DETECTION OF RECYCLED FIELD PROGRAMMABLE GATE ARRAYS USING CLUSTERING ALGORITHM

by
TANVIR AHMAD TARIQUE
1017312012

MASTER OF SCIENCE
IN
INFORMATION AND COMMUNICATION TECHNOLOGY
(M.Sc. in ICT)

Institute of Information and Communication Technology (IICT)
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY (BUET)
Dhaka, Bangladesh
November 2022

The thesis titled "DETECTION OF RECYCLED FIELD PROGRAMMABLE GATE ARRAYS USING CLUSTERING ALGORITHM", submitted by Tanvir Ahmad Tarique, Student ID: 1017312012, Session: October 2017, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Information and Communication Technology (M.Sc. in ICT) held on 28th November 2022.

BOARD OF EXAMINERS

1. Dr. Md. Liakot Ali — Chairman (Supervisor)
   Professor
   Institute of Information and Communication Technology (IICT)
   Bangladesh University of Engineering and Technology (BUET)
   Dhaka-1000, Bangladesh.

2. Dr. Md. Rubaiyat Hossain Mondal — Member (Ex-Officio)
   Professor and Director
   Institute of Information and Communication Technology (IICT)
   Bangladesh University of Engineering and Technology (BUET)
   Dhaka-1000, Bangladesh.

3. Dr. Md. Jarez Miah — Member
   Assistant Professor
   Institute of Information and Communication Technology (IICT)
   Bangladesh University of Engineering and Technology (BUET)
   Dhaka-1000, Bangladesh.

4. Dr. Md. Fokhrul Islam — Member (External)
   Professor
   Department of Electrical and Electronic Engineering (EEE)
   Islamic University of Technology (IUT)
   Gazipur, Bangladesh.

AUTHOR'S DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree, diploma, or other qualifications.

Tanvir Ahmad Tarique
ID: 1017312012

DEDICATION

I dedicate my thesis work to my beloved parents (Late Zafar Ahmad and Sufia Jabeen), my dear sister Dr. Shazia Jabeen, and my dear wife Eumna Bushra along with my lovely 1 year 10 months old son Tauqir Ahmad Ayaan for giving me their generous love, care, support and enthusiasm to complete my Master of Science research.

CONTENTS

List of Figures

List of Abbreviations

Abbreviation	**Elaboration**
FPGA	Field-Programmable Gate Array
IC	Integrated Circuits
KFF	Known Fresh FPGA
RO	Ring Oscillator
CP	Comparisons
FP	Fingerprint
X-FP	Exhaustive Fingerprint
VP	Virtual Probe
uLSIF	unconstrained Least-Squares Importance Fitting
AI	Artificial Intelligence
ML	Machine Learning
USD	United States Dollar
CA	Clustering Algorithm
ERAI	Electronic Resellers Association International
HPC	High-Performance Computing
DL	Deep Learning
CAGR	Compound Annual Growth Rate
COVID	Coronavirus Disease
SRAM	Static Random-Access Memory
NLP	Natural Language Processing
IoT	Internet of Things
TV	Television
USICA	U.S. Innovation and Competition Act
FUT	FPGA under test
SVM	Support Vector Machine
CLB	Combinational Logic Block

Abbreviation	**Elaboration**
LUT	Look-Up Table
IP	Intellectual Properties
ADAS	Advanced Driver-Assistance System
OCM	Original Component Manufacturer
SIA	Semiconductor Industry Association
ESD	Electro-Static Discharge
PIN	Part or Identifying Number
PCB	Printed Circuit Board
ASIC	Application Specific Integrated Circuits
HDL	Hardware Description Language
VHDL	Very High-Speed Integrated Circuit HDL
CPU	Central Processing Unit
ALM	Adaptive Logic Module
LB	Logic Block
TPU	Tensor Processing Unit
GPU	Graphics Processing Unit
TDDB	Time-dependent Dielectric Breakdown
BTI	Bias Temperature Instability
PBTI	Positive Bias Temperature Instability
NBTI	Negative Bias Temperature Instability
HCI	Hot Carrier Injection
EM	Electromigration
OC-SVM	One-Class SVM
XOR	XOR logic gate
XNOR	XNOR logic gate
JTAG	Joint Test Action Group
PLL	Phase-Locked Loop
UART	Universal Asynchronous Receiver Transmitter
LFSR	Linear Feedback Shift Register

Abbreviation	Elaboration
ROC	Receiver Operating Characteristic
AUC	Area under the ROC Curve
RMSE	Root Mean Square Error
GAN	Generative Adversarial Network

List of Tables

ACKNOWLEDGEMENT

At first, I would like to thank and all praise to the Great Almighty Allah, the most merciful, the most gracious, the source of knowledge and wisdom endowed to mankind, who provided me with the power of mind, strength, patience and capability to carry me through the work and enable me to complete this thesis.

I would like to thank greatly to my Supervisor, **Prof. Dr. Md. Liakot Ali**, Institute of Information and Communication Technology (IICT), Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh for his kind, constant, and inspiring guidance, close encouragement, advice, and valuable suggestions at all stages for preparing this dissertation.

I am also very grateful to **Prof. Dr. Md. Rubaiyat Hossain Mondal**, Director of IICT, BUET for providing the great support of different facilities of IICT especially the **Embedded System Lab of IICT** to conduct my thesis research works. My special thanks to all the Faculty Members and Staffs of IICT, BUET.

In completing this work, I have been fortunate to get help, support and encouragement from many people. I would like to acknowledge all of them for their cooperation. Especially, I am greatly grateful to my ex-colleague, Dr. Foisal Ahmed, Postdoc Fellow at Department of Computer System, Tallinn University of Technology, Estonia. I am also thankful to the Dependable System Lab (DSL) at Nara Institute of Science and Technology (NAIST), Japan for providing their FPGA datasets to use them in my thesis research.

Finally, I would like to thank to my parents, my sister, my wife and my son for their continuous support and inspiration throughout the whole period of this research.

ABSTRACT

Field Programmable Gate Array (FPGA) is a popular electronic component used in many applications due to their cost-effectiveness, competitive performance, and power efficiency. However, some third-party vendors in the semiconductor industry collect used FPGAs and refurbish them to sell as a new, posing security and reliability issues for mission-critical systems. Researchers have proposed various methods to detect recycled FPGAs, including ring oscillator-based delay analysis or fingerprint (FP) analysis using supervised machine learning (ML) technique. However, these methods require a large amount of data and time, which is not practical due to the rapidly changing technology and large number of FPGAs in the industry. Unsupervised machine learning approaches require less data but still require a significant amount of comparison calculations to achieve high accuracy, which is costly and time-consuming. Finding a faster and cheaper solution to this problem is necessary. Fresh FPGAs have different FP patterns than that of recycled FPGAs. This property has been used by other researchers for classification of recycled FPGAs from fresh FPGAs. However huge computation is required in this case. This thesis has introduced a novel technique to reduce the computational complexities using the property of symmetricity of the structure of FPGAs. Due to systematic process variation within the FPGA, the neighboring combinational logic blocks (CLBs) of FPGAs have similar or symmetrical array structures, leading to similar FPs in the neighboring logic blocks. This symmetrical property has been exploited for detecting recycled FPGAs using Clustering Algorithm (CA)-based anomaly or outlier detection scheme with K-means++ technique which analyzes the neighboring ring oscillator (RO) frequencies' symmetrical or similarity information. The proposed symmetry analysis method efficiently detects all the recycled FPGAs through outlier detection, achieving 92% accuracy in a very short period of time with around 41% less computations compared to the previous unsupervised ML-based method. In future, research can be carried out to improve the accuracy using more reduced computations.

Chapter 1
Introduction

1.1 Introduction

Nowadays, modern civilization inevitably relies on computer systems to improve our lives in every sector. The dependability of these systems is essential to ensure good functionality and performance in delivering the services to the systems. In general, the hardware dependability includes the attribute of availability, reliability, safety, integrity, maintainability, and confidentiality [1]. Specifically, hardware security is currently one of the most important reliability issues for the computer system.

Field-programmable gate arrays (FPGAs) have become highly prevalent among integrated circuits (ICs) due to their advantageous features such as low development expenses and quick time-to-market. Consequently, even reused FPGAs are frequently employed, considering the complex nature of contemporary electronics supply chains. As a result, FPGAs are now regarded as the most sought-after ICs. [2]. Moreover, there is a novel trend of using FPGAs as accelerators for artificial neural networks. [3]. PGAs offer multiple benefits, including cost-effective integration, superior performance, and energy efficiency, which are driving their adoption in edge AI devices, AI workstations, and High-Performance Computing (HPC) applications. AI-enabled FPGAs are employed in several data center devices, such as networking equipment, storage racks, and server systems, enabling users to manage high-speed data processing and monitor network traffic. Furthermore, the major data center operators' concerted efforts to enhance process efficiency will encourage market growth. [4].

The FPGA market has experienced growth in the healthcare industry since 2020, largely due to the COVID-19 pandemic. The heightened need for high-performance detection devices in hospitals has spurred developers to create infection detection systems based on FPGAs. For instance, ALDEC Inc. developed a COVID-19 lung infection detection system. The ongoing research and development and innovations in the healthcare sector are expected to foster market expansion in the years to come. [4].

To address the aforementioned challenge and explore new opportunities in the industry, FPGA market players are concentrating on developing cutting-edge SRAM memory solutions. For example, in February 2022, QuickLogic Corporation, a California-based semiconductor manufacturing company, launched a low-power FPGA based on SRAM technology to mitigate semiconductor supply and availability issues. These advancements are expected to foster market growth in the projected timeline. [4].

In Fig. 1.1, the low-range segment in the FPGA market is anticipated to exhibit a growth rate of approximately 15% until 2028, primarily driven by the numerous high-end features it offers, including low logic density, high power efficiency, and reduced complexity. These features have accelerated the adoption of low-range FPGAs in several portable electronic devices such as wearable devices, edge computing devices, and wireless gateways. [4]. The <28 nm segment of the FPGA market generated over USD 1.5 billion in revenue in 2021 and is expected to grow at a rate of 14% during the forecast period. The growth can be attributed to the various high-end features offered by this segment, such as high-speed processing, compact size, and improved efficiency, among others. These characteristics have accelerated the adoption of <28 nm FPGAs in multiple markets, including automotive electronics, high-performance computing, and telecommunications. [4].

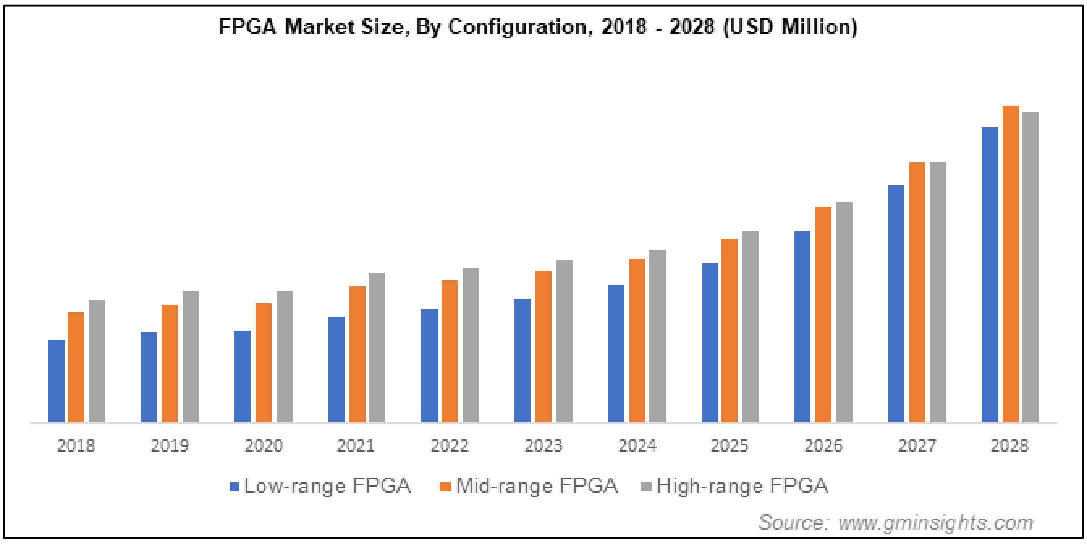


Fig. 1.1: Rising demand for low-power FPGAs in portable devices for high energy efficiency [4]

In 2021, the consumer electronics sector accounted for approximately 9% of the FPGA market share. The growing disposable income in developing countries is driving the demand for new appliances, leading to an increasing market demand. FPGA solutions are integrated into various consumer electronics such as smartphones, laptops, digital cameras, game consoles, and tablets. Furthermore, the rising adoption of new technologies such as IoT, Natural Language Processing (NLP), and AI in smart speakers, smart TVs, and edge AI devices will fuel market growth in the future. [4].

The North American FPGA market is projected to grow at a CAGR of over 14% from 2022 to 2028, driven by the increasing government initiatives and funding activities to boost the regional semiconductor sector. For instance, in July 2021, the U.S. government passed the U.S. Innovation and Competition Act (USICA), which is an initiative aimed at boosting semiconductor manufacturing. The bill includes a total funding of USD 250 billion to launch innovative products and USD 52 billion for R&D activities in semiconductors, among other initiatives. [4].

The security of the integrated circuits (ICs), which is the most essential part of any computer system, is now becoming a rising threat, especially the counterfeit ICs [5]. Counterfeited electronics components are now a deep-rooted problem that has created significant concern in the ICs supply chain, and are impacting the IC industries, computers, communication systems, medical, and telecommunication systems. Specifically, the problem of counterfeit ICs attracts a lot of attention not only to the private sectors but also the government because the global counterfeit market has grown significantly compared to the past history. Fig. 1.2 shows the recent data provided by Electronic Resellers Association International (ERAI) showing the scenarios of recent incidents of counterfeit components since 2005 [6]. These results indicate that the risk of counterfeit material still exists in large numbers although some preventive measures have been taken.

Counterfeit ICs can be classified into several categories, including recycled, remarked, overproduced, defective, cloned, and more. Recycled components are the most common type of counterfeit ICs, accounting for over 80% of the total counterfeit components. Recycled components refer to those that have been previously used or recycled and are being sold as new, genuine products. This type of counterfeit ICs poses a significant risk to the electronics industry, as recycled components may not function as intended and can compromise the safety and reliability of electronic devices. [5].

Certainly, recycling of FPGAs is a significant concern given their increasing usage and the prevalence of recycled counterfeit components in the market. Using recycled FPGAs may compromise the performance, reliability, and safety of electronic devices. Therefore, it is crucial for the electronics industry to ensure the authenticity and quality of FPGAs and other semiconductor components to avoid the risks associated with using counterfeit or recycled components. This can be achieved through proper testing, inspection, and certification procedures, as well as by working with trusted suppliers and distributors. [7].

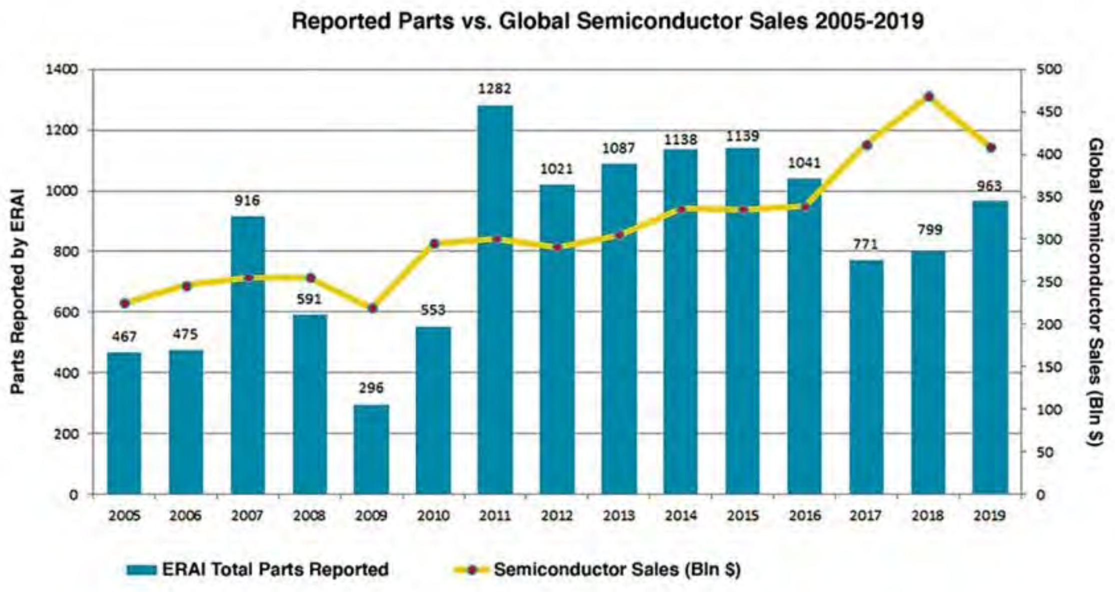


Fig. 1.2: Counterfeit incident report [6]

Of course, recycled FPGAs pose significant risks to the reliability and performance of electronic devices. Due to their prior usage, recycled FPGAs may have already undergone wear and tear, leading to degradation in performance over time. Additionally, recycled FPGAs may be compromised and contain hidden defects or malicious code, which can cause serious reliability and security issues in critical applications.

