testability features using ensemble classifier," *J. Electron. Test.*, vol. 37, no. 4, pp. 473–487, Aug. 2021.

[31] R. Lu, H. Shen, Z. Feng, H. Li, W. Zhao, and X. Li, "HTDet: A clustering method using information entropy for hardware trojan detection," *Tsinghua Sci. Technol.*, vol. 26, no. 1, pp. 48–61, Feb. 2021.

[32] Y. Su, H. Shen, R. Lu, and Y. Ye, "A stealthy hardware trojan design and corresponding detection method," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–6.

[33] K. Huang and Y. He, "Trigger identification using difference-amplified controllability and dynamic transition probability for hardware trojan detection," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3387–3400, 2020.

[34] Q. Liu, P. Zhao, and F. Chen, "A hardware trojan detection method based on structural features of trojan and host circuits," *IEEE Access*, vol. 7, pp. 44632–44644, 2019.

[35] P. Zhao and Q. Liu, "Density-based clustering method for hardware trojan detection based on gate-level structural features," in *Proc. Asian Hardw. Oriented Secur. Trust Symp. (AsianHOST)*, Dec. 2019, pp. 1–4.

[36] R. Sharma, N. K. Valivati, G. K. Sharma, and M. Pattanaik, "A new hardware trojan detection technique using class weighted XGBoost classifier," in *Proc. 24th Int. Symp. VLSI Design Test (VDAT)*, Jul. 2020, pp. 1–6.

[37] K. Hasegawa, M. Oya, M. Yanagisawa, and N. Togawa, "Hardwaretrojans classification for gate-level netlists based on machine learning," in *IEEE 22nd Int. Symp. Line Test. Robust Syst. Design (IOLTS)*, Jul. 2016, pp. 203–206.

[38] 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," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 100, no. 12, Dec. 2017, pp. 2857–2868.

[39] K. Hasegawa, M. Yanagisawa, and N. Togawa, "Hardware trojans classification for gate-level netlists using multi-layer neural networks," in *Proc. IEEE 23rd Int. Symp. On-Line Test. Robust Syst. Design (IOLTS)*, Jul. 2017, pp. 227–232.

[40] C. Dong, J. Chen, W. Guo, and J. Zou, "A machine-learning-based hardware-trojan detection approach for chips in the Internet of Things," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 12, Dec. 2019, Art. no. 155014771988809.

[41] M. Du, Z. Huang, Y. Chen, L. Li, Q. Wang, and J. Liu, "A HT detection and diagnosis method for gate-level netlists based on machine learning," in *Proc. IEEE 6th Int. Conf. Signal Image Process. (ICSIP)*, Oct. 2021, pp. 1070–1074.

[42] Z. Huang, C. Xie, Z. Li, M. Du, and Q. Wang, "A hardware trojan detection and diagnosis method for gate-level netlists based on different machine learning algorithms," *J. Circuits, Syst. Comput.*, vol. 31, no. 7, May 2022, Art. no. 2250135.