Preventing the infiltration of recycled FPGAs is a challenging and costly task, as it requires stringent testing and verification procedures to ensure the authenticity and quality of the components. This is particularly important in critical applications, such as aerospace, defense, and medical devices, where the reliability and safety of the system are of utmost importance. To mitigate these risks, it is essential for the electronics industry to work with trusted suppliers and distributors and implement robust testing and inspection procedures to ensure the authenticity and quality of FPGAs and other semiconductor components. So, it can be understood that the importance and usages of the FPGAs in the current time are very huge. And so, as the preparing and infiltrating of the recycled FPGAs are also booming for the many dishonest suppliers or third-party vendors. So, this is the concern of this work to tackle the infiltration of these recycled FPGAs in the supply chain of the FPGA market.

1.2 Existing Works & Challenges

It has been realized that the recycled FPGAs are a major concern in the IC supply chain due to the increasing number of third-party IC vendors, leading to a higher risk of counterfeit components. As mentioned earlier, recycled components account for more than 80% of the counterfeit components, which can pose a significant threat to the reliability and performance of critical applications. [5]. These recycled FPGAs may have reliability risks and trustworthiness issues due to the aging-induced performance degradation. As FPGAs are used, they may experience wear and tear, which can lead to degradation in performance over time. This aging process can impact the reliability and trustworthiness of recycled FPGAs, making them less suitable for critical applications. As a result, it is essential to take steps to prevent the infiltration of recycled FPGAs in the IC supply chain and ensure the authenticity and reliability of FPGAs used in critical applications. Meanwhile, presently FPGAs are extensively used in autonomous applications such as UAVs and self-driving cars owing to the excellent performance of AI implementation in safety and critical applications [8]. If untrusted FPGAs infiltrate these mission-critical systems, the system's reliability may suffer, causing significant incidents.

Several supervised machine learning (ML) based methods for detecting recycled FPGAs have been proposed in [9-13]. These methods use a combination of data-driven techniques and statistical analysis to identify recycled FPGAs. For instance, some researchers have proposed using Support Vector Machines (SVM), Decision Trees, Random Forests, and Neural Networks to detect recycled FPGAs by analyzing their electrical characteristics such as power consumption, delay, and power supply noise. These methods have shown promising results in detecting recycled FPGAs with high accuracy. The key idea behind these methods is to employ the ring oscillator (RO) frequency to analyze the deterioration of circuit characteristics caused by the aging process. Fresh FPGA's RO frequencies are measured and used to train the supervised ML models. As these frequencies degrade with use, the trained model can determine whether the FPGA under test (FUT) is new or recycled. These supervised ML methods [9, 11, 12, 14] are predicated on the presence of known fresh FPGAs (KFFs). But a very large number of FPGA datasets have been required for the supervised ML methods for the accurate classification [9-10]. As the FPGA manufacturing technology is improving very rapidly, so this vast dataset collection is very difficult as well as costlier. Besides, for the preprocessing of the datasets and the training of the supervised ML models with these large datasets will require a very large amount of time and huge amount of memory for the whole process of the detection for every FPGA, training requires a significant number of measurements for the ROs with these supervised ML models which is very time-consuming. This is time consuming because the KFF datasets are not readily available and also the recycled datasets are unavailable to the researchers. But the overall performance is not up to the mark.

Due to the lesser accuracy, exhaustive fingerprint (X-FP) analysis method, based on another supervised ML model, has been proposed for better detection of recycled FPGAs in [12]. Here, the frequencies for all paths of look-up tables (LUT) in all combinational logic blocks (CLB) are taken into account. This method correctly detected the aging issues, means it has detected all the recycled FPGAs. However, the X-FP method can lead to a large number of frequencies, which in turn increases the dimensionality of the feature-vector of the ML model. This can severely degrade the accuracy of the model. This is a problem because in real-life scenarios, hundreds of thousands of ICs or FPGAs have to be tested in a unit time (such like within 24 hours assuming) otherwise the testing cost will be increased, so it is required to find other ways in order to reduce this extra-cost.

These methods are unrealistic approaches to get desired the accuracy. Because, for accurate classification through supervised ML, FPGA manufacturers require the measurement of a significant number of KFFs, but this large volume of KFFs is unavailable due to many factors such as rapid upgradation of technology, costing etc. These methods do not work properly to get desired level of accuracy if there are fewer KFFs available. Another thing is that the recycled FPGA datasets are not available beforehand, because there are so many third-party vendors who are distributing FPGAs, whether those are fresh or recycled, and it is impossible to collect the FP data of those for the training and testing purpose as the rapid advancement of technologies, process variations etc. In order to tackle these problems, unsupervised ML algorithms based methods have been proposed in [16-17]. Unsupervised ML algorithms doesn't require too many datasets for the training. These methods [16-17] used clustering algorithm (CA) which is the widely used unsupervised ML algorithm. This algorithm is one kind of anomaly or outlier detection scheme, which can be used to detect the recycled FPGAs as the anomalous data. As mentioned above, CA's most intriguing characteristics is that it doesn't require so many KFFs for the accurate classifications, and also it can be used without the negative-class data.

There are different types of clustering algorithms available, among which K-means++ is one. To address the limitations of supervised methods, previous works [16-17] proposed unsupervised methods for detecting recycled FPGAs. However, the classification accuracies of these methods are limited due to process variation in the KFFs, which use the measured frequencies as input vectors for the K-means++ method. Choosing the correct logic blocks for RO measurement is crucial, as selecting the wrong or inadequate ones can significantly reduce classification accuracy. To detect recycled FPGAs, previous approaches exhaustively compared the frequencies of neighboring blocks using direct density ratio estimation technique, which is a lengthy process. Moreover, these methods require a large dataset to increase accuracy, which is not always feasible in real-life scenarios and also requires a significant amount of memory for the huge amount of computations comparisons of those ICs or FPGAs which increases the testing-time and testing-cost. But it is required to achieve the desired accuracy faster and cheaper.

In brief it can be said that, the existing methods require a very large number of computations of neighboring RO FPs. And also, they require a large number of FPGA datasets in the supervised ML approaches which is impractical. As the technology is rapidly changing, the collection of these vast datasets not feasible. Meanwhile, the unsupervised methods don’t require a very number of datasets for the whole work, but existing unsupervised methods require a large number of calculations to achieve desired accuracy. So, there are scopes for finding some different approaches for achieving desired accuracy with lesser amount of computations.

1.3 Motivation

It is already mentioned that FPGA is an electronic component that is widely used in many applications due to its competitive performance and power benefits, as well as low non-recurring engineering costs. However, the use of the recycled components in counterfeit FPGAs has threatened the security and reliability of critical systems such as those used in airplanes, automobiles, and medical equipment etc. Several research works have been conducted to detect those recycled FPGAs, including using ML approaches based on RO delay information. However, these methods require a large number of computation measurements and are time-consuming and memory-expensive. An alternative method is the exhaustive fingerprint (X-FP) analysis, which takes into account the frequencies for all paths of look-up tables in all CLB. This method accurately detects aging issues but increases the dimensionality of the feature-vector of the ML model, which reduces accuracy. These issues have been addressed using with-in die (WID) modelling in literature [10]. However, accurate classification through supervised machine learning requires a large volume of known-fresh FPGAs (KFF), which may not be available due to various factors such as technology upgrades and cost. Also, the unsupervised ML methods addressed in [16-17] improves the accuracy but leaves one issue that is the huge comparison computations which leads to very time-consuming testing and costing. Recycled FPGAs exhibit distinct FP patterns compared to their fresh counterparts, which has been leveraged by previous researchers to differentiate between the two types in studies [16-17]. However, this approach requires extensive computation, and it is crucial to find a more efficient and cost-effective solution to address this issue.

This thesis has introduced a novel technique to reduce the computational complexities using the property of symmetricity of the structure of FPGAs that does not depend on KFFs or requires a low amount of KFFs [18]. Due to systematic process variation within the FPGA, the neighboring combinational logic blocks (CLBs) of FPGAs have similar or symmetrical array structures. This symmetrical property has been exploited for detecting recycled FPGAs using Clustering Algorithm (CA)-based anomaly or outlier detection scheme with K-means++ technique which analyzes the neighboring ring oscillator (RO) frequencies' symmetrical or similarity information. The proposed symmetry analysis method efficiently detects all the recycled FPGAs through outlier detection in a very short period of time with lesser computations compared to the previous unsupervised ML-based method. The proposed method eliminates the necessity of KFFs by exhaustively comparing all neighboring ROs, regardless of their frequency values [19-20]. This is because the assumption that the frequency distributions of neighboring columns ideally match due to the systematic component of process variation does not hold if there is any aging-induced degradation on either side [21]. As this method will utilize the symmetry analysis, the number of computations will be reduced which in turn will yield lesser time and lesser memory requirement. So, there are the scope of research in this area.

1.4 Objective with Specific Aims

The objective of this research is to develop a technique for improving the recycled FPGA detection performance using unsupervised machine learning approach. To achieve this goal this research will have the following aims:

i. To develop a CA-based unsupervised ML-model using with K-means++ method to solve the proposed research problem by exploring the symmetricity of the fingerprint (FP) data of the neighboring columns.

ii. To train the model using available FP data of the KFFs by finding the PDF values and anomaly scores of their neighboring columns.

iii. To test and verify the model by finding the best accuracy of that unsupervised K-means++ model.

1.5 Organization of the Report

In Chapter 1, the introductory information has been discussed. The common preliminaries used throughout the related works or topics and the fundamentals of the recycled FPGA detection are discussed in Chapter 2. The details of the proposed method have been discussed in Chapter 3. Simulation results and its discussions are provided in Chapter 4. Finally, the conclusions from this work along with the recommendations for the future works of this research are presented in Chapter 5.

Chapter 2
Preliminaries and Fundamentals of Recycled FPGA Detection

2.1 Introduction

In this chapter, recycled FPGAs and various terminologies related to detecting recycled FPGAs are presented as preliminaries. In the related terms section, the topics regarding this research work and its previous works in fingerprint analysis to detect recycled FPGAs will be summarized.

2.2 Counterfeit ICs

The global economic market has now reduced the cost of electronics due to a growing large horizontal business model that offers low-cost fabrications. As like ASIC, FPGA vendors similarly design and develop FPGA in their own lab, but fabricate them in offshore countries. This trend in the supply chain makes the backdoor for the corrupt market who instigate attacks like counterfeiting, malicious activities, or stealing of intellectual properties (IP) in real design etc. Specifically, the problem of counterfeiting of IC is now a major concern issue that drawn much attention to not only the media and industry but also government because of the global counterfeited market increasing exponentially over the past decades. Table 2.1 shows reports from 2021 of the five most commonly counterfeited electronic components. Among all incidents, the programmable logic IC is 8.3% of the counterfeited components [21].

The impact of counterfeited IC is more vulnerable in case of some critical applications like communication systems, medical equipment, aero-space etc. The

U.S. Department of Commerce reported over ten thousand occurrences relating to recycled ICs itself than other types of counterfeited components [22]. Moreover, according to statistical reports, FPGAs are among the top five most counterfeited electronic components. [23]. Yes, that's correct. With the increasing adoption of advanced technologies like IoT, Artificial Intelligence (AI), and Advanced Driver-Assistance System (ADAS), the demand for FPGAs has been on the rise. As a result, the global FPGA market is expected to grow rapidly and reach a value of USD 9.50 Billion in 2022. This growth can be attributed to the unique benefits that FPGAs offer, such as flexibility, low power consumption, and high performance, which make them ideal for use in real-time applications. [24]. Due to the increasing popularity of FPGAs, they have become an even more attractive target for counterfeiters, which raises concerns about their reliability for both government and industry stakeholders.

Table 2.1: Top-5 Most Counterfeited Semiconductors in 2021 [21]

Rank	**Commodity Type**	**% of Reported Incidents**
1	Analog IC	25.3
2	Microprocessor IC	13.4
3	Memory IC	13.1
4	Programmable logic IC	8.3
5	Transistor	7.6

2.2.1 Classification of Counterfeit ICs

The Semiconductor Industry Association (SIA) recommends that the best way to avoid counterfeit components is to purchase semiconductor products directly from the Original Component Manufacturer (OCM) or from authorized distributors or resellers. This ensures the authenticity of the components and helps to mitigate the risks associated with counterfeit components [25]. On the base of these points, the US Department of Commerce has marked some following points to classify a counterfeit component [22]:

1. Unauthorized copy: The component is not authorized or licensed by the original component manufacturer.
2. Non-conformance: The component does not meet the original design, model, and/or performance standards of the original component manufacturer.
3. Unauthorized production: The component is produced by unauthorized contractors or manufacturers, not by the original component manufacturer.
4. False representation: The component is misrepresented as new, working, or meeting specifications when it is actually off-specification, defective, or used. It may also have incorrect or false markings and/or documentation.

Fig. 2.1 shows a comprehensive classification of different types of counterfeiting components that are widely accepted in the community [26-27]. This expanded classification will help us understand the counterfeiting components more deeply and take potential techniques to measure and avoid counterfeited components in the supply chain.

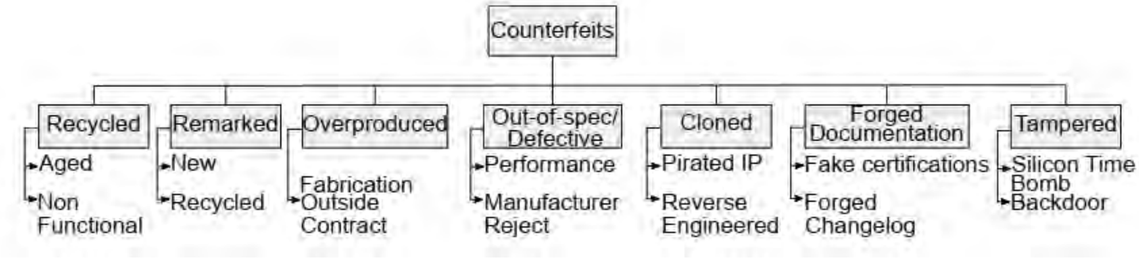


Fig. 2.1: Classification of counterfeit types [26-27]

A brief idea of each of counterfeit parts is given as follows:

1. Recycled: Refers to electronic components that were previously used and recovered from a system, then transferred to the supply chain as new components from the Original Component Manufacturer (OCM).
2. Remarked: Counterfeiters remove the original marking from a recycled IC and add fake information to uniquely identify it.
3. Overproduced: An untrusted foundry illegally accesses the IC layout and fabricates more ICs than authorized, transferring them into the supply chain.
4. Out-of-spec/Defective: ICs that failed post-manufacturing tests are sold into the supply chain instead of being destroyed.
5. Cloned: Counterfeiters reproduce ICs using reverse engineering or pirated intellectual property.
6. Forged Documentation: Counterfeiters may ship ICs without documentation and then forge fake documentation before transferring them into the supply chain.
7. Tampered: Malicious alterations or insertions, such as hardware Trojans, or external factors like high temperature, are used to decrease the security and reliability of ICs.

Recycled ICs are one of the most common types of counterfeit components in the electronics supply chain, and it has been reported that they make up more than 80% of all reported counterfeit components. This is due to the fact that recycled ICs can be more difficult to detect and identify compared to other types of counterfeits, as they may look like genuine components and have similar markings. [24]. In the next subsection, the recycled ICs will be discussed more elaborately.

2.2.2 Recycled ICs

Although recycled ICs show lower performance due to aging effects for their prior usage, the reliability issue will become more vulnerable as it undergoes very harsh environments during the recycling process. Initially, discarded printed circuit board (PCB) from different damaged or old systems is collected. The PCB is then heated to a specific high temperature so that soldering material begins to melt in order to collect used-ICs. Recycled ICs undergo a process of surface cleaning and polishing, following which a new set of markings such as PIN number, lot number, manufacturer logo, country of manufacture, etc. are printed on the surface. This gives the recycled ICs a fresh appearance, and they are then introduced into the supply chain as new components. [58].

Despite having good functionality initially, the performance of recycled ICs deteriorates rapidly due to the aging effects caused by their previous use. Furthermore, the recycling process subjects them to extreme electrical, mechanical, and temperature stresses, which contribute to a significant decrease in their performance. However, preventing the entry of recycled ICs into the IC market is challenging due to the increasingly complex supply chain. Therefore, it is crucial to have an effective method for detecting recycled ICs, especially recycled FPGAs.

2.3 Field Programmable Gate Array (FPGA)

The FPGAs are reconfigurable ICs where any logic function can be performed by appropriately configuring logic elements. Unlike the ASICs, fabricated for a specific application, FPGAs can be reprogrammed for any desired applications. This versatility makes FPGAs more popular day-by-day. Specific hardware description languages (HDL) like Verilog HDL or VHDL (Very High-Speed Integrated Circuit HDL) can be used to configure the logic function on its own field; thus, it is called field-programmable. CLBs and complex routing interconnection make it possible to implement complex logic functions inside any part of the FPGAs. Presently, most of the FPGAs are SRAM-based and it is used to store information.

2.3.1 Basic Structure of FPGAs

Fig. 2.2 depicts the basic structure of an FPGA. In an FPGA, the CLB (Configurable Logic Block), connection block, and switch block are arranged to form an array. The CLB is composed of one or more clusters of basic logic blocks (LB) that are interconnected within the cluster. Each LB is made up of several LUTs (Look-Up Tables), a carry chain, and flip-flops as shown in Fig. 2.2.

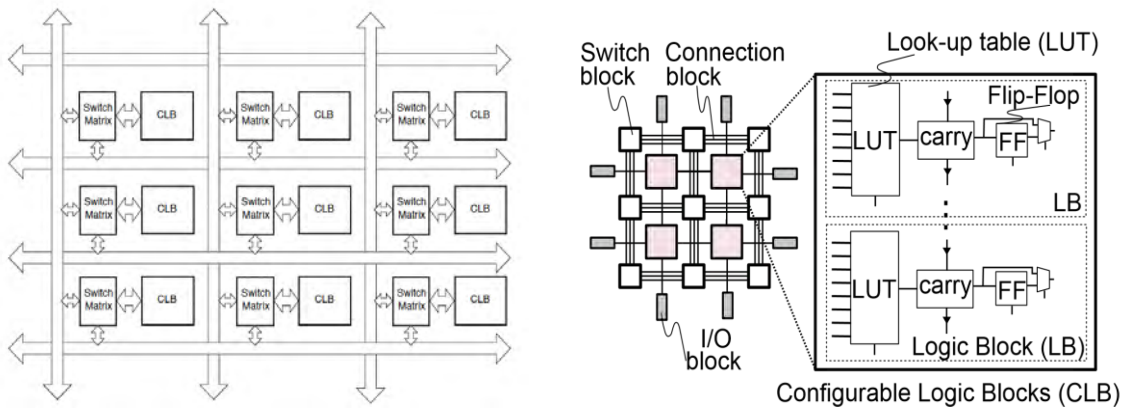


Fig. 2.2: Typical structure of an FPGA

An LUT, or Lookup Table, is a fundamental logic element that defines a function in an FPGA. It is capable of implementing an arbitrary i-input Boolean function through a chain of multiplexers (MUX), with n being the number of inputs. The input of the LUT is selected to utilize the value stored in a memory element with 2^i bits as the input of the truth table for the function. In modern SRAM-based FPGAs, these memory element entries are represented by configuration bits stored in SRAM cells.

Fig. 2.3 shows a 3-input LUT can be implemented using pass transistors or transmission gates. The inputs of the LUT are connected to the SRAM cells which store the configuration bits. Each input is connected to two pass transistors or transmission gates, and the gates of these transistors are connected to the corresponding configuration bit. The output of each pass transistor or transmission gate is connected to the input of a buffer. The output of the buffer is the output of the LUT. When the LUT is configured with a particular set of values, the pass transistors or transmission gates corresponding to those values are turned on, allowing the values to pass through to the output buffer [31]. The FPGA architecture consists of an array of CLBs, connection blocks, and switch blocks. Each CLB contains one or more clusters of basic logic blocks (LBs), which are made up of multiple LUTs, carry chains, and flip-flops. LUTs are the basic logic elements that determine the function of the FPGA. In modern SRAM-based FPGAs, each LUT is typically implemented as an n-input LUT using configuration bits stored in SRAM cells.

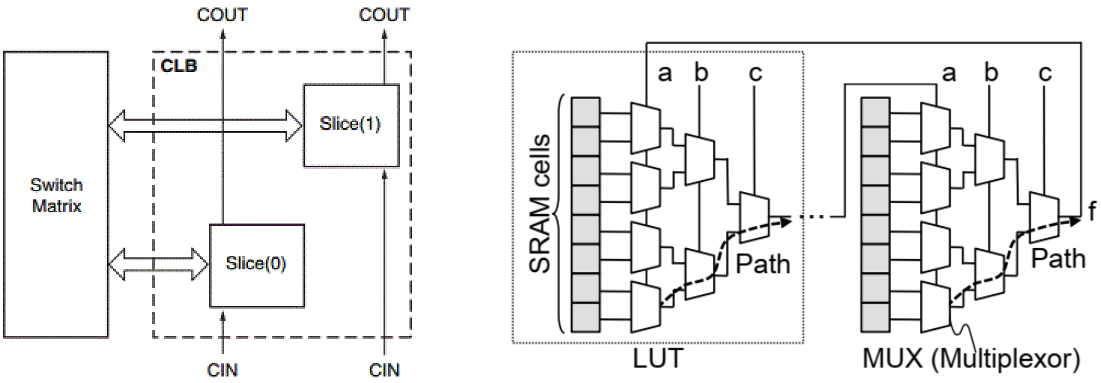


Fig. 2.3: Basic idea of implementation of RO in CLBs using LUTs in the FPGA

A 3-input LUT can be designed with pass transistors or transmission gates to transfer the value from SRAM cells to the output. This LUT requires eight SRAM bits to set the truth table value for any 3-input Boolean function. The direction of the LUT path can be changed by altering the input values of the LUTs. The FF logic circuit enables FPGAs to implement sequential circuits by storing the output value obtained from the corresponding LUTs at each clock cycle. The CLBs are connected to the switch boxes via programmable interconnection blocks.

In order to implement specific logic functions, the signals between the CLBs and the I/O blocks are routed through the programmable interconnects, which are controlled by the switch box. In this way, the interconnects can be configured to create specific logic elements. For the purpose of detecting recycled FPGAs, this study uses a RO (Ring Oscillator) logic circuit, which collects frequency information from each CLB and uses it as a measure of aging.

2.3.2 Applications of FPGAs

FPGAs have a wide range of applications due to their reprogrammable and customizable nature. They are often used in low volume and high complexity projects where flexibility and performance are critical. Some of the main applications of FPGAs include medical and scientific equipment, video and image processing, telecommunications and data communications, aerospace and defense, and server and cloud computing. In medical and scientific applications, FPGAs are used in equipment such as MRI and CT scanners, as well as in research equipment for simulations and data analysis. In video and image processing, FPGAs are used for real-time video processing and image recognition. In telecommunications and data communications, FPGAs are used in network routers and switches, and for encryption and decryption. In aerospace and defense, FPGAs are used in radar and sonar systems, as well as in navigation and guidance systems. In server and cloud computing, FPGAs are used for acceleration of specialized workloads such as machine learning and big data processing [36]. Fig. 2.4 shows some important applications of FPGAs.

FPGAs are commonly used in both wired and wireless communication systems. In wired communications, FPGAs are used in applications such as serial backplanes, network switches and routers, and high-performance computing systems. In wireless communications, FPGAs are used for networking solutions and to address standards such as WiMAX, 5G/6G, and HSDPA. FPGAs are also used in the infrastructure side of communication systems to process and analyze data at high speeds [36].

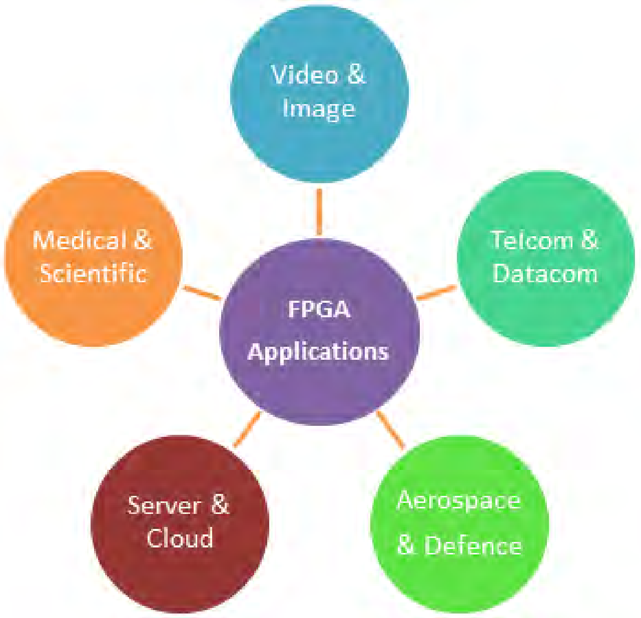


Fig. 2.4: Applications of FPGAs

FPGA chips find applications in medical equipment for processing data and serving diagnostic and monitoring purposes [36].

FPGA chips also find extensive applications in the aerospace and defense industries, where they are utilized for image processing, generating waveforms, and for enabling partial reconfigurations in software-defined radios (SDRs) [36].

FPGA technology presents an option for ASIC companies to quickly prototype and test ideas and concepts without undergoing a lengthy process. This is helpful in improving time to market of various technological products and reducing engineering costs in several processes such as industrial automation and surveillance [36].

The reconfigurability of FPGA technology makes it an attractive option for reducing the long-term maintenance costs of a system. This flexibility also enables FPGA to keep up with modifications and changes, further reducing costs associated with system updates. Even developers at Microsoft have access to FPGA chips, and they use open source tools like the Microsoft Cognitive Toolkit. Microsoft utilizes Intel FPGA chips to increase their use of AI in their operations.

FPGAs have emerged as a key technology for the development of deep neural networks (DNNs), which are the foundation of artificially intelligent systems. When compared to GPUs, high-performance FPGAs can be even more beneficial in certain applications, making them the preferred choice for developing machine learning technology [36].

The recent acquisitions of Altera by Intel and Xilinx by AMD demonstrate the increasing importance of FPGAs in the server and computing market. This market segment is expected to show significant growth for the FPGA industry [36].

2.3.3 FPGA Fingerprinting using ROs

Ring oscillators (ROs) are commonly used in the semiconductor industry for process control and characterization. They are simple to design and can provide valuable information about process variations, such as timing delays and device performance, which can impact the overall functionality of integrated circuits (ICs). ROs can also be used for frequency testing and calibration of various electronic systems. [32-33]. In addition, ROs are also extensively used in FPGAs for delay variation to use it as fingerprint in recycled FPGA detection [34-35]. The output of a ring oscillator (RO) is a periodic waveform with a frequency that depends on the delay of the inverters in the ring. A ring oscillator typically consists of an odd number of inverters (such as 3, 5, or 7) connected in a chain to form a closed loop, as you mentioned. When the output of the last inverter feeds back to the input of the first inverter, the circuit can oscillate at a certain frequency determined by the delay of each inverter. The oscillation frequency is typically measured and used to extract information about the manufacturing process variation and delay characterization in ICs. Fig. 2.5 depicts a 7-stage inverter-based RO with an enable logic showing oscillation.

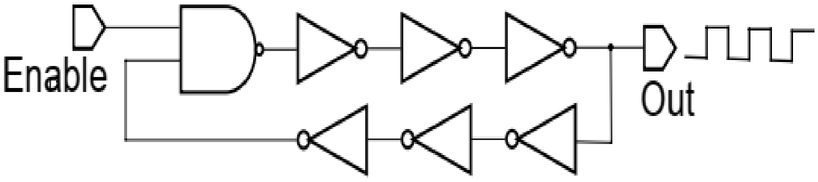


Fig. 2.5: 7-stage ring oscillator (RO) with enable signal

2.4 Exhaustive Fingerprint Analysis (X-FP)

A new technique called X-FP has been introduced to enhance the efficiency of detecting recycled FPGAs. [12]. Using advanced RO design, the method X-FP can thoroughly analyze the aging deterioration of all paths in LUTs of all the CLBs, fully characterizing their frequencies. Results from experiments conducted on 10 commercially available FPGAs demonstrate that X-FP can effectively capture the degradation effects with high accuracy.

The main contributions of this method can be outlined as follows:

- To improve the detection of recycled FPGAs and effectively observe aging degradation, the proposed method utilizes X-FP. This technique enables the examination of the aging-induced delay characteristics of all paths in all LUTs of all the CLBs.

- The X-FP characterization of 10 commercial Xilinx Artix-7 FPGAs, including aging acceleration of 3 FPGAs, revealed various path differences between fresh and aged FPGAs, as demonstrated in the experiments.

Fig. 2.6 shows the RO implementation within a single CLB for the capturing of the FP values of all paths in an FPGA. As the LUTs in the Xilinx Artix-7 are made of 6-input, thus there are total $2^{6-1} = 32$ paths for a single FPGA. And every FPGA contains 3,173 CLBs, thus total 3,173 ROs were placed on a geometrical grid of 33×120 (except the empty space of the layout) using Verilog HDL script. With the X-FP analysis, it becomes possible to observe the aging degradation of all paths in all LUTs within the FPGA. However, although X-FP analysis accurately captures aging information of recycled FPGAs, it raises two potential issues with ML-based detection. Firstly, to characterize the X-FP, a larger number of RO measurements are required which increases the testing cost considerably. Secondly, the X-FP technique generates a significant amount of measurement data, which cannot be effectively handled by typical machine learning algorithms for the purpose of detecting recycled FPGAs [12].

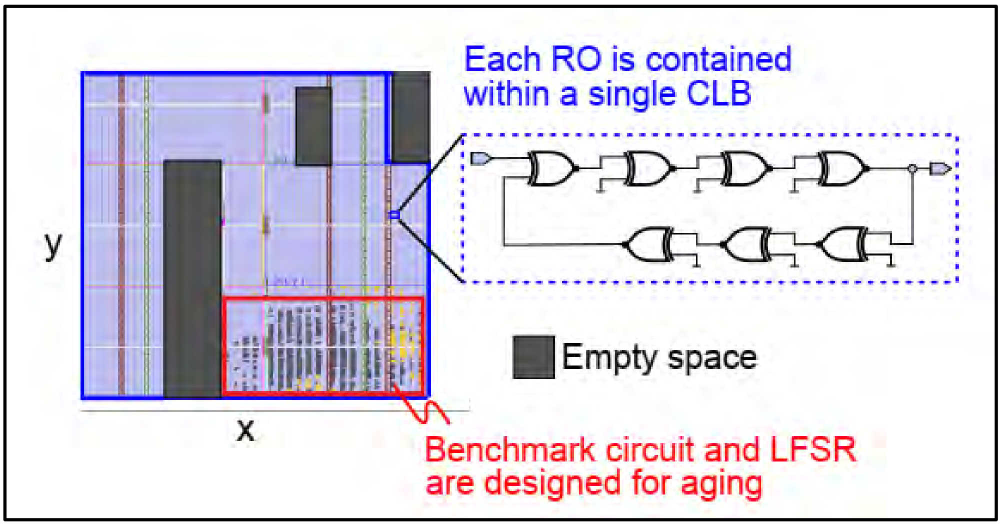

Each RO is contained within a single CLB
y
x
▇ Empty space
Benchmark circuit and LFSR are designed for aging

Fig. 2.6: An array of ROs in the Xilinx Artix-7 FPGA

2.5 Virtual Probe (VP) Technique

The main concept behind the virtual probe (VP) technique [41] is to strategically place and measure a small number of test structures at specific locations on a wafer or chip. The parametric variations at other locations are not directly measured during hardware testing. Instead, virtual probes are virtually placed at these locations to predict the variation information using a numerical algorithm, as shown in Fig. 2.7. In contrast to the conventional approach that involves monitoring variability at many locations using numerous test structures, the virtual probe technique proposes to monitor variability at only a few specific locations and then employ an intelligent algorithm to predict the complete spatial variation accurately. This is made possible by leveraging the sparse structure in the spatial frequency domain.