[43] C. Dong, Y. Liu, J. Chen, X. Liu, W. Guo, and Y. Chen, "An unsupervised detection approach for hardware trojans," *IEEE Access*, vol. 8, pp. 158169–158183, 2020.
[44] K. Hasegawa, M. Yanagisawa, and N. Togawa, "A hardware-trojan classification method utilizing boundary net structures," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2018, pp. 1–4.
[45] T. Kurihara and N. Togawa, "Hardware-trojan classification based on the structure of trigger circuits utilizing random forests," in *Proc. IEEE 27th Int. Symp. Line Test. Robust Syst. Design (IOLTS)*, Jun. 2021, pp. 1–4.
[46] T. Lu, F. Zhou, N. Wu, F. Ge, and B. Zhang, "Hardware trojan detection method for gate-level netlists based on the idea of few-shot learning," in *Proc. IEEE 21st Int. Conf. Commun. Technol. (ICCT)*, Oct. 2021, pp. 301–305.
[47] S. Li, Y. Zhang, X. Chen, M. Ge, Z. Mao, and J. Yao, "A XGBoost based hybrid detection scheme for gate-level hardware trojan," in *Proc. IEEE 9th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC)*, Dec. 2020, pp. 41–47.
[48] Y. Zhang, S. Li, X. Chen, J. Yao, Z. Mao, J. Yang, and Y. Hua, "Hybrid multi-level hardware trojan detection platform for gate-level netlists based on XGBoost," *IET Comput. Digit. Techn.*, vol. 16, nos. 2–3, pp. 54–70, Mar. 2022.
[49] S. E. Garza and S. E. Schaeffer, "Community detection with the label propagation algorithm: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 534, Nov. 2019, Art. no. 122058.
[50] *Trust-HUB*. [Online]. Available: http://www.trust-hub.org
[51] K. Prokopchik, "Nonlinear label spreading on hypergraphs," Gran Sasso Sci. Inst., Italy, Tech. Rep., 2022.
[52] J. Beinecke and D. Heider, "Gaussian noise up-sampling is better suited than SMOTE and ADASYN for clinical decision making," *BioData Mining*, vol. 14, no. 1, pp. 1–11, Dec. 2021.
[53] M. Li, D. Zou, S. Luo, Q. Zhou, L. Cao, and H. Liu, "A new generative adversarial network based imbalanced fault diagnosis method," *Measurement*, vol. 194, May 2022, Art. no. 111045.
[54] M. Esmailoghli, J. A. Quiané-Ruiz, and Z. Abedjan, "COCOA: COrrelation COefficient-aware data augmentation," in *Proc. EDBT*, 2021, pp. 331–336.
[55] H. BM and A. AM, "A review of principal component analysis algorithm for dimensionality reduction," *J. Soft Comput. Data Mining*, vol. 2, no. 1, pp. 20–30, 2021.
[56] B. Akay, D. Karaboga, and R. Akay, "A comprehensive survey on optimizing deep learning models by metaheuristics," *Artif. Intell. Rev.*, vol. 55, pp. 829–894, Mar. 2021.
[57] J. Chen, F. Zhao, Y. Sun, and Y. Yin, "Improved XGBoost model based on genetic algorithm," *Int. J. Comput. Appl. Technol.*, vol. 62, no. 3, pp. 240–245.
[58] *Design Compiler Reference Manual*, Simulator, Synopsys, Mountain View, CA, USA, 1994.
[59] F. Pedregosa, S. Varoquaux, A. Gramfort, V. Michel, and B. Thirion, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Dec. 2011.

VAISHNAVI SANKAR (Student Member, IEEE) received the bachelor's degree in electronics engineering from the Marian Engineering College, Ettimadai, India, in 2017, and the Master of Engineering degree from Amrita Vishwa Vidyapeetham, India, in 2019. She is currently a Research Scholar at the Hardware Security Research Group, Amrita Vishwa Vidyapeetahm, Ettimadai. Her research interest includes hardware security.

NIRMALA DEVI. M (Member, IEEE) received the B.E. degree in electronics and communication engineering (ECE) from the Government College of Technology, Coimbatore, in 1990, and the M.E. degree in applied electronics from Bharathiar University, in 1996. She is currently working as a Professor with the Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore. She is the Vice Chairperson with the Electronics and Communication Engineering Department, School of Engineering, Coimbatore. Her research interests include VLSI design and testing, computational intelligence, hardware security and trust, evolvable hardware, and RF CMOS system design. She has served as the Board of Studies Member. Marquis Who's Who in the World 2011 distinguishes her as one of the leading achievers from around the country. International Biographical Centre, Cambridge, U.K., has chosen her for inclusion in the prestigious publication "2000 Outstanding Intellectuals of the 21st Century–2011."

JAYAKUMAR. M (Member, IEEE) worked at the Vikram Sarabhai Space Center, Indian Space Research Organization, Trivandrum, as a Scientist in the radio frequency systems for space vehicle projects, from 1996 to 2001. He joined Amrita Vishwa Vidyapeetham, in July 2004, after working in satellite communication industries for eight years. He currently works as the Chairperson with the Electronics and Communication Engineering Department, School of Engineering, Coimbatore. His research interests include radio frequency system design, planar antenna systems, RFIC design, hardware security, systems design for air-borne vehicles, and wireless communication systems. He has got the IEEE Student Award in one of the IEEE Conference (IEEE–InterMag–MMM-1994), Albuquerque, NM, USA. He was awarded Dr. K. S. Krishnan Research Fellowship by the Department of Atomic Energy (DAE) in Engineering Science.