Fig. 2.7 illustrates an example of the virtual probe technique. The left side shows the conventional approach, where a large number of test structures are deployed and measured to completely characterize process variations. On the right side, the virtual probe technique proposes to deploy and measure only a few test structures, while virtual probes are conceptually added to recover the spatial variation using a numerical algorithm.

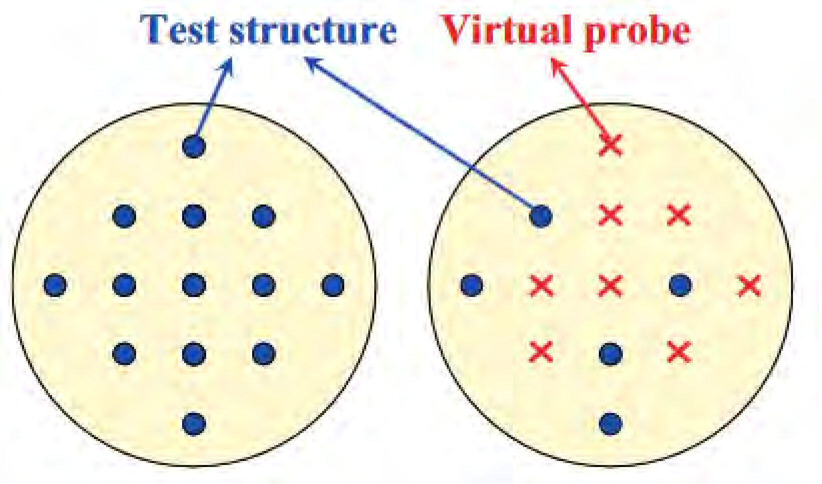


Fig. 2.7: Virtual Probe Technique

To summarize, the virtual probe technique provides several key advantages over traditional techniques, such as [41]:

1. **Cost-effectiveness:** VP minimizes the number of required test structures, which reduces the cost of testing and measurements, such as area overhead, testing/characterization time, and yield loss during testing.
2. **High accuracy:** VP can accurately reconstruct the spatial variation with a probability close to 1. The accuracy can be verified in real time using efficient techniques such as cross-validation and Bayesian inference. Additional sampling points can be collected to further improve accuracy until the prediction error is sufficiently small.
3. **Versatility:** VP can predict the spatial pattern of both inter-die and spatially-correlated intra-die variations. The prediction is based on the measurement data from the current wafer/chip only.

The virtual probe (VP) technique has a wide range of potential applications in various fields beyond integrated circuits. It can be used in semiconductor manufacturing for testing and characterizing process variations, as well as in design verification and optimization. Additionally, VP can be applied in other fields such as biotechnology, environmental monitoring, and material science, where spatial variability measurements are critical. In summary, the versatility of the VP method makes it a valuable tool in various applications that require spatial variability measurements. Some important of them are listed below [41]:

1. Wafer-level Silicon Characterization
2. Chip-level Silicon Characterization
3. Speed-binning of the manufactured chips to determine their maximum operation frequency
4. Post-Silicon tuning technique to measure the presence of large-scale process variation

2.6 Clustering Analysis

Clustering, also known as clustering analysis, is a process of organizing a collection of objects into groups or clusters, where objects within a cluster are more similar to each other than to objects in other clusters. Clustering is an essential component of exploratory data analysis and a widely used technique for statistical data analysis. It finds its application in several fields, including but not limited to pattern recognition, image analysis and classification, information retrieval, bioinformatics, data compression and processing, information encoding and decoding, computer graphics, and machine learning (ML) [44]. Clustering analysis is not a single algorithm, but rather a task that involves grouping objects together based on their similarities in a specific way. There are many algorithms available to perform clustering, each with their own understanding of what makes up a cluster and how to find them efficiently. Clusters can be defined as groups with small distances between members, dense areas in the data space, particular statistical distributions, or other criteria. Clustering can be thought of as a multi-objective optimization problem, with different algorithms and parameter settings suited to different datasets and intended uses. Cluster analysis is an iterative process that involves knowledge discovery and interactive optimization, often requiring adjustments to data preprocessing and model parameters until desired properties are achieved. [44].

2.6.1 Cluster Analysis Algorithms

There have been approximately 100 types of clustering algorithms published so far, though not all provide models for their clusters and cannot be easily categorized. There is no objectively "correct" clustering algorithm, and the most appropriate one for a particular problem often needs to be chosen experimentally. Clustering is subjective and dependent on the individual's perspective. One cluster model may work well for a particular dataset, while another model may fail. Therefore, it is important to carefully consider the problem at hand and experiment with various algorithms until a satisfactory solution is achieved.

A list of some of the clustering algorithms is as follows: [45]

1. BFR algorithm
2. Canopy clustering algorithm
3. Cluster-weighted modeling
4. DBSCAN
5. K-means clustering
6. K-means++
7. K-medians clustering
8. Nearest-neighbor chain algorithm

K-means++ clustering algorithm has been chosen for this research purpose because of its advantages such as it is faster and provides a better performance. Not all clustering algorithms have their models or library codes because most of them are still theoretical, and also some of them are being used in the industries in recent time, and K-means++ has most advantages and least disadvantages in contrast to others. That’s why it has been chosen for this research work.

2.6.2 K-means++

K-means++ is a widely used clustering algorithm that aims to partition a given set of observations into k clusters, where each observation is assigned to the cluster with the nearest mean. This algorithm is used for vector quantization, originally from signal processing, and is commonly used in data mining for choosing the initial values or "seeds" for the K-means clustering algorithm. The algorithm was proposed by David Arthur and Sergei Vassilvitskii in 2007 as an approximation algorithm for the K-means problem, and it addresses the limitations of the standard

K-means algorithm that sometimes results in poor clusterings [46]. In Fig. 2.8, the basic idea of K-means++ scheme of clustering algorithm has been shown.

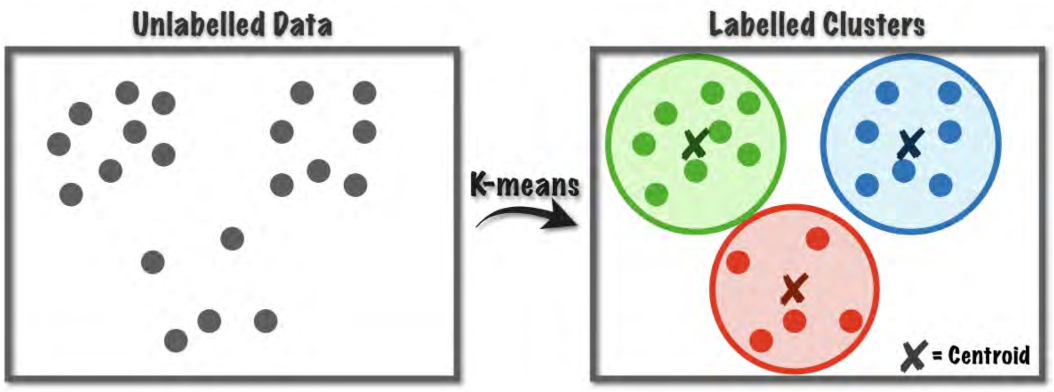


Fig. 2.8: K-means++ clustering algorithm

The standard approach to finding an approximate solution to the K-means problem is widely used due to its efficiency in finding reasonable solutions quickly. The algorithm works by randomly selecting k initial cluster centers, then assigning each data point to the nearest cluster center, and finally computing new cluster centers based on the mean of the data points assigned to each cluster. This process iterates until convergence, that is, until the cluster centers no longer change or a maximum number of iterations is reached.

To elaborate, the K-means++ algorithm starts by selecting a single data point as the first cluster center randomly from the given data set. Then, the algorithm selects the next cluster center from the remaining data points in such a way that the probability of choosing a data point as the next center is proportional to its squared distance from the closest existing center. This approach ensures that the new cluster centers are well separated from each other and have a high chance of representing different regions of the data. The remaining cluster centers are selected using the same probabilistic approach until k centers have been chosen. Finally, the standard K-means optimization iterations are performed starting from these initial cluster centers to obtain the final cluster assignments. This initialization step often results in better clustering results than the random initialization used by the standard K-means algorithm.

By choosing the initial centers in a way that spreads them out across the data space, K-means++ aims to avoid getting stuck in suboptimal solutions that can occur when the initial centers are too close together or too far apart. The idea is to increase the chances of finding good starting points for the K-means algorithm that are representative of the overall data structure. This is achieved by selecting centers that are far from each other and from previously chosen centers, which is accomplished by assigning higher probabilities to data points that are farther from the nearest center [46].

2.6.3 Applications of K-means++

K-means++ has been widely applied since its initial proposal. According to a review by Shindler [47], which covers various types of clustering algorithms, the K-means++ approach successfully overcomes some of the problems associated with other methods of defining initial cluster centers for K-means clustering. Lee et al. [48] used K-means++ to create geographical clusters of photographs based on latitude and longitude information attached to the photos. Howard and Johansen reported an application of K-means++ to financial diversification. Ongoing discussions and support for the method can also be found online.

2.7 Anomaly Detection

Anomaly detection is an important task in various fields, including data mining, machine learning, and computer security. It is used to detect unusual or suspicious behavior, which can be indicative of fraud, errors, or attacks. Anomalies can be detected by comparing data points to a statistical model of normal behavior, or by using unsupervised learning techniques to identify data points that are significantly different from the majority of the data. Anomaly detection has applications in many areas, including fraud detection in finance, intrusion detection in computer networks, and fault detection in industrial processes [50].

Anomalies or outliers may represent critical events or rare occurrences that are of particular interest and value to the analyst. For example, in fraud detection, detecting an unusual pattern of financial transactions could help uncover fraudulent activity. Similarly, in medical diagnosis, identifying unusual symptoms or test results can aid in the detection of rare diseases or disorders. In these cases, anomaly detection techniques can be used to identify and highlight these rare or unusual events, allowing analysts to investigate further and take appropriate actions [50].

Anomaly detection is a critical task in many real-world applications where detecting rare and unusual events can provide valuable insights and prevent potentially dangerous or costly situations. For example, in cyber security, detecting anomalous network traffic patterns can help identify potential threats and prevent cyber-attacks. In the medical field, anomaly detection can help diagnose diseases by identifying abnormal patterns in medical images or patient data. In finance, detecting unusual patterns in financial transactions can help prevent fraud and financial crimes.

Anomaly detection can be performed using various techniques, including statistical methods, machine learning algorithms, and deep learning techniques. Statistical methods such as the Z-score or Mahalanobis distance are commonly used to detect anomalies based on the deviation from the mean or normal distribution. Machine learning algorithms such as k-nearest neighbors (k-NN) or support vector machines (SVM) can also be used to detect anomalies based on the distance from neighboring points or the separation of classes.

In recent years, deep learning techniques such as autoencoders and generative adversarial networks (GANs) have shown promising results in anomaly detection. Autoencoders can learn to reconstruct input data and detect anomalies based on the reconstruction error, while GANs can generate synthetic data that mimics the real data distribution and detect anomalies based on the difference between the real and generated data.

Despite the growing interest in anomaly detection, it remains a challenging task due to the inherent difficulty in defining what constitutes an anomaly and the high variability and complexity of real-world data. Anomaly detection algorithms often require careful tuning and domain-specific knowledge to achieve satisfactory results. However, with the increasing availability of large and diverse data sets and the development of more advanced algorithms, anomaly detection is becoming an increasingly important and powerful tool in many fields.

There are three main categories of techniques used in anomaly detection:

1. Supervised anomaly detection
2. Semi-supervised anomaly detection
3. Unsupervised anomaly detection

2.7.1 Anomaly scores

Anomaly detection techniques involve identifying data points that are significantly different from the majority of the data. One approach involves developing a model of the normal behavior of the data and marking any data points that fall outside of the predicted range as anomalies. To provide a clear understanding of the results, an anomaly score is typically calculated for each time interval. According to [16], lower anomaly scores indicate positive-class or accepted data, while higher (even very high) scores indicate negative-class or not-acceptable data.

2.8 Probability Density Function (PDF)

Probability distributions are a fundamental concept in probability theory and statistics. They are used to describe and analyze various phenomena in the real world, such as the distribution of heights or weights in a population, the frequency of certain types of weather events, or the likelihood of various outcomes in a game of chance. Different types of probability distributions are used to model different kinds of phenomena, depending on the characteristics of the data and the research question of interest. In Fig. 2.9, the PDF curve for the normal distribution case has been shown.

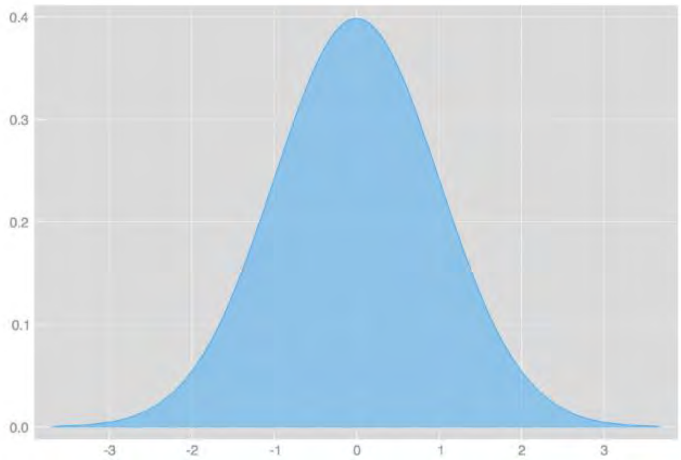


Fig. 2.9: PDF curve of Normal Distribution

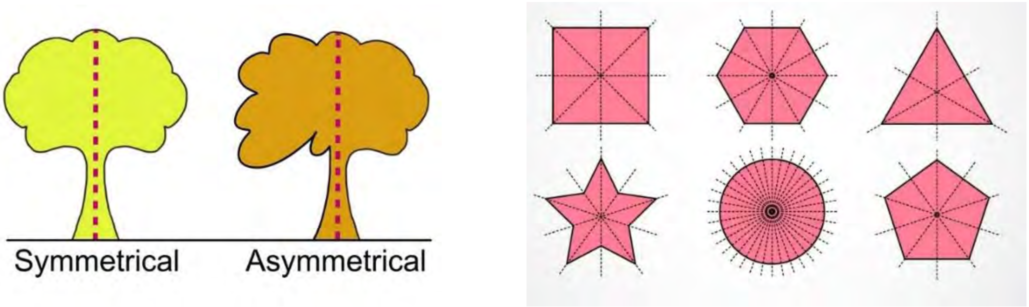
The normal distribution is a very common probability distribution that is widely used in statistics and many other fields. It is also known as the Gaussian distribution, after the mathematician Carl Friedrich Gauss who first described it. The normal distribution has a bell-shaped curve, with the mean, median, and mode all being equal and located at the center of the curve. Many natural phenomena, such as measurements of physical properties, tend to follow a normal distribution, which makes it a useful tool for modeling and analysis. The general form of its probability density function (PDF) is,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \qquad \text{.........................(2.1)}$$

In eq^{n}. 2.1, The parameter μ represents the mean or expected value of the distribution, and it also serves as its median and mode. On the other hand, the parameter σ corresponds to the distribution's standard deviation. The variance of the distribution is σ^2.

2.9 Symmetry Analysis

Symmetry has a more precise definition in mathematics than in everyday language. It refers to an object that remains unchanged under certain transformations, such as translation, reflection, rotation, or scaling. In other words, a symmetric object looks the same before and after the transformation. Symmetry can be observed in various ways, including with respect to time, spatial relationships, geometric and other functional transformations, as well as in abstract objects like models, language, and music. Asymmetry, on the other hand, refers to the absence or violation of symmetry [49].



(a) (b)

Fig. 2.10: (a) Difference between symmetrical and asymmetrical shapes. (b) Some other symmetrical shapes

In this work, symmetry analysis or symmetrical neighborhood means the FP data of the KFFs have symmetry because the FPGAs have the similar or symmetrical type of architecture. But this will not be held true for the recycled or aged FPGAs because after starting of the using of the FPGAs, their fingerprint will be changed. This indicates that, for the fresh FPGAs, there will be symmetricity, but for aged or recycled FPGAs these FP will be changed and thus there will be no symmetricity. Thus, if this symmetric property can be analyzed then the recycled FPGAs can be detected. This theory is explained below in the Fig. 2.11.

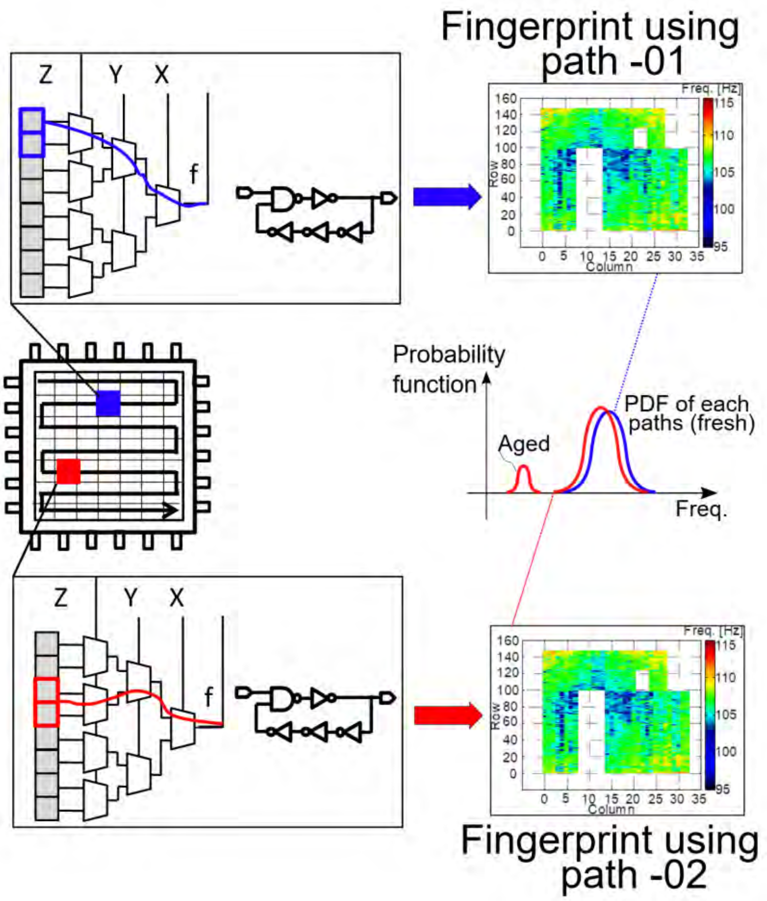


Fig. 2.11: Key idea of the Symmetry Analysis

In Fig. 2.11, fingerprints for two different paths of the same KFF have been analyzed, and it has been shown as the high values of the PDFs in Red and Blue curves. Both of them are having similar types of curves with almost in the same region. They are equivalent but not fully same due to the process variation of the IC chips. And again this process has been done with that FPGA but making it artificially aged or recycled, and after analyzing it’s PDF in small red curve in the left-side, we see that fresh and recycled FPGAs have different types of PDF curves which can be used to analyze the binary classification of fresh or recycled FPGAs.

2.10 uLSIF Method

The least squares (LS) method is a type of mathematical regression analysis utilized to determine the line of best fit for a given set of data, which can visually illustrate the relationship between the data points. The data points represent the connection between a known independent variable and an unknown dependent variable [51]. In this study, the uLSIF (unconstrained Least-Squares Importance Fitting) method was employed, which is a variant of the LS method. The uLSIF-based approach to anomaly detection is computationally efficient, enabling it to be applied to large-scale datasets.

2.11 Receiver Operating Characteristic (ROC) Curve

An ROC curve is a visualization tool that displays the performance of a binary classifier system across various discrimination thresholds. The name "ROC curve" originated from its use in military RADAR receiver operations during World War II. The curve is constructed by plotting the true positive rate (TPR) against the false positive rate (FPR) at different threshold values. TPR is also known as sensitivity, recall, or probability of detection, while FPR is also known as probability of false alarm. A true positive (TP) indicates a correct detection of the presence of a condition or characteristic, while a false positive (FP) indicates an incorrect detection of a condition or characteristic that is not present. In Fig.2.12, the ROC curve has been shown for more clarity.

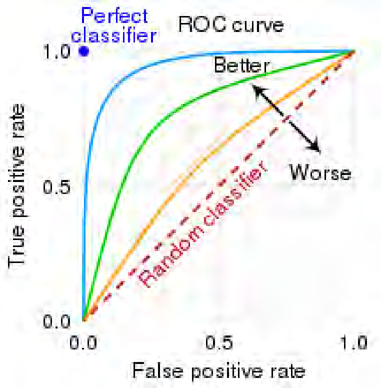


Fig. 2.12: ROC curve

ROC curves are commonly used in medical research, machine learning, and other fields where binary classification is important. In medical research, for example, ROC curves are often used to evaluate the accuracy of diagnostic tests. In machine learning, ROC curves are used to evaluate the performance of binary classifiers, such as support vector machines (SVM) or neural networks.

A perfect classifier would have a TPR of 1 and an FPR of 0, meaning it would correctly identify all positive cases and never mistakenly identify a negative case as positive. In reality, however, most classifiers have limitations, and ROC curves help to illustrate the trade-off between sensitivity and specificity.

One way to summarize the diagnostic ability of a binary classifier using the ROC curve is to calculate the area under the curve (AUC). The AUC provides a single number that represents the overall performance of the classifier. An AUC of 1 indicates perfect discrimination, while an AUC of 0.5 indicates a classifier that is no better than random guessing.

ROC curves are a useful tool for selecting a threshold that balances sensitivity and specificity based on the needs of the application. For example, in medical research, a test with high sensitivity may be preferred if a false negative result could be life-threatening, while a test with high specificity may be preferred if false positives would lead to unnecessary treatments or procedures.

2.11.1 Area Under the Curve (AUC)

It is worth noting that the area under the curve (AUC) can have various interpretations and applications in different fields. For example, in economics and finance, the area under the demand curve can represent the total revenue generated by a product, while in probability theory, the area under the probability density function curve represents the total probability of an event occurring. In addition, the area under the ROC curve, as mentioned earlier, is a common evaluation metric for binary classifiers in machine learning.

Overall, the concept of area under the curve and its calculation through integration is a fundamental concept in calculus and has many practical applications in different fields.

Finding the area between a curve and a line involves similar steps, with the addition of finding the point(s) of intersection between the curve and the line. Once these points are found, the limits of integration can be set accordingly. The integral is then taken between the limits, with the absolute value taken if the curve dips below the line.

Finding the area between two curves involves finding the points of intersection between the curves and setting the limits of integration accordingly. The integral is then taken between the limits, with the difference between the integrals of the upper and lower curves taken to obtain the area between them.

It should be noted that the process of integration can be quite complex for some curves, especially those that do not have a simple equation. In such cases, numerical methods, such as the trapezoidal rule or Simpson's rule, can be used to approximate the area under the curve.

The concept of area under the curve is widely used in various fields such as physics, engineering, economics, and finance. It is often used to calculate the total value or quantity of a variable over a given time period or range of values. For example, the area under a velocity-time graph gives the total distance traveled by an object over a given time period.

Overall, the calculation of area under the curve is a fundamental concept in calculus and has many practical applications in various fields. In Fig.2.13, the AUC calculation process has been shown for more clarity.

$$\text{Area} = \int_a^b \mathrm{y.\,dx} = \int_a^b \mathrm{f(x).\,dx} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots(2.2)$$

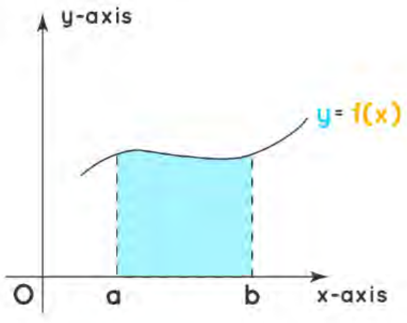


Fig. 2.13: AUC calculation

2.12 Understanding IC Cost

The complexity of ICs has grown tremendously over the years, and this has led to debates and discrepancies in calculating the final cost of an IC. As technology advances at a rapid pace, chip designers have to keep up with the changes and advancements in order to accurately estimate the IC cost. In the past, silicon die size used to be the dominant factor in calculating the cost of an IC. However, this is no longer the case, as there are now numerous other components and factors that play an equally important role in determining the final cost.

It is no longer enough to simply focus on the silicon die size when estimating IC costs. Other factors, such as the number of layers in the chip, the complexity of the design, the type and amount of memory, the manufacturing process, and the packaging, all play a significant role in determining the final cost. This complexity has led to a variety of methods for calculating the cost of an IC, which can sometimes lead to differing opinions on the final cost [54].

Despite the challenges in determining the final cost of an IC, experts have developed equations and models to estimate the cost. While these models can be helpful, they are not foolproof and can sometimes lead to inaccuracies. As technology continues to evolve and ICs become even more complex, it is likely that debates over the final cost of an IC will continue. Experts have noted that there is a very simple equation one can use in order to determine the final chip cost:

Final IC cost = package cost + test cost + die cost + Shipping cost

In addition, it's important to consider the scale of production when calculating IC costs. The cost per unit of an IC can be significantly reduced with large-scale production, as the fixed costs of setting up the manufacturing process can be spread over a greater number of units. This is known as the "economy of scale" and can have a significant impact on the final cost of the IC.

Another factor to consider is the complexity of the IC design. More complex designs may require specialized manufacturing processes or materials, which can increase the cost of production. On the other hand, a simple design may require less testing and verification, which can lower the overall cost.

Ultimately, calculating the cost of an IC is a complex process that requires consideration of multiple factors. However, with a clear understanding of the key components that impact cost and early analysis of the project, it's possible to make an accurate estimate and keep costs under control. [54]

Chapter 3
Symmetry Analysis based Recycled FPGA Detection

3.1 Introduction

In this chapter, the details of the proposed method for the detection of recycled FPGAs will be discussed including its experimental setup and workflows. The required hardware and software will be mentioned at the last of this chapter.

3.2 Methodology of the Work

The proposed methodology will be divided into the following possible stages:

- Collection of FP data for different FPGAs (KFFs) etc.
- Jupyter Notebook platform on online-GPU which is of Google Colab will be used for designing, preprocessing, training, testing and validating of the unsupervised CA-based K-means++ model for anomaly detection of the FPGAs.
- Numpy, Pandas, Scikit-learn, Sci-Pi etc. library modules of Python programming will be used to implement this model.
- After the collection of datasets, the preprocessing and training of those dataset will be performed on online-GPU as mentioned above.
- Then test set will be used to evaluate the performance of the trained network. Appropriate weights and other results will be saved which will give better performance.
- The system will then be simulated for measuring accuracy, anomaly scores and other performance metrics to detect the FUT as fresh or recycled based on their anomaly scores.
- Then the performance of the proposed model will be compared with that of other researchers.

3.3 Dataset Collection and Details of the Fingerprint Values

The datasets of 10 known fresh FPGAs or KFFs used in this work have been collected from **Dependable System Lab** at Nara Institute of Science and Technology (NAIST), Japan. All of them were designed with RO circuits with 2-input XOR & XNOR gates with 32 paths (16 paths for both XOR and XNOR gate-based RO circuits). For each path, there data are measured similarly for all 10 KFFs and then 3 of them are artificially accelerated aged FPGAs.

In the experimental study of the proposed method, a total of 13 FPGAs were utilized, comprising of 10 newly acquired ones and 3 that had already been in use, to showcase the precision of the method. The FP (heatmap) values are ranging from 90.0 to 140.0 MHz for the 33×120 dimension (=3,960) values in which there about 787 positions are NaN values (Not a Number) or empty place because there RO circuits couldn't be placed. There are pre-set or installed hardware circuits for FPGAs such as multiplier circuit and others in those empty places. So, total 3,960 – 787 = 3,173 ROs can be placed in the FPGA. These fingerprint values were transformed into the heatmaps using **gnuplot** software. Now, some of those heatmaps of the fresh FPGAs are shown in the Fig. 3.1. Aged FPGAs have also same type of heatmaps but with higher values than that of their fresh counterparts. There are total 13 FPGAs × 32 paths = 416 fingerprints. All these fingerprint datasets are stored in .csv (Comma Separated Values) format files. A sample snippet of the datasets has been shown in the Appendix I.

3.4 Recycled FPGA Detection using Symmetric Path Analysis

In this work, the aim is to reduce the number of computations of the comparison for the anomaly score by the symmetry analysis. The proposed method performs the RO measurement and the frequency comparisons in the unsupervised recycled

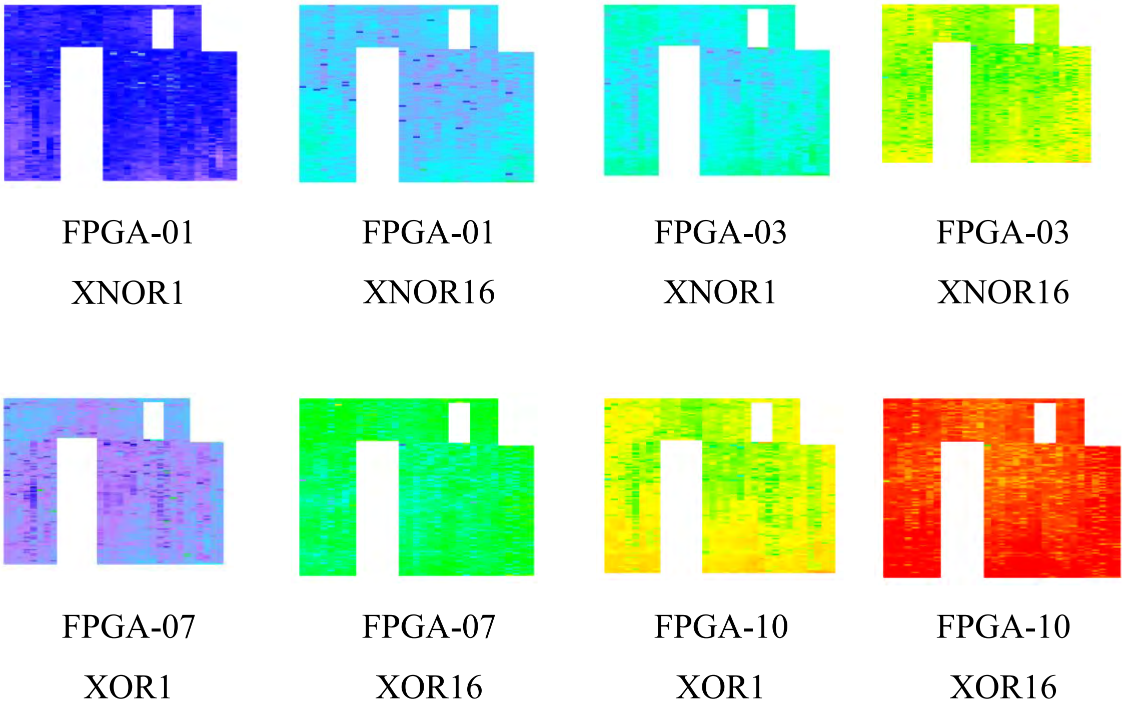


Fig. 3.1: Heatmaps of some paths of some fresh FPGAs

FPGA detection. This method includes an additional step that analyzes the symmetry among the different FPs to get the best match for minimizing the number of comparisons. The score for detecting anomalies is determined exclusively from the symmetrical FPs. The X-FP measurement technique is employed to conduct RO measurements for all the LUT paths of every CLB. To classify the data, a self-referencing outlier detection approach is devised, utilizing the unconstrained least squares importance fitting (uLSIF) algorithm.

X-FP measurement is carried out for every LUT path across all CLBs within the FPGA, thoroughly capturing the impact of aging. If a LUT contains P paths, the total number of X-FPs can be denoted as $F = F_1, F_2, ..., F_p$, where F_p is the X-FP of the p-th path and each FP contains n number of RO measurements. For finding the symmetry among the X-FPs, a Virtual Probe (VP)-based X-FP estimation have been utilized by using various sample frequencies. The root-mean-square error

(RMSE) is computed by comparing the estimated X-FP with the measured X-FP. The symmetry path fingerprints are then obtained based on the RMSE values. For instance, the RMSE value of F_1 and F_2 are very similar, so they are considered as symmetry path fingerprints. In this proposed method, the comparisons are performed only on the symmetry path fingerprints. The proposed method detects recycled FPGAs by assessing the anomaly score determined through the uLSIF algorithm. To compute the anomaly score, the RO frequencies of the X-FPs are denoted as $F_p = f_{p;1}, f_{p;2}, ..., f_{p;n}$, where $f_{p;n}$ is the RO frequency of the n-th RO in the p-th path, and are compared with symmetry path FP based on the RMSE value of the estimated X-FP using the VP technique. To compute the anomaly scores, the uLSIF algorithm is provided with two vectors of X-FPs for the symmetrical path FPs, denoted as F and F'. The key advantage of this method is that it does not necessitate comparing all possible combinations. If there are C columns in each X-FP, then the total number of comparisons required is C × P/2, whereas in prior works [10], a total of (C − 1) × P comparisons were needed.

3.5 Implementation of uLSIF Method

If there are two data samples x_1 and x_2 from which the probability distribution functions are p(x) and q(x) respectively, then, estimated density ratio function,

$$w(x) = \frac{p(x)}{q(x)} \qquad \text{............................(3.1)}$$

There is a Python package called ***densratio*** which provides a function densratio() which computes the density ratio function w(x), and it implies the value of anomaly score.

This package is based upon RuLSIF method which is also called the α-relative density ratio,

$$w(x) = \frac{p(x)}{\alpha .p(x) + (1 - \alpha) .q(x)} \qquad \text{............................(3.2)}$$

where α is in the range [0, 1].

If α is 0, this reduces to the ordinary density ratio w(x) like eq^{n}. (3.1) which provides the anomaly scores. In this way, uLSIF with Python has been implemented in this work.

3.6 Experimental Setup

To validate the efficacy of the proposed method, experiments were conducted using the Xilinx Artix-7 FPGA (XC7A35T-ICPG236C) produced using 28 nm process technology. The experimental datasets used in this study were obtained from the DS Lab of NAIST, Japan, and were used for simulation analysis. A total of 10 FPGAs (FPGA-1 to FPGA-10) were utilized, in addition to 3 artificially aged or recycled FPGAs (FPGA-1a to FPGA-3a), to showcase the results of the experiment. It should be noted that these 10 FPGAs were manufactured in different production lots and obtained from various distributors at different times. Since the proposed method depends on the systematic component of process variation, it is crucial to evaluate FPGAs with diverse process variations from different lots.

The schedule for the three FPGAs was designed to reflect the actual stress and recovery phases. The stress phases were limited to 24 hours (one day) only. Fingerprint measurements were conducted solely at room temperature after the 5-day recovery phase. The FPGA states at the start and end of the recovery phases during fingerprint measurement were referred to as the "stress state" and the "recovery state," respectively. In the ML learning detection, only 5 days of recovery measurement data were utilized to simulate a real-life scenario. Once again, it should be noted that these aging processes were carried out by researchers at the DS Lab of NAIST, Japan.

3.7 Algorithm of the Proposed Method

The step-by-step algorithm of the proposed method is mentioned below:

(i) The data collected from DS Lab of Japan has been preprocessed before feeding into the program algorithm. Preprocessing includes data-cleaning, renaming the filenames etc. Here the X-FP data of the KFFs have been collected.

(ii) Find the PDF (Probability Density Function) values from the column frequencies of each of the paths of each FPGAs.

(iii) If the adjacent columns of each path have 120 numbers of data then they are fed into the uLSIF method to find the anomaly scores based on their density ratios. To maintain the symmetricity and similar amount of data for each comparison, only the column FP data which have 120 values were chosen.

(iv) The maximum anomaly scores were stored for the detection of recycled FPGAs.

(v) Those maximum valued anomaly scores were fed into the K-means++ algorithm using SciKit-Learn library of Python language to find the binary-clusters, 0 means fresh and 1 means aged/recycled.

(vi) Finally plotting the ROC-AUC curve, and calculating of the accuracy.

3.8 Flowchart of the Proposed Method

Fig. 3.2 shows the flowchart of the program code of the proposed method.

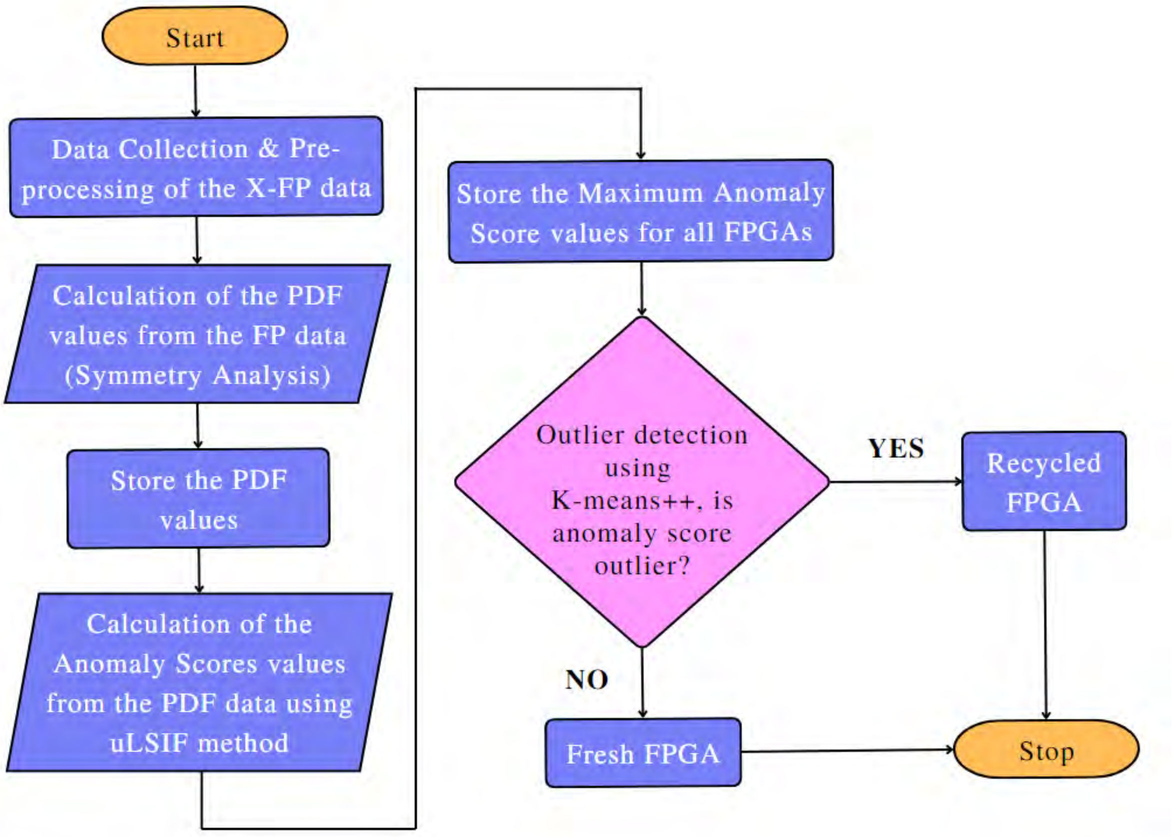

Start
Data Collection & Pre-processing of the X-FP data
Calculation of the PDF values from the FP data (Symmetry Analysis)
Store the PDF values
Calculation of the Anomaly Scores values from the PDF data using uLSIF method
Store the Maximum Anomaly Score values for all FPGAs
Outlier detection using K-means++, is anomaly score outlier?
YES
Recycled FPGA
NO
Fresh FPGA
Stop

Fig. 3.2: Flowchart of the Program Code of the Proposed Method

3.9 Required Hardware and Software

To implement the uLSIF and K-means++ clustering algorithms for the detection of the anomalies, **Jupyter Notebook** with **Python** Development Environment using **Google Colab or Colaboratory** cloud-platform has been used. The reasons for using this platform are:

1. ML requires huge calculations with very long time with CPU, but Google provides online GPU which can reduce this time requirement by 10-times.

2. GPU costs very high, but due to Google this cost did not occurred.
3. No need to install any software on the local PC neither to install locally the libraries of **Python** language

So, hardware used in this work are:

1. Asus Notebook, Intel core i3 processor of 1.8 GHz, 8 GB RAM
2. Google's online-GPU available on Google Colab

And the software required in this work are:

1. Python 3.8 on Google's online-GPU
2. gnuPlot

Chapter 4
Results and Discussion

4.1 Introduction

This chapter shows the results achieved from the experiments using the model as developed in this thesis. Then it has been compared with that of other researches followed by necessary discussion.

4.2 Anomaly Score Computation Details

A sample picture of the column frequencies of each path known as heatmap of each FPGA has been shown in Fig. 4.1. Sample snippet of the datasets is presented in

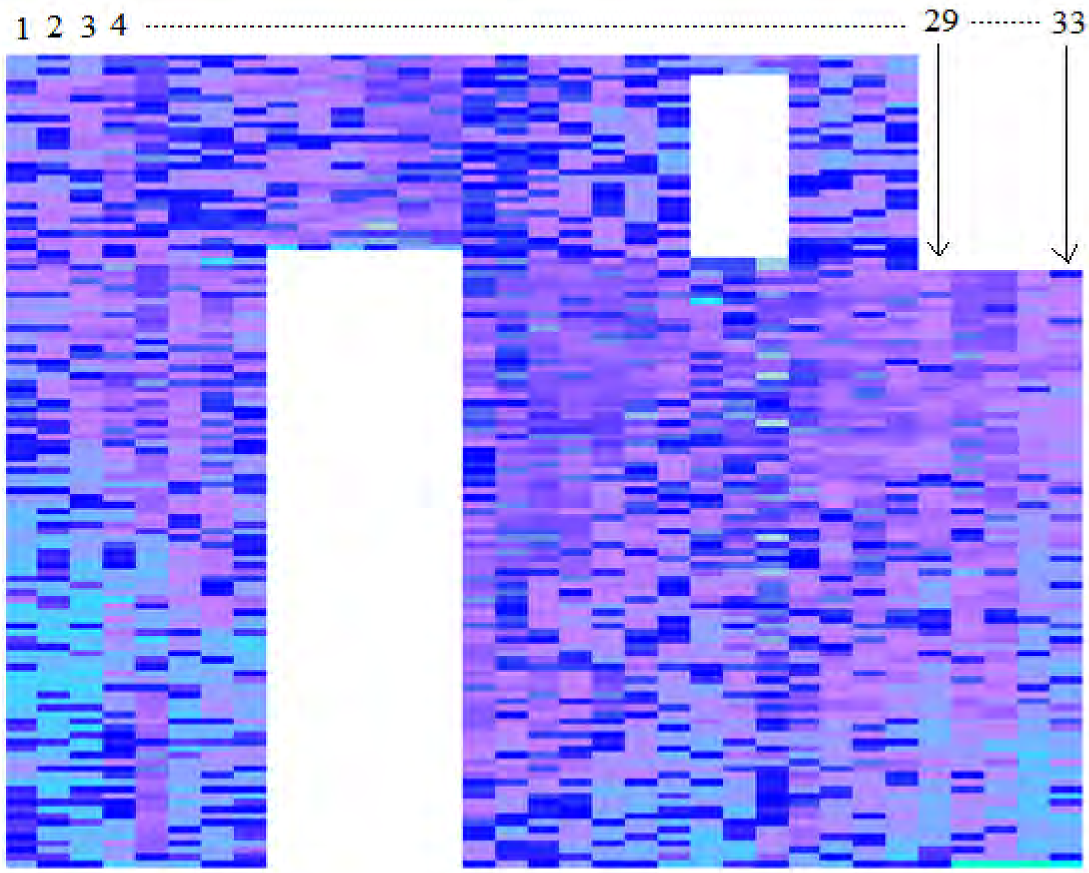


Fig. 4.1: A sample heatmap with 120 rows × 33 columns

Appendix I. At first, for FPGA-01, XNOR01 and XOR01 paths (path 01 & 17 respectively) have been selected for column frequencies comparison. For both paths, the frequency values of adjacent columns, for example columns 1&2, 2&3, 3&4 etc. were at first compared or checked whether they both have 120 values or not. If they have 120 values, then their Mean, Standard Deviation and PDF values were calculated using the Sci-Py library of Python language. These PDF values were then given input to the uLSIF method to find the density ratios and then the anomaly scores for those paths were computed. In this way, 19 acceptable anomaly scores out of 32 comparisons have been found for XNOR01 and XOR01 paths (path 01 & 17 respectively) of FPGA-01. Only the maximum anomaly score from those 19 values for XNOR01 and XOR01 paths of FPGA-01 has been stored for the training and testing phase to detect the recycled FPGAs using K-means++ algorithm.

This process is then repeated for FPGA-01's XNOR02 and XOR02 paths (path 02 & 18 respectively), then FPGA-01's XNOR03 and XOR03 paths (path 03 & 19 respectively) and so on for ten(10) fresh FPGAs (FPGA-01 to FPGA-10) and three(3) recycled/aged (accelerated aged) FPGAs (FPGA-01 to FPGA-03).

4.3 Measurement Results

At first, the RMSE data of 10 fresh FPGAs is shown in Fig. 4.2 at different paths of the X-FP. This RMSE data is obtained from the estimated X-FP over the actual measurement of each X-FP. From this figure, the symmetry among various paths is clearly observed. For example, the RMSEs of path-15 and path-31 of FPGA-01 are 3.20 MHz and 3.17 MHz, respectively. As the RMSE values of path-15 and path-31 are very similar, they could be considered as symmetry paths and these symmetry paths are used for the comparison (CP) in determining the anomaly

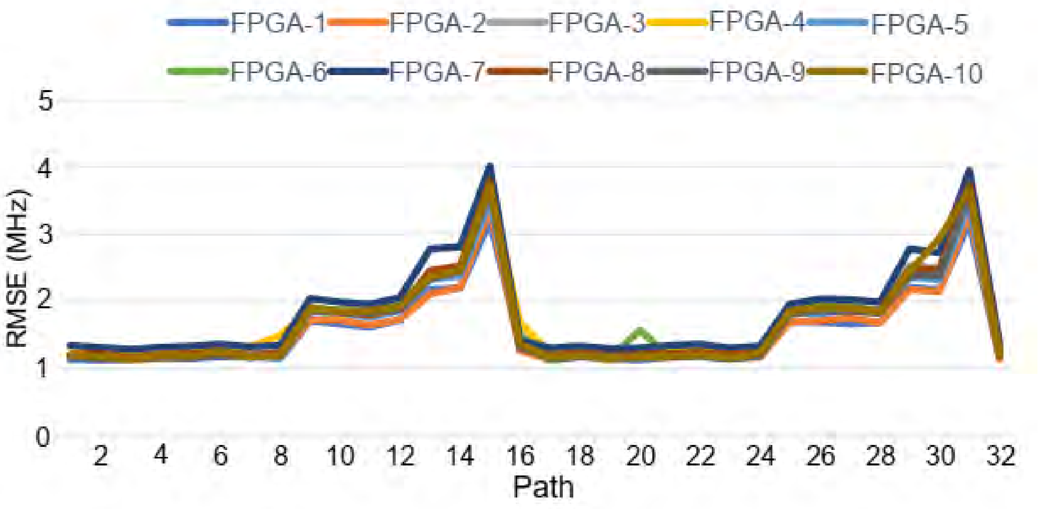


Fig. 4.2: RMSE of 10 fresh FPGAs samples data.

scores. In Fig. 4.2, a slight peak is found at path=20 for FPGA-06 because of the systematic process variation. Thus, based on these RMSE values of 10 fresh FPGAs at 32 paths, the 16 symmetry CPs are considered in this study.

Table 4.1: Comparison Table of Different Paths for the Anomaly Scores

CP no. Path	**CP1** 1, 17	**CP2** 2, 18	**CP3** 3, 19	**CP4** 4, 20	**CP5** 5, 21	**CP6** 6, 22	**CP7** 7, 23	**CP8** 8, 24
CP no. Path	**CP9** 9, 25	**CP10** 10, 26	**CP11** 11, 27	**CP12** 12, 28	**CP13** 13, 29	**CP14** 14, 30	**CP15** 15, 31	**CP16** 16, 32

The 16 CPs of different symmetry paths are shown in Table 4.1. Based on the CP values shown in Table 4.1, the anomaly scores using the proposed method. Fig. 4.3 presents the anomaly scores of 10 fresh FPGAs (FPGA-01 to FPGA-10) and 3 aged FPGAs (FPGA-01 to FPGA-03). The vertical axis shows the anomaly scores at 16 different CPs. From Fig. 4.4, it has been observed that, in most cases the anomaly score of the aged FPGAs is higher than the fresh FPGAs. For instance,

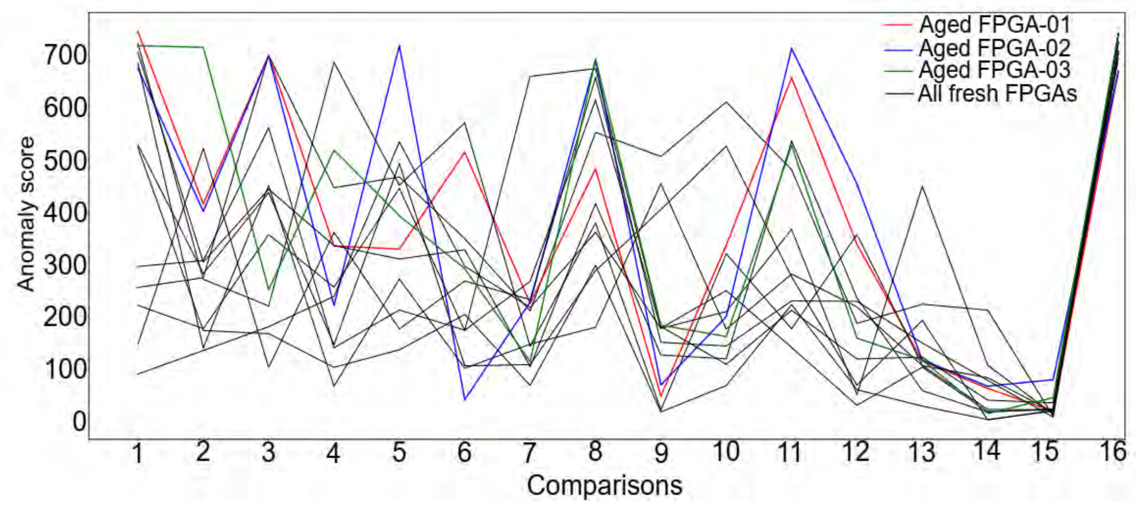


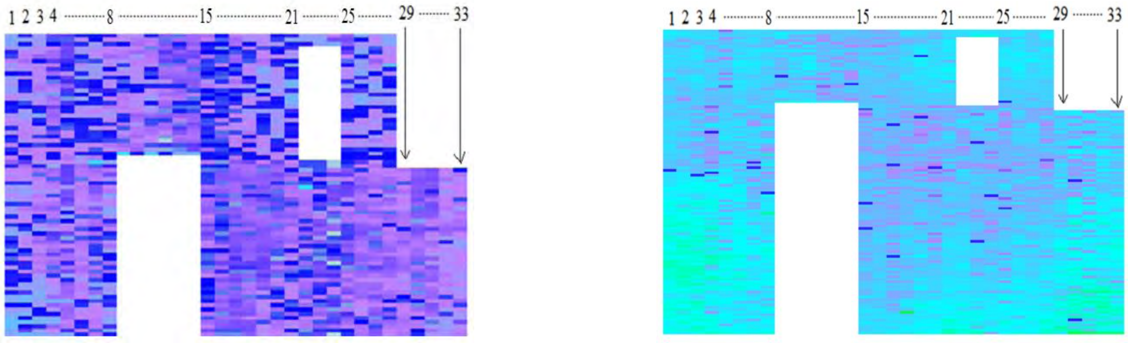
Fig. 4.3: Maximum anomaly scores of the 16 CPs using the proposed approach where used 10 fresh FPGAs and three aged FPGAs

aged FPGA-01 (red color) and FPGA-03 (green color) are found high anomaly scores in three CPs (1, 2, and 11) and four CPs (1, 2, 8, and 11), respectively. There are few cases found when fresh FPGAs have shown higher anomaly values because of the process variation, buying from different manufacturers at different times etc. Since high anomaly scores were observed in the aged FPGAs at various critical paths (CPs), it is anticipated that the unsupervised ML model can accurately detect the aged FPGAs.

4.4 Calculation of Comparison Computations

To demonstrate the difference of the computation numbers in this proposed method with respect to other unsupervised methods, how the calculation of those previous method’s process of the comparison computations has been done is shown which is as follows:

For each FP data, the computations done as follows: For PDF calculations, column data of (1,2), (2,3), (3,4), (4,5), (5,6), (6,7), (7,8), (15,16), (16,17), (17,18), (18,19), (19,20), (20,21), (25,26), (26,27), & (27,28) are used and fed into the **densratio** library package of Python. Here total 16 comparisons have been done for all the 32 paths of a single FPGA. The sample heatmaps are shown in Fig. 4.4. Here one thing is to be noted that, each FPGA has total 32 paths. 16 of them named as XNOR1, XNOR2,…..,XNOR16, and other 16s are named as XOR1, XOR2,….,XOR16. These paths are named in this way because the ROs are designed with the XNOR and XOR logic gates.



(a) (b)

Fig. 4.4: Sample heatmaps with XNOR1 circuit in (a) and XOR1 in (b)

So, total **calc1** = 16 × 32 = 512 comparison calculations per FPGA

And, in this proposed method, the calculation of the comparison computations are as follows:

For each FPGA, the computations has been done on the FP data for similar or symmetric paths of (XNOR, XOR) with the columns (1,1), (2,2), (3,3), (4,4), (5,5), (6,6), (7,7), (8,8), (15,15), (16,16), (17,17), (18,18), (19,19), (20,20), (21,21), (25,25), (26,26), (27,27), & (28,28). And these column data will be fed into the **densratio** package. Here total 19 comparisons have been done for all 16 similar paths of a single FPGA.

So, total **calc2** = 19 × 16 = 304 comparison calculations per FPGA

And the difference,

$$d = \frac{\text{calc1} - \text{calc2}}{\text{calc1}} \times 100\ \%$$

$$= \frac{512 - 304}{512} \times 100\ \% = \text{around 41 \% less calculations}$$

4.5 Recycled FPGA Detection and Comparison

The proposed method for recycled FPGA detection was evaluated by analyzing the anomaly score obtained from the symmetry critical paths. The results were presented using a receiver operating characteristics (ROC) curve in Fig. 4.5 to visualize the classification outcomes. The "Recycled FPGA detection ratio" in the figure refers to the true positive rate, while the "Misclassification ratio of fresh FPGAs" represents the false positive rate. The best performance is indicated by the upper left corner of the ROC curve. Based on the results presented in Fig. 4.5, it can be observed that the proposed method was successful in detecting all aged FPGAs in all cases, as indicated by the high true positive rate. However, in one

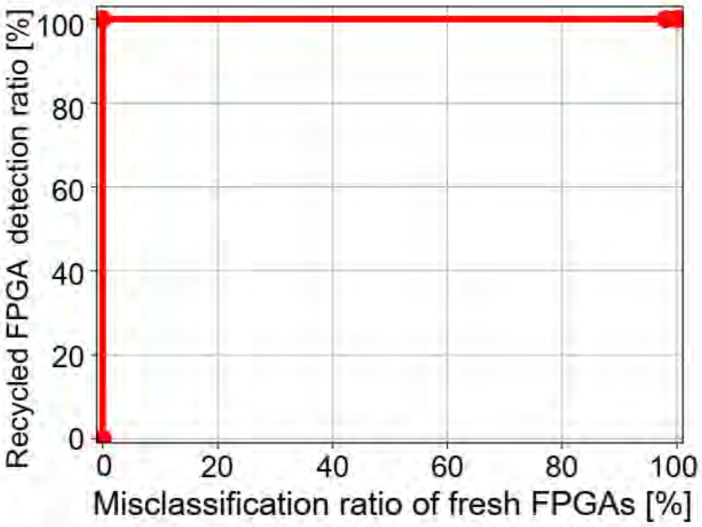


Fig. 4.5: ROC curve of the proposed recycled FPGA detection method using 10 fresh FPGAs and 3 aged FPGAs

instance, the fresh FPGA-03 was misclassified due to the effects of process variation, leading to a lower value for fresh FPGA detection. Overall, it can be concluded that the unsupervised ML algorithm effectively detected recycled FPGAs using the proposed technique.

The FPGA detection results are as follows:

Table 4.2: Detection Results of the FPGAs

Sl. No.	**FPGA No.**	**Output**	**actual_label**	**K-means++**	**K-means++ label**
1.	FPGA-01	0	fresh	0	fresh
2.	FPGA-02	0	fresh	0	fresh
3.	**FPGA-03**	**0**	**fresh**	**1**	**aged**
4.	FPGA-04	0	fresh	0	fresh
5.	FPGA-05	0	fresh	0	fresh
6.	FPGA-06	0	fresh	0	fresh
7.	FPGA-07	0	fresh	0	fresh
8.	FPGA-08	0	fresh	0	fresh
9.	FPGA-09	1	aged	1	aged
10.	FPGA-10	1	aged	1	aged
11.	FPGA-03a	1	aged	1	aged
12.	FPGA-04a	1	aged	1	aged
13.	FPGA-05a	1	aged	1	aged

It is showing that, 12 out of 13 FPGAs are correctly detected by this proposed method. Here ‘a’ means artificially or accelerated aged. Thus, for k = 2 clusters,

$$\text{K-means++ Accuracy} = \frac{\text{No.of correct detections}}{\text{Total no.of samples}} \times 100\ \%$$

$$= \frac{12}{13} \times 100\ \% \quad = 92.31\ \%$$

Table 4.3: Comparison of the Proposed Method with Other Research Results

Method	**Fresh FPGAs**	**Aged FPGAs**	**ML Algorithm**	**Accuracy %**	**Computations per FPGA**
Ref. [16]	10	2 (6h)	K-means++	95	512
Ref. [17]	35	9 (6h)	K-means++	100	512
Proposed Method	10	3 (24h)	K-means++	92.31	304 (41% less)

Table 4.2 shows the comparison of the proposed method with the previous works on detecting the recycled FPGAs using different ML algorithms. Most of the time, amount of KFFs are very low, and the unsupervised methods provide better result with lesser KFFs. Though some unsupervised methods require either large volume of KFFs or vast calculations of the fingerprint analysis of the RO frequencies to achieve better result, but the proposed method achieves an almost similar accuracy using around 41% fewer computations for each FPGA. As the proposed method require less calculations, thus there will be lesser time require to test the ICs or FPGAs with this method. Also, the memory requirement will be lesser too. Though the proposed method didn't yet achieve 100% accuracy as like the [17], but that method requires more than 3 times KFFs for getting that result. Practically, such a large amount of KFFs are not available for testing in the industry level because it will then increase the testing cost, testing time and memory requirement for the FUTs. So, overall it can be said that the proposed method lowers the testing cost, testing time by around 41%, and also the memory requirement.

4.6 Discussion

In this work, to detect the recycled FPGAs, a novel mechanism has been used. As the FPGAs have the symmetrical structures in their design, so those symmetry will be in the neighboring columns, and thus their FP values will be similar. In order to find those symmetries, the same columns (with maximum number of values which is 120) of symmetrical paths of an FPGA have been chosen to find the PDF values. Those values then fed into the densratio package of Python to get the anomaly scores of those paths. For the fresh FPGAs, those anomaly score should be low values, and for the aged or recycled FPGAs it will be higher or large numbers. This is shown in Fig. 4.4. In this work, the similar results of anomaly scores have been found and then those will be used to determine whether the FPGAs are fresh or recycled by using the unsupervised ML algorithm K-means++.

It has been shown in Fig. 4.5 that the all the aged/recycled FPGAs were detected correctly in this proposed method with around 41% less computation and around 92% accuracy in overall, which is really a desired outcome of this work. This is desired because in the real-life scenarios, there are hundreds of thousands of ICs have to be checked with as minimum as possible time required, and if there are recycled ICs or FPGAs which were not checked before releasing into the market, then those companies will lose their reputation and many valuable worth. But if these computations can be done with lesser time, as this work showed, then the time-to-market will be reduced and thus the goodwill of those companies will be increased.

This novel method has shown that the around 41% less computations has been reduced which gives around 92% accuracy. The accuracy can be increased by tweaking the different parameters of the K-means++.

Chapter 5 Conclusion

5.1 Conclusion

As the IC supply chain continues to expand, recycled FPGA poses a significant threat not only to IC manufacturing companies, but it also leads to vulnerabilities in mission-critical applications. This research work adequately investigated and addressed the problems of the existing methods in detecting recycled FPGA by introducing symmetry analysis. The proposed detection method is based on exhaustive fingerprinting, which involves collecting aging information from all paths in all LUTs of the FPGA. This approach enables comprehensive analysis and detection of any anomalies or changes caused by aging, resulting in more accurate and effective detection of recycled FPGAs. The prospective advantage of this proposed recycled FPGAs detection method includes a lesser amount of testing-time requirement as well as the lesser amount memory.

In this work, a novel method using symmetry analysis has been proposed for unsupervised detection of recycled FPGAs between X-FPs with the K-means++ algorithm. The method calculates the anomaly score by estimating the direct density from the symmetry comparisons. It uses the K-means++ clustering algorithm to detect recycled FPGAs from the fresh samples. The proposed method is able to detect all aged FPGAs with 92% accuracy and with 41% less computational effort than the previous method. This symmetry analysis based proposed method makes it a faster and more cost-effective way of detecting recycled FPGAs with less memory usage.

5.2 Recommendations for Future Works

Some of the recommendations regarding the future works based upon this proposed method can be stated as follow:

1. While this method has been demonstrated using FPGAs, it can also be extended to other types of integrated circuits (IC) for hardware security.
2. Accuracy can potentially be increased and time-complexity or computations can potentially be reduced by selecting a different and smaller number of symmetric fingerprint (FP) patterns.

References

[1] M. Tehranipoor, H. Salmani, and X. Zhang. “Integrated Circuit Authentication: Hardware Trojans and Counterfeit Detection’, Springer International Publications, (2014).

[2] M. M. Tehranipoor, U. Guin, and D. Forte, “Counterfeit integrated circuits”, in *Counterfeit Integrated Circuits*, pages 15–36. Springer (2015).

[3] H. Sharma, J. Park, D. Mahajan, E. Amaro, J. K. Kim, C. Shao, A. Mishra, and H. Esmaeilzadeh, “From high-level deep neural models to FPGAs”, in *Proceedings of IEEE/ACM International Symposium on Microarchitecture*, pages 1–12, (2016).

[4] P. Wadhwani, P. Saha, “Field Programmable Gate Array (FPGA) Market Size Forecast-2028”, [Online: https://www.gminsights.com/industry-analysis/field-programmable-gate-array-fpga-market-size]

[5] U. Guin, K. Huang, D. DiMase, M. Tehranipoor, and Y. Makris, “Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain”, Proceedings of the IEEE, 102(8):1207–1228, (2014).

[6] D. Akhoundov, “Counterfeit componets incident report”, (2017). [Online: https://www.erai.com/erai_blog/3139/_2017_erai_reported_parts_analysis]

[7] H. Dogan, D. Forte, and M. M. Tehranipoor, “Aging analysis for recycled FPGA detection”, in *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 171–176, (2014).

[8] F. Ahmed *et al*., “A survey on UAV computing platforms: A hardware reliability perspective,” *Sensors*, vol. 22, no. 16, p. 6286, (2022).

[9] H. Dogan *et al*., “Aging analysis for recycled FPGA detection,” in *Proc. DFT*, pp. 171–176, (2014).

[10] F. Ahmed *et al.,* “Feature engineering for recycled FPGA detection based on WID variation modeling,” in *Proc. ETS,* (2019).

[11] V. Jyothi *et al.*, “Fingerprinting field programmable gate arrays,” in *Proc. ICCD*, pp. 337–340, (2017).

[12] F. Ahmed *et al.,* “Accurate recycled FPGA detection using an exhaustive-fingerprinting technique assisted by WID process variation modeling,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 8, pp. 1626–1639, (2021).

[13] M. M. Alam *et al.,* “Recycled FPGA detection using exhaustive LUT path delay characterization and voltage scaling,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 27, no. 12, pp. 2897–2910, (2019).

[14] F. Ahmed et al., “Low cost recycled FPGA detection using virtual probe technique,” in IEEE International Test Conference in Asia (ITC-Asia), 2019, pp. 103–108, (2019).

[15] D. Arthur *et al.*, “K-means++: The advantages of careful seeding,” Stanford, Tech. Rep., (2006).

[16] Y. Isaka *et al.*, “Unsupervised recycled FPGA detection based on direct density ratio estimation,” in *IEEE 27th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2021, pp. 1–6., (2021).

[17] Y. Isaka, M. Shintani, F. Ahmed and M. Inoue, "Systematic Unsupervised Recycled Field-Programmable Gate Array Detection," in *IEEE Transactions on Device and Materials Reliability*, vol. 22, no. 2, pp. 154-163, (2022).

[18] Jin, W., Tung, A.K.H., Han, J., Wang, Ranking Outliers Using Symmetric Neighborhood Relationship. In: Ng, WK., Kitsuregawa, M., Li, J., Chang, K. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD Lecture Notes in Computer Science, vol. 3918. Springer, Berlin, (2006).

[19] S. Ohkawa, M. Aoki, and H. Masuda, “Analysis and characterization of device variations in an LSI chip using an integrated device matrix array,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 17, no. 2, pp. 155–165, (2004).

[20] S. Saxena, C. Hess, H. Karbasi, A. Rossoni, S. Tonello, P. McNamara, S. Lucherini, S. Minehane, C. Dolainsky, and M. Quarantelli, “Variation in transistor performance and leakage in nanometer-scale technologies,” *IEEE Transactions on Electron Devices*, vol. 55, no. 1, pp. 131–144, (2008).

[21] S. Bickel, M. Br¨uckner, and T. Scheffer, “Discriminative learning for differing training and test distributions,” in *Proceedings of International Conference on Machine Learning*, pp. 81–88, (2007).

[22] “Top 5 Most Counterfeited Parts Represent a $169 Billion Potential Challenge for Global Semiconductor Market”, [online: https://www.electronicdesign.com/news/article/21194728/top-5-most-counterfeited-parts-represent-a-169-billion-potential-challenge-for-global-semiconductor-market], (2012).

[23] “Defense Industrial base Assessment: Counterfeit Electronics, Bureau of Industry and Security, U.S. Department of Commerce”, [Available: https://www.bis.doc.gov/index.php/other-areas/office-of-technology-evaluation-ote/industrial-base-assessments], (2010).

[24] “Estimating the global economic and social impacts of counterfeiting and piracy”, *A report commissioned by business action to stop counterfeiting and piracy (BASCAP), An ICC Initiative*, (2011). [Online: https://iccwbo.org/publication/estimating-global-economic-social-impacts-counterfeiting-piracy-2011/]

[25] “Field Programmable Gate Array Market Size, Share & Trends Analysis Report By Technology (SRAM, Antifuse, Flash), By Application (Military & Aerospace, Telecom), By Region, And Segment Forecasts, 2020 – 2027”, [Online:https://www.grandviewresearch.com/industry-analysis/fpga-market]

[26] SIA Anti-Counterfeiting Task Force, “Winning the battle against counterfeit semiconductor products”, pages 4–8, [Online: https://www.semiconductors.org/wp-content/uploads/2018/06/SIA-Anti-Counterfeiting-Whitepaper-1.pdf] (2013).

[27] U. Guin, D. DiMase, and M. Tehranipoor, “A comprehensive framework for counterfeit defect coverage analysis and detection assessment”, *Journal of Electronic Testing*, 30(1):25–40, (2014).

[28] U. Guin et al., “Counterfeit IC detection and challenges ahead”, *ACM SIGDA*, 43(3):1– 5, (2013).

[29] L.T. Wang, C.W. Wu, and X. Wen., “VLSI test principles and architectures: design for testability”, Elsevier, (2006).

[30] I. McLoughlin, “Secure embedded systems: The threat of reverse engineering”, in *Proceedings of IEEE International Conference on Parallel and Distributed Systems*, pages 729–736, (2008).

[31] F.S. Hossain, T. Yoneda, M. Shintani, M. Inoue, and A. Orailoglo, “Intra-die-variation-aware side channel analysis for hardware Trojan detection”, in *Proceedings of IEEE Asian Test Symposium*, pages 52–57, (2017).

[32] H. Wong, L. Cheng, Y. Lin, and L. He, “FPGA device and architecture evaluation considering process variations”, In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, (2005).

[33] S. Fujimoto, A. K. M. M. Islam, T. Matsumoto, and H. Onodera, “Inhomogeneous ring oscillator for within-die variability and rtn characterization”, *IEEE Transactions on Semiconductor Manufacturing*, 26(3):296–305, (2013).

[34] H. Onodera and H. Terada, “Characterization of WID delay variability using RO-array test structures”, In *Proceedings of International Conference on ASIC*, pages 658–661, (2009).

[35] M.M. Alam, M. Tehranipoor, and D. Forte, “Recycled FPGA detection using exhaustive LUT path delay characterization”, In *Proceedings of IEEE International Test Conference*, pages 1–10, (2016).

[36] V. Jyothi, A. Poojari, R. Stern, and R. Karri, “Fingerprinting field programmable gate arrays”, In *Proceedings of International Conference on Computer Design*, pages 337–340, (2017).

[37] "FPGA Applications", https://hardwarebee.com/fpga-common-applications/

[38] https://codilime.com/blog/fpga-programming-how-it-works-and-where-it-can-be-used/

[39] https://finance.yahoo.com/news/worldwide-fpga-industry-projected-reach-194500614.html

[40] https://www.gminsights.com/industry-analysis/field-programmable-gate-array-fpga-market-size

[41] F. Ahmed, M. Shintani and M. Inoue, "Accurate Recycled FPGA Detection Using an Exhaustive-Fingerprinting Technique Assisted by WID Process Variation Modeling", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 40 (8), pp. 1626-1639, (2021).

[42] Google Colab, https://colab.research.google.com/

[43] Y. Isaka, M. Shintani, F. Ahmed and M. Inoue, "Systematic Unsupervised Recycled Field-Programmable Gate Array Detection," in *IEEE Transactions on Device and Materials Reliability*, vol. 22, no. 2, pp. 154-163, (2022).

[44] X. Li, R. R. Rutenbar and R. D. Blanton, "Virtual probe: A statistically optimal framework for minimum-cost silicon characterization of nano-scale integrated circuits," IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers, pp. 433-440, (2009).

[45] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of Royal Statistical Society*, vol. 58, no. 1, pp. 267-288, (1996).

[46] E. Candes, "Compressive sampling," *International Congress of Mathematicians*, (2006).

[47] M. Shindler "Approximation Algorithms for the Metric k-Median Problem", [https://web.archive.org/web/20110927100642/http://www.cs.ucla.edu/~shindler/shindler-kMedian-survey.pdf]

[48] "Least Square Method: What It Means, How To Use It, With Examples", [Online: https://www.investopedia.com/terms/l/least-squares-method.asp]

[49] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor and Y. Makris, "Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain", Proceedings of the IEEE, Vol. 102 (8), pp. 1207-1228, (2014).

[50] H. Dogan, D. Forte and M. Tehranipoor, "Aging analysis for recycled FPGA detection", IEEE International Symposium on Defect and Fault Tolerance (DFT) in VLSI and Nanotechnology Systems, pp. 171-176, (2014).

[51] https://en.wikipedia.org/wiki/Anomaly_detection

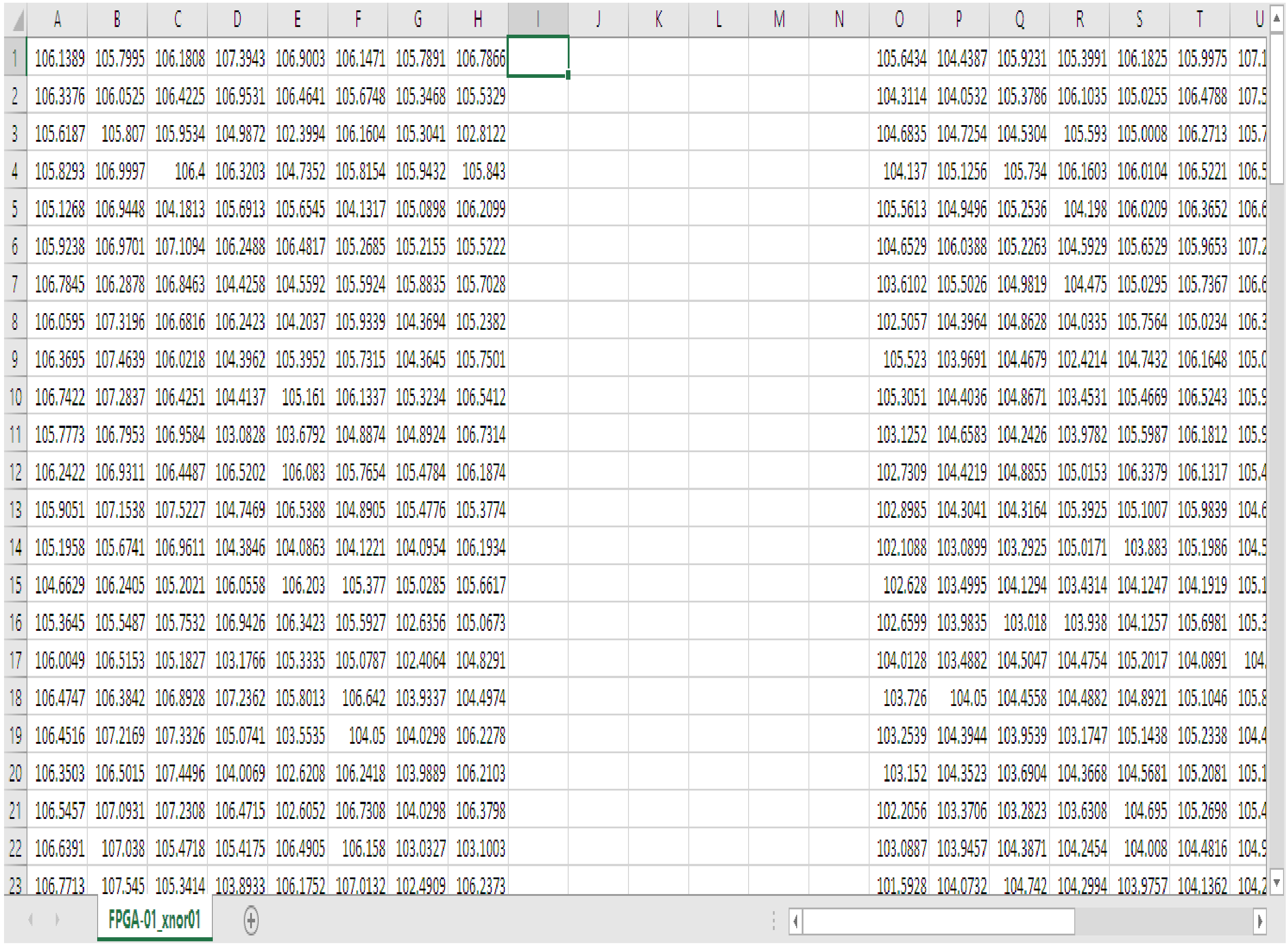
[52] https://www.elastic.co/guide/en/machine-learning/7.17/ml-gs-results.html

[53] https://en.wikipedia.org/wiki/Receiver_operating_characteristic

[54] https://anysilicon.com/understanding-ic-cost/

Appendix I – Sample Snippet of Datasets

There are some snippets of the datasets shown below which are used in this research work:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	106.1389	105.7995	106.1808	107.3943	106.9003	106.1471	105.7891	106.7866							105.6434	104.4387	105.9231	105.3991	106.1825	105.9975	107.1
2	106.3376	106.0525	106.4225	106.9531	106.4641	105.6748	105.3468	105.5329							104.3114	104.0532	105.3786	106.1035	105.0255	106.4788	107.5
3	105.6187	105.807	105.9534	104.9872	102.3994	106.1604	105.3041	102.8122							104.6835	104.7254	104.5304	105.593	105.0008	106.2713	105.7
4	105.8293	106.9997	106.4	106.3203	104.7352	105.8154	105.9432	105.843							104.137	105.1256	105.734	106.1603	106.0104	106.5221	106.5
5	105.1268	106.9448	104.1813	105.6913	105.6545	104.1317	105.0898	106.2099							105.5613	104.9496	105.2536	104.198	106.0209	106.3652	106.6
6	105.9238	106.9701	107.1094	106.2488	106.4817	105.2685	105.2155	105.5222							104.6529	106.0388	105.2263	104.5929	105.6529	105.9653	107.2
7	106.7845	106.2878	106.8463	104.4258	104.5592	105.5924	105.8835	105.7028							103.6102	105.5026	104.9819	104.475	105.0295	105.7367	106.6
8	106.0595	107.3196	106.6816	106.2423	104.2037	105.9339	104.3694	105.2382							102.5057	104.3964	104.8628	104.0335	105.7564	105.0234	106.3
9	106.3695	107.4639	106.0218	104.3962	105.3952	105.7315	104.3645	105.7501							105.523	103.9691	104.4679	102.4214	104.7432	106.1648	105.0
10	106.7422	107.2837	106.4251	104.4137	105.161	106.1337	105.3234	106.5412							105.3051	104.4036	104.8671	103.4531	105.4669	106.5243	105.9
11	105.7773	106.7953	106.9584	103.0828	103.6792	104.8874	104.8924	106.7314							103.1252	104.6583	104.2426	103.9782	105.5987	106.1812	105.9
12	106.2422	106.9311	106.4487	106.5202	106.083	105.7654	105.4784	106.1874							102.7309	104.4219	104.8855	105.0153	106.3379	106.1317	105.4
13	105.9051	107.1538	107.5227	104.7469	106.5388	104.8905	105.4776	105.3774							102.8985	104.3041	104.3164	105.3925	105.1007	105.9839	104.6
14	105.1958	105.6741	106.9611	104.3846	104.0863	104.1221	104.0954	106.1934							102.1088	103.0899	103.2925	105.0171	103.883	105.1986	104.5
15	104.6629	106.2405	105.2021	106.0558	106.203	105.377	105.0285	105.6617							102.628	103.4995	104.1294	103.4314	104.1247	104.1919	105.1
16	105.3645	105.5487	105.7532	106.9426	106.3423	105.5927	102.6356	105.0673							102.6599	103.9835	103.018	103.938	104.1257	105.6981	105.3
17	106.0049	106.5153	105.1827	103.1766	105.3335	105.0787	102.4064	104.8291							104.0128	103.4882	104.5047	104.4754	105.2017	104.0891	104
18	106.4747	106.3842	106.8928	107.2362	105.8013	106.642	103.9337	104.4974							103.726	104.05	104.4558	104.4882	104.8921	105.1046	105.8
19	106.4516	107.2169	107.3326	105.0741	103.5535	104.05	104.0298	106.2278							103.2539	104.3944	103.9539	103.1747	105.1438	105.2338	104.4
20	106.3503	106.5015	107.4496	104.0069	102.6208	106.2418	103.9889	106.2103							103.152	104.3523	103.6904	104.3668	104.5681	105.2081	105.1
21	106.5457	107.0931	107.2308	106.4715	102.6052	106.7308	104.0298	106.3798							102.2056	103.3706	103.2823	103.6308	104.695	105.2698	105.4
22	106.6391	107.038	105.4718	105.4175	106.4905	106.158	103.0327	103.1003							103.0887	103.9457	104.3871	104.2454	104.008	104.4816	104.9
23	106.7713	107.545	105.3414	103.8933	106.1752	107.0132	102.4909	106.2373							101.5928	104.0732	104.742	104.2994	103.9757	104.1362	104.2

Fig. Appendix I: Sample snippet of the dataset of FPGA-01's XNOR01 path

Outcome of the Thesis

- T. A. Tarique, F. Ahmed, M. Jenihhin and L. Ali, "Unsupervised Recycled FPGA Detection Using Symmetry Analysis," in the *Proceedings of International Conference on Electrical and Computer Engineering*, ICECE, BUET, Dhaka, Bangladesh, (2022). [Published